

UNIVERSITY OF CALIFORNIA, MERCED

**Informative planning and robotic
exploration for surveying
environmental fields**

by

Lorenzo Ade Booth

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

in

Electrical Engineering and Computer Science

Committee in charge:

Prof. Thomas Harmon

Prof. Shijia Pan

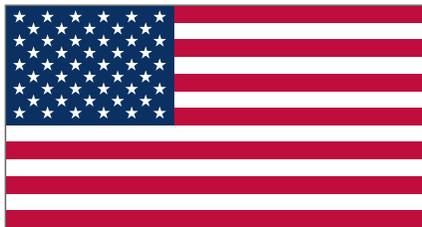
Prof. Stefano Carpin (Chair)

2024

To Victor Booth (Jan 30, 1925 - Sept 4, 2018), an exemplary citizen.

The work and efforts which comprise this doctoral thesis would not be possible without the wealth of resources and intellectual investments present in the United States of America.

It was an privilege and honor to be supported by the following research initiatives and programs that are funded by the American taxpayer. Specifically, these are: the National Science Foundation (NSF) National Research Traineeships in Intelligent Adaptive Systems (NRT-IAS) (DGE-1633722) and the Labor & Automation in California Agriculture (LACA) project, which is part of the University of California's Office of the President Multicampus Research Programs & Initiatives (UC-MRPI M21PR3417), the IoT4Ag Engineering Research Center (under NSF Cooperative Agreement No. ECC-1941529) and the US Department of Commerce, Economic Development Administration under Build Back Better Regional Challenge, Investment 07-79-07893 "Fresno-Merced Future of Food (F3) Initiative" and the UC Merced iCREATE project (Award 07-79-07913). Additionally, I thank the UC Merced School of Engineering for Multiple teaching assistant assignments over the years. I also thank the many Authors (*around the world*) of the various Free and Open Source softwares they Graciously have contributed to our collective knowledge. In this ability to Advance humanity through distributed efforts, we constantly improve.



Contents

1	Introduction	1
1.1	Overview	2
1.2	Field robotics for surveying	4
2	Related Work	6
2.1	Mapping, planning, and surveying	6
2.2	Autonomous surveying is constrained optimization	7
2.3	Optimal sensor placement	10
2.4	Informative path planning	12
2.4.1	Orienteering Problem	13
2.4.2	Planning under uncertainty	13
2.4.3	Single-robot IPP	15
2.4.4	Adaptive sampling	16
2.4.5	Multi-robot surveying	16
3	Principal Concepts and Theory	18
3.1	Kriging: as a framework for modeling environmental fields	19
3.2	Gaussian process regression: as a basis to inform sampling	21
3.2.1	Gaussian process regression	22
3.2.2	Spatial prior	24
3.3	Problem statement: Informative Path Planning	25
3.4	Sampling-based path planning	27
3.4.1	Rapidly-exploring Information Gathering (RIG) algorithms	27
3.4.2	Mutual information as a utility function for informative planning	29
3.4.3	Comparison of sampling routines	32
3.5	Conclusion	39
4	Distributed estimation of scalar fields with implicit coordination	41
4.1	Background	41
4.2	Problem Statement	43
4.3	Methods	44
4.3.1	Spatial prior	44
4.3.2	Exploration	45
4.3.3	Vertex quality computation	48
4.4	Experimental Evaluation and Discussion	49

4.5	Conclusion	54
5	Informative path planning for scalar dynamic reconstruction using coregionalized Gaussian processes and a spatiotemporal kernel	55
5.1	Introduction	55
5.2	Problem Formulation	57
5.3	Methods	59
5.3.1	Environmental Model	59
5.3.2	Spatiotemporal prior	60
5.3.3	Informative Planning	61
5.3.4	Information Functions	62
5.3.5	Convergence criterion	63
5.3.6	Path selection and planning	64
5.4	Experimental Evaluation and Discussion	64
5.4.1	Experimental setting	65
5.4.2	Consequences of the temporal prior	67
5.4.3	Ocean particulate mapping scenario	70
5.5	Conclusion	71
6	Unified adaptive and cooperative planning using multi-task coregionalized Gaussian processes	72
6.1	Introduction	72
6.2	Problem Formulation	75
6.3	Methods	76
6.3.1	Overview	76
6.3.2	Environmental model	76
	GP regression	76
6.3.3	Spatiotemporal and task priors	76
	Spatiotemporal prior	76
	Task prior	77
6.3.4	Utility Formulation	78
6.3.5	Path selection and planning	79
	General framework	79
	Convergence criterion and path selection	79
	Adaptive planning	79
6.4	Experimental Evaluation and Discussion	80
6.4.1	Experimental setting	81
6.4.2	Comparison of cooperative and adaptive planning	81
6.4.3	Environmental monitoring scenario	83
6.5	Conclusions	84
7	Concluding Remarks	86
7.1	Main conclusions	86
7.2	Future work	89

7.3	Closing thoughts	91
.1	Appendix B: Mathematical Notation and Terminology	93
.1.1	Matrix terminology	93

List of Figures

1.1	Analysis challenges for remote sensing	3
1.2	The future of GIS	3
2.1	Mobile robotics conceptual overview	7
2.2	Autonomous surveying conceptual overview	9
2.3	Objective functions for optimal sensor placement	11
2.4	Early example of informative planning for environmental monitoring	12
2.5	Vignette of the past and future of environmental surveying	17
3.1	Kriging in GIS software	20
3.2	Visualization of the model posterior covariance matrix	24
3.3	Schematic overview of a complete surveying system	26
3.4	Experimental setup for IPP over a 2D scalar field	33
3.5	Reward, cost, and RIC plots for different informative planning configurations	35
3.6	Example survey generated by the baseline coverage path planner	36
3.7	Example survey generated by informative planner using the a-optimal utility function	37
3.8	Data collection setting for lake experimental dataset	38
3.9	Experimental results from lake mapping scenario	39
4.1	Benchmark scalar fields used in the SOP-CC IPP experiments	50
4.2	Example maps obtained from planner/scenario combinations	51
4.3	Average map accuracy for different length-scales	52
4.4	Average MSE for different values of l_0 and number of robots.	53
5.1	Overview for spatiotemporal IPP experiments	57
5.2	A visualization of the benchmark (coverage) sampling scenarios	65
5.3	A comparison of map error and posterior variance for the advection/diffusion experiment	66
5.4	Example results from the ocean modeling experiments	70
6.1	An example of 3 robots in a turbidity monitoring mission	73
6.2	Schematic overview of the adaptive planning routine	80
6.3	Chart summary of advection/diffusion simulation experiment	81
6.4	Chart summary of ocean turbidity experiment	84

List of Tables

3.1	RIG parameters used for scalar field experiments	33
3.2	Summary of planning strategies performance and accuracy	34
3.3	Aggregated performance of RIG planner for A-optimal and D-optimal utility functions	36
4.1	Average statistics regarding visited and unvisited waypoints . . .	53
5.1	Aggregated map accuracy and posterior variance of various planners	68
6.1	Summary of average map error throughout and at the end of a survey mission	83

List of Algorithms

1	Rapidly exploring information gathering graph (RIG-graph) . . .	30
2	InformationGPVR()	32
3	SOPCC	46
4	Adaptive planning with SOPCC and GP-variance	47
5	UPDATE_REWARD	47
6	Information_GPVR-ST()	63

Acknowledgements

In some cultures, there is the idea that people are the products of community efforts— that “it takes a village” or that we “stand on the shoulders of giants”. Indeed, we are blessed to live in an age where many fundamental principles of matter and information are so well theorized that entire domains of study have emerged concerning the practical applications of electrons flying around in polished pieces of stone. It’s easier than ever to tap into the collective knowledge of billions on the *information super-highway* (or *charabia* of a chat-bot). This dissertation would not have happened without the books on geostatistics compiled by some brilliant guys before the internet-age and the free/open source software that abstract away some truly confounding mathematical optimizations. However minor my contributions may be, it’s a real privilege to participate in the ongoing human struggle to create order from the chaos of the universe.

And, as much as some people love to whine about how *burdensome* it is to be a student participant in academia, I’m blessed and truly honored to have been supported by our public institutions throughout this decade of scholarly pursuit. I’m grateful for my professional mentors: Dr. Stefano Carpin and Dr. Josh Viers who extended their hands of patient support even when I was not on my A-game. I really can’t thank the two of you enough. My committee, with Dr. Tom Harmon and Dr. Shijia Pan, thank you for your insight and example. Dr. Qinghua Guo, thank you for setting me on the path of research. Dr. Leigh Bernacchi, thank you for being my north star. If I am able to return more value to the world than I have taken, I will die with peace.

And, indeed I am blessed with the good fortune of a swell community of people in Merced, where I have made my home for over 12 years, and a supportive circle of colleagues and friends scattered across the world. All of you know who you are, there are too many to name, but I remember all of you including: the family (truly a family) from LACES, the swamp crew from the early days, the homies from Best Buy 1510 and 844 (RIP), r/a/dio, the friends from Merced College, collaborators at Kathmandu University, VICELAB and in the Robotics lab, the 3 generations of friends from UC Merced who I’ve seen come and go, the colleagues at Siemens and the greater community at the countless conferences and symposia that I’ve attended. All people who have seen me at different stages in personal development (the good times and the less good times) and yet were still there—I hope you’re all doing well. I hope you’re doing well for yourselves and for society broadly.

To my cats Durga (2019-2022) and Kali (2019-2024), I was blessed to share this Earth with you two for that brief moment and watch you grow into fine felines.

Victor Booth, you brought me to Merced and set the standard to which I aspire. Mother and Father, Rochelle and Ricardo, you laid the foundations well and I will never be able to repay your personal sacrifices.

And Lauren, my homegirl—you’re my greatest blessing.

Abstract

In the present day, we are collecting more information about our built and natural environments than ever before, to enable an unprecedented level of global prosperity. Additionally, we are relying on these insights to help us manage our natural and human systems so that we can maintain them sustainably for the foreseeable future. There still remains the motivation to push further in efficiently managing our built and natural systems, by achieving the gold-standard of intelligence — that is, systems that are able to reproduce and/or exceed human-level ability in surveying and environmental modeling. Such an ambitious task will either require a lot of labor or the strategic use of intelligent autonomous systems.

For autonomous robots tasked with surveying the spatial and temporal dynamics of a changing environment, there are a variety of (sometimes conflicting) criteria that must be satisfied over the course of a surveying mission: Where should the robot travel to? Which areas are worth observing? Does it make sense to revisit a previous area? How should the task be divided if there are multiple robots in the team? When should the mission stop? We can view this objective through an information-theoretic lens, where the robot is tasked with finding a path through a domain that maximizes the information that it collects along the way. In the robotics literature, this objective is known as *informative path planning* (IPP), which is NP-hard.

This dissertation explores a few dimensions of the IPP problem and presents a few information-theoretic approaches of generating solutions that satisfy various surveying criteria. The methods used in this dissertation are grounded in geostatistics and explore different applications of a core motivation: from surveying a scalar field at fixed monitoring locations with a team of robots, to reconstructing a time-varying phenomena in a continuous planning space. By specifying an information-theoretic utility function, it is possible to reconcile prior knowledge about a system with new knowledge obtained by the robotic surveyors. The reader will gain insights into the nuances embodied the practical applications of this method in single-robot and multi-robot systems.

Finally, the reader will be presented with a collection of topics that serve as points of departure from this dissertation into separate lines of further inquiry. The methods explored in this dissertation can be applied to a variety of environmental monitoring tasks, enabling a host of analyses from national security objectives, to the analysis of trade flows, and other agricultural and ecosystem management objectives.

1

Introduction

*“From Middle English *surveyen*, from Old French *sourveoir*, *surveer* (“to oversee”), from *sour-*, *sur-* (“over”) + *veoir*, *veeir* (“to see”), from Latin *videre*.... ultimately, from Proto-Indo-European **weyd-* (“to know; see”), a stative verb.”*

– anonymous contributors, *wiktionary* CC BY-SA

Surveys are essential to a variety of disciplines: from civil engineering, to ecology, even social sciences and market research. If there is a desire to make informed decisions, then it often comes with a desire to understand the *lay of the land* by visiting some locations, making some observations, and recording them for later analysis. Naturally, this leads to a question: “Where do we go?” Or more precisely, “Where do we send our robot, so we don’t have to go actually there?” A deep question that seems simple at first glance—within lies nuance, complexity, and lines of investigation that span multiple disciplines. This thesis explores these nuances and offers a few strategies for constructing robotic systems that can perform autonomous environmental surveys.

Consider the task of modeling a soil property in an agricultural field with a point sensor. Typically, samples are acquired at a certain number of locations and a model is constructed to “interpolate” between these point measurements, so that the manager can make informed guesses as to what properties can be found in the in-between spaces. Whether the sensor is wielded by a human or an autonomous robot, the sensing agent (the surveyor) is tasked with deciding where to capture observations in order to populate the model with training data. If the environmental properties are dynamic and can change over the course of the survey, the surveyor is also faced with the decision of measuring an unvisited location, or re-visiting an old location whose state may have changed since the last observation.

Now imagine that the surveyor is working under a constraint such as remaining daylight or the amount of fuel left in the vehicle. Now, the surveyor must be strategic about where to sample in order to arrive at the best estimate of the state of the world. An ideal robotic system would incorporate all of these competing objectives and produce an *optimal* surveying plan (with respect to

this desire to arrive at the most accurate map of the world), or at least one that has a known likelihood of being optimal.

With this framing in mind, we will now develop a more complete context and motivation for the contents of this thesis.

1.1 Overview

Geospatial models of the environment can be built from observations collected in a variety of ways, including: from physical sensors deployed at different locations, remote sensors which can observe a larger area from a given vantage point, or even from physically-realistic models of larger systems. Robots are attractive platforms for deploying physical and remote sensors, in part due to their ability to acquire observations in situations where it would be physically hazardous or costly for human operators, and where operational challenges exist (such as performing surveys at night or on a continuous basis). Additionally, multi-robot systems can multiply the effectiveness of a sampling campaign, both on a cost basis and in the potential area covered in a single setting.

Admittedly, this claim might be met with skepticism by scientists who are familiar with the current state of earth science and surveying. NASA has eighteen robotic spacecraft currently active in its Earth science fleet [1] and hundreds more robotic earth observation (EO) satellites are currently gathering data every day, maintained by governmental agencies and commercial entities across the world. In some ways, we are no longer limited by data when it comes to Earth observation. [Figure 1.1](#) and [Figure 1.2](#) describe a paradigm where the most effort no longer lies in data acquisition—instead it lies in the ability to create meaningful insights from the information that we have.

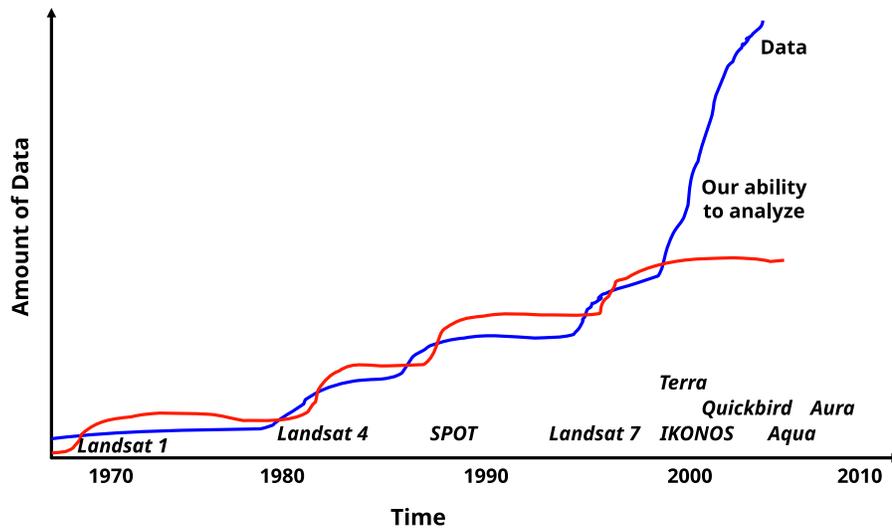


FIGURE 1.1: Remote sensing analysis challenges. 15 years ago we less limited by the availability of EO data than our ability to analyze the wealth of data. Figure by Prof. Qinghua Guo from my first GIS course 10 years ago.

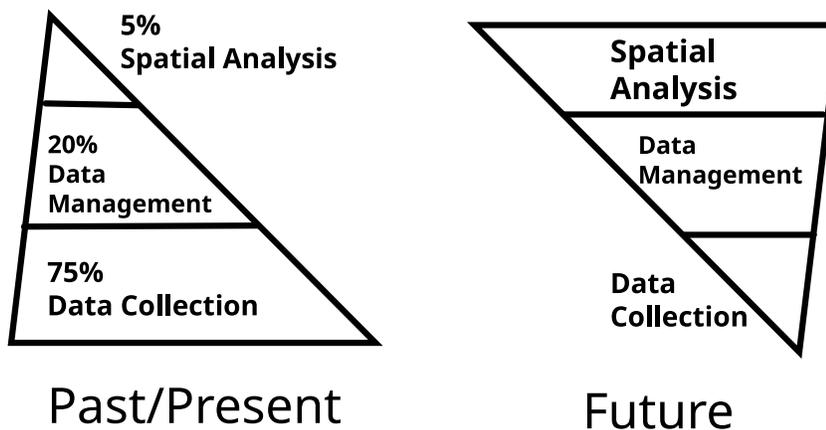


FIGURE 1.2: Was this a good prediction 15 years ago? I'd argue that our ability to analyze large amounts of spatial data has been greatly bolstered by "big data" management tools and other hardware and software infrastructure. When the Landsat archive went online in 2008, you still had to download tiles and perform a lot of pre- and post-processing on your local machine. With Google Earth Engine, mosaicking, color mapping, and zonal statistics can be performed on a data set that is /already/ prepared with atmospheric corrections. Time-series for decades of observations can be calculated in seconds! Figure by Prof. Qinghua Guo (modified from Ronald Briggs) from the same introductory GIS course.

Underpinning the modern "art" of satellite remote sensing are algorithms that allow scientists to *retrieve* information about the state of earth systems from the wavelengths of reflected light that are observed by satellites hundreds of kilometers above the Earth's surface. Many of these *retrieval algorithms*

are physics-based and are grounded in the optical/emissive properties of matter. Other algorithms are model-driven, enabled by statistical analysis (read: machine learning) that relate conditions observed on the ground with observations at the satellite. Even for the physically-based algorithms, there is still the motivation to continuously test the effectiveness of these retrieval algorithms through validation through co-located observations.

In spite of the terabytes of satellite images being collected each day, there continue to be new satellite constellations and drone-based tools that enable us to acquire more precise observations at a faster operational cadence—and with this, comes a never-ending desire to derive increasingly precise insights from these new tools. Ultimately, the gold standard for data acquisition is direct-observation, that is *actually going there* and acquiring a sample. It is the reason why we send technicians to the field and rovers to other planets. There will always be a role for ground surveys and arguably, in a world with a growing amount of remote sensing, it more essential than ever to verify and validate with direct surveying. However, the logistical difficulty and expense of *actually going* to some locations create a dearth of the “expensive” sorts of data that are very useful for a variety of observation tasks. This is a need that can be uniquely met by mobile robots, as long as the right balance is attained between usefulness and operational cost.

1.2 Field robotics for surveying

When applied to field robotics, surveying can be understood as a robotic exploration task. Specifically, sensors can be deployed on a robotic platform, which in turn is tasked with traveling across a territory. Observations of the environment obtained by the sensors are used to create a map of the territory. Choices of physical platform, sensors, and sampling frequency all affect the performance of the robotic survey. Regardless of the platform and sensing modality, all autonomous surveys depend on a system for *planning* the *path* taken by the autonomous system during the survey execution.

There are different approaches to robotic planning in general. In the simplest case, a system may be tasked with navigating toward a goal. In more realistic scenarios, the robot may find the most efficient path toward some goal while avoiding hazards and obstacles. In the surveying task, there may not be a final destination to navigate toward *per se*, and there may be other practical constraints on navigation, such as a fuel or energy limit that restricts the total possible distance traveled.

The main contributions of this work are:

1. Quantifying how some planning approaches are "better" than others for the task of conducting environmental surveys with sensors on mobile robots.
2. Demonstrating how the meaning of "better" also depends on the modeling approach used in constructing the model of an environment and on the choice of objective function(s).
3. Showing how given combinations of sensing modalities and modeling approach, there exists a best or most optimal approach for planning a path for sample collection.

This dissertation is organized into chapters that build upon this core theme with different dimensions to the robotic surveying problem. **Chapter 2** presents a brief overview of related literature in field robotics and adjacent disciplines. **Chapter 3** presents core concepts and theory, from the geostatistical underpinnings of certain classes of environmental models, to the information-theoretic framing that underpins the sampling strategy explored in this dissertation. This chapter also contains an exploration of a prototype planning algorithm, variants of which will be explored in the subsequent chapters. **Chapter 4** presents a study that demonstrates an example of how to augment a path planning algorithm with a utility function suited to the environmental surveying task. **Chapter 5** extends the planner introduced in **Chapter 3** to consider the more realistic scenario of monitoring dynamic environments that change in space and time. **Chapter 6** completes this planning framework with extension that can be used by a multi-robot, distributed team-surveying mission, that is able to adapt the planning strategy to changing conditions. Finally, **Chapter 7** concludes by identifying avenues for further extension of this work and investigation of alternate experimental scenarios.

2

Related Work

Consider the practical motivation poised in the introduction: “*We want to survey an environment, but we don’t want to have to actually go there ourselves. So, where do we send our robot, so get the best survey?*” We made the claim that this deep question holds nuanced complexity that lies across multiple disciplines. This section will explore the previous work that has been accomplished along different lines of investigation. By the end of this chapter, the context will be established for framing this thesis within the current state of the art.

2.1 Mapping, planning, and surveying

Figure 2.1 presents a conceptual overview of a typical robotic system represented as a rotated Sierpiński triangle (S_1). Starting from the top right ($S_{1,2}$), mobile robots are typically equipped with sensors that allow it to create some sort of representation of the environment in the process of *perception* (although not always, see the eponymous vehicle by Braitenberg[2]). This representation can be a map, such as an *occupancy grid*, which identify which areas in the world are traversable vs occupied (with an obstacle). Next, the robot can establish where it is in the map through *localization*. Finally, a path planner can issue a trajectory to a destination and motor controllers can move the robot toward its prescribed goal. Many robots perform *simultaneous localization and mapping* (SLAM) and these processes typically are performed on a continuous basis as a mobile robot maneuvers through the world.

When a map has been constructed (or is previously known), mobile robots are often tasked with *safely* traversing an environment from a starting location to a destination. Here, safety can be defined a variety of ways, including: physical constraints (eg. static obstacles like walls or moving obstacles like vehicles and people) and vehicle constraints (eg. kinodynamic limits regarding how fast a robot can turn without flipping over). A low-level *planning algorithm* can produce a series of obstacle-free paths satisfying desired safety requirements, and there is a rich literature of different approaches to robotic path planning [3].

If one were to search the literature for “robotic mapping”, most studies would relate to this task central to perception and state estimation, which is core to modern mobile robotics. From now on, we will concentrate on the case where the mobile robot is tasked with a *surveying task*. That is: the robot is tasked with building a model (or map) of an environment *separate* from that which is required for safe navigation (path planning).

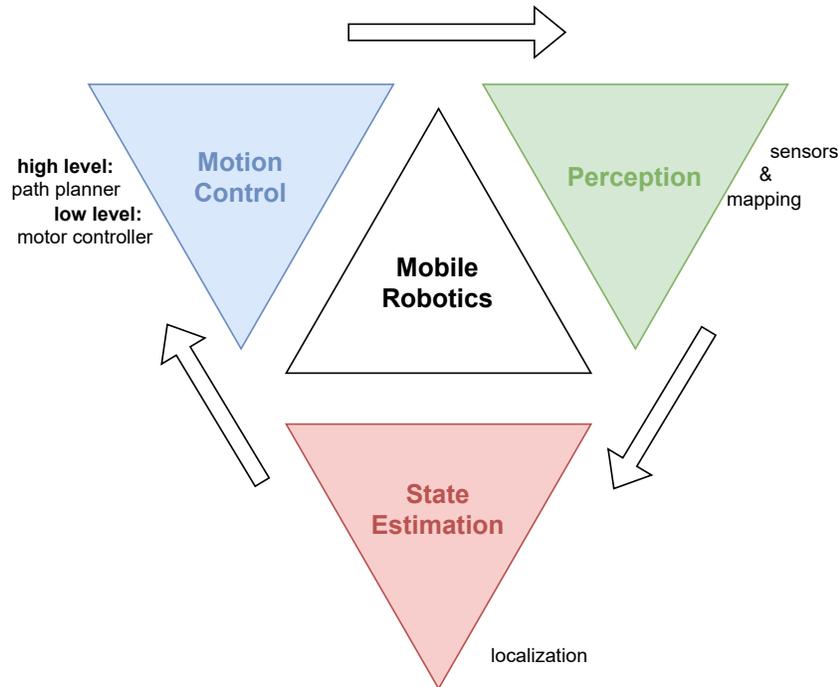


FIGURE 2.1: When considering a mobile robotic system, there are three central pillars. Perception (“What do I see?”) feeds into state estimation (“Where am I?”), which feeds into motion control (“Now, where do I go?”) in a loop (“Now, what do I see?”).

2.2 Autonomous surveying is constrained optimization

Let us further refine the surveying task and consider a particular motivation, where:

1. We desire to gain insight about some unknown **spatial environment** (eg. to create a model of the environment)
2. We desire to attain insights through limited **observations** of the environment

Consider the motif presented in [Figure 2.2](#). Each observation has a few components, namely: the *outcome* that the agent has witnessed (eg. the output of a function) and the *context* of the observation (eg. the input of a function).

Naturally, for observations that have a **spatial context**, we can assign an index corresponding to:

- The location of the observation in space (*where* did we acquire this observation?)
- The location of the observation in time (*when* did we acquire this observation?)

It is important to add that in the real-world, all observations and state estimations will be clouded with noise, as most of the sensors that we commonly use are not perfect and obtain noisy measurements. This is especially relevant for mobile robots, as environmental factors (eg. heat, dust, vibration, impact, water exposure) can introduce noise to sensors which may produce accurate readings on the bench. Even worse, we can't rely on our actuators to reliably reach desired states. For example, if a robot is commanded to move a meter to the north-west, any number of factors could cause the robot to miss the target including: wheel slippage on dirt and inaccuracies in wheel movement due to under-inflated tyres. Therefore, our planning algorithms have to be able to cope with *uncertainty* in the state of the robot and the environment it operates in. In the broader literature of decision theory (and robotics), this is known as *decision making under uncertainty* [4]. In robotics, this is often achieved using probabilistic methods [5] such as Markov decision processes (MDPs), which consider uncertainty in the evolution of a system. When coupled with uncertainty in the state of a system due to observational uncertainty, this becomes a Partially-Observable Markov Decision Process (see [subsection 2.4.2](#)).

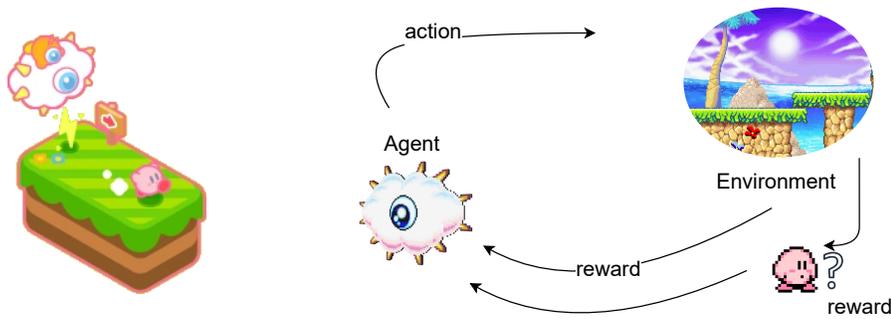


FIGURE 2.2: Kracko wants to get the lay of dreamland in order to find Kirby and deliver his thunderbolts of love. He can either stay where he is, hoping Kirby will come or go looking for him. Where does he decide to go? Image assets: ©Nintendo Co., Ltd.

Thus, if we are tasking a robot with autonomously gathering observations, for our surveying task, we can frame our objective as an **optimization**:

We want to acquire the *best* observations with regard to the *criteria* of our surveying *objective* and the *hand that we were dealt*.

For different definitions of the italicized words, we can conjure different approaches, theories and methodologies. For example:

- **the objective:** Is our goal to build a complete model of a spatial system? Are we trying to reconstruct some variable from sparse observations (interpolation)? Or, are we just monitoring for the presence/absence of some item of interest?
- **the criteria:** If we are making a model, are we trying to maximize its accuracy? Or minimize the effort required to gather observations?
- **the hand that we are dealt:** What is constraining our operation? Are limited by how much time we have? By how much distance we can travel in total? By how far we can travel from a base of operation?
- **the best observations:** Relative to what? A counterfactual “oracle” that allows us to compare against the *true optimum*? Or, relative to what a human surveyor would traditionally do, given the current best practices?

Let us now concentrate on a specific application: where we wish to construct a model of a physical environment, via observations collected from sensors on a robotic platform. We desire a methodology that guides where those observations should be collected.

2.3 Optimal sensor placement

One approach considers the task of monitoring a spatial phenomena using a limited number of static sensors. This problem of *optimal sensor placement*, refers to the question of deciding where to place these sensors (eg, thermometers in a refrigerated warehouse) in order to best accomplish the monitoring task.

Geometric approaches: This task can be formulated as an instance of the art-gallery problem, where sensors are assumed to have a fixed sensing radius and the algorithm must find an optimal arrangement of these sensing “disks”. As this is an NP-hard problem, different solutions have been proposed including polynomial-time approximation schemes [6] and other greedy algorithms [7].

Inferential approaches: The task can also be considered from the perspective of knowledge, or *information*. If the goal is to construct a model of the environment, then we desire observations that improve our model’s predictive ability in some statistically quantifiable way. More abstractly, we can specify a goal of gaining information *per se*, which can be used to establish an information-theoretic criterion as to how beneficial it is to sample/observe at a given location. In fact, these are two ways of describing the same overall objective, and this framing is especially applicable to the field of *spatial statistics*, where models must account for the probability that phenomena that are physically proximate are more likely to be correlated than phenomena that are distant (an insight that is known as Tobler’s First Law of Geography [8]).

From this framing, different approaches have been used to model spatial correlations in multivariate systems, where observations are considered to be realizations of a spatially-distributed random variable. A popular approach is to use prior knowledge of a system to construct a *Gaussian process* (GP) model, which allows for predictions of model uncertainty to be made across a spatial field [9]. From this, sensor placement strategies include: sampling at locations with the highest model variance (maximum entropy) [10] [11] and sampling according to an optimal design analysis [12].

However, as identified by several investigators ([14], [15], [13]), entropy-based placement criteria usually result in a maximally dispersed placement of sensors, causing an undesirable effect where sensors are placed along the borders of the sampling area of interest. If we model the sensors as having a circular sensing radius, then half of the sensing area would fall outside of the region of interest, for these sensors placed on the border.

[12] proposed *mutual information* as a preferred information-theoretic criterion. In their formulation, sensors are placed at locations that result in the greatest expected reduction of uncertainty *across* the entire region of interest. [13] applied this principle to the case where an environment is modeled as

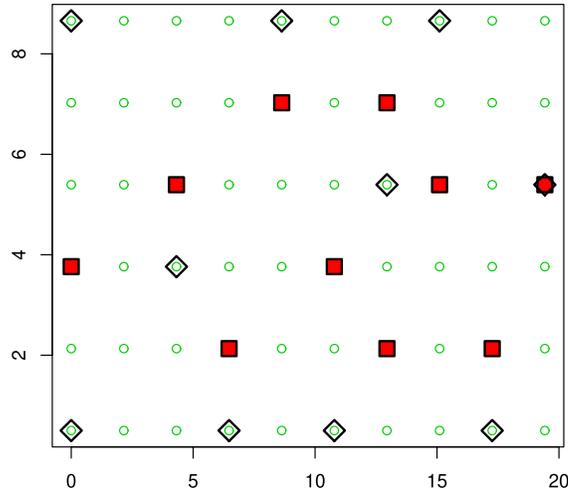


FIGURE 2.3: Objective functions for optimal sensor placement: Here, an example is shown for sensor placement using different information criteria. Positions determined using entropy are marked by diamonds and positions chosen using mutual information are marked by squares. Note that while entropy results in the most even coverage of sensing stations, there is “wasted information” for all of the sensors placed at the edge of the survey domain, as the only half of a sensor’s observation window falls within the sensing domain. From: Krause et al. 2008 [13]

a GP and they proposed a polynomial-time approximation algorithm that is optimal to within a constant-factor. More recently, there have been attempts to balance between optimal and efficient sensor placement for surveying and monitoring objectives, including heuristic [16] and reinforcement learning-based approaches [17].

When it comes to robotic surveying, we typically concern ourselves with a robotic sensing agent as a surface vehicle (a boat, a plane, a rover) that moves in \mathbb{R}^2 and whose observations are used to construct a model in the same domain. However, this task can be reduced or extended (*caveat emptor*) to an arbitrary number of dimensions. For example, in [18] entropy-based sampling method was used to reconstruct a 1-dimensional model of river pH along a transect, based on a GP regression.

Wireless sensor networks Once sensors have been placed, there remains the additional task of retrieving observation from a deployment of sensors. Studies in this domain operate under different assumptions regarding the ability to communicate sensor observations to the surveyor. For networked sensors, communication distance may serve as a constraint in determining where sensors can be located [19] and there exists a large domain of research, conferences, and publications devoted to the study of wireless sensor networks (eg. IPSN, SenSys, and MobiCom).

2.4 Informative path planning

We can consider some alternate scenarios where sensors are deployed statically, but are unable to communicate their observations unless visited by a mobile sensing agent (such as a robotic platform). Alternatively, the sensors can be deployed on the mobile robot. In both scenarios, one can conceive of an algorithm that commands the mobile robot to travel to locations that are optimal with regard to a surveying objective (see [section 2.2](#)).

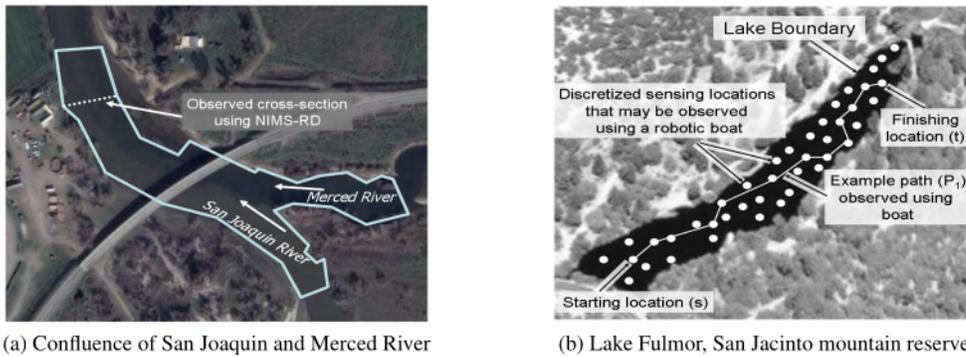


FIGURE 2.4: An early example of informative path planning for environmental monitoring. On the left, the informative planner operated on a aquatic sensing platform towed across a transect (indicated with the arrow) at a river confluence. On the right, the authors performed IPP on a data set acquired from an existing network of aquatic sensors. From: Singh et al. 2009 [20]

Generally, *Informative Path Planning* (IPP) [21], [22] describes a high-level planning procedure, where plans are generated for mobile robots according to an information objective. For surveying, this information objective is typically related to the accuracy of the resulting survey, collected from the robot's observations. Depending on the planning space, informative paths can be constructed in free space, or paths can be constructed from a set of measurement locations that may correspond to waypoints or sensor locations.

[20] introduced the *Multi-robot Informative Path Planning* (MIPP) problem, where: a spatial domain is partitioned into a finite number of sensing locations (eg. monitoring stations) and the robot must traverse a path in this space. The robot incurs traveling costs as it moves from location-to-location and the robot gathers information as it visits different sensing locations. The authors presented a recursive-greedy algorithm for solving the MIPP problem, by leveraging the submodular information function presented in [13].

2.4.1 Orienteering Problem

If each node has a fixed reward and the objective for the robot is to find a path that maximizes the sum of these rewards, then the problem can be formulated as the Traveling Salesman Problem With Profits (TSP-WP), more commonly known as the orienteering problem [23]. Along with the TSP, this is a well-studied problem in operations research and it is known to be NP-hard [24]. The OP can be classified into cases where the robot is *rooted* (that is, starts and finishes at a pre-specified location) and extended to consider teams of robots (TOP), where multiple tours must be computed [25]. If the cost of movement to a node is not known *a priori*, then the problem is known as the *Stochastic Orienteering Problem* (SOP) [26] (see [27, 28, 29] chapter 3 for an extension of this concept). When applied to environmental monitoring tasks, correlations can be leveraged between observation locations to provide a utility that is quadratic with respect to the size of the node network [30]. Variants of the OP have also been applied to scheduling tasks, such as the Optimal Tourist Problem, where a robot is tasked with visiting n spatially-distributed points of interest for a surveillance objective [31].

2.4.2 Planning under uncertainty

As with many NP-hard tasks, there is an incentive to find approximate solutions to problems such as the OP that are approximate (ideally with a guarantee over a set of conditions). Alternatively, researchers explore heuristics that provide a desirable trade offs between accuracy and computational complexity. One method that has found a great deal of success in both planning and robotic control tasks is reinforcement learning (RL) [32]. The problem of RL is typically formulated as the optimal control of an incompletely-known Markov decision process (MDP). This framing is sufficiently general so as to describe a variety of planning tasks: from board games like chess and Go [33] (which led to the well-known AlphaZero search algorithm [34]) to energy grid modeling and optimization [35] and other robotic control tasks, such as balancing an inverted pendulum [36]. It is also possible to frame IPP as an incompletely-known MDP, specifically a Partially-observable Markov decision process (POMDP). This formulation and RL methods in general have guided a host of learning-based approaches to solving IPP problems and we will briefly describe the main thrusts of investigation in this domain.

MDPs are a mathematical framework for modeling sequential decision making in situations where the probability of reaching a state by a given action is modeled as samples from a known transition probability kernel (this can

incorporate random perturbations by unknown factors in the environment). The goal of an MDP is to find a strategy (policy) that maximizes the cumulative reward, typically through techniques like value iteration and policy iteration, which iteratively evaluate and improve policies. An example of a typical MDP framing is presented in [Figure 2.2](#). At each time step t , an agent receives feedback about the state of an environment (S_t). From this information, the agent selects an action to perform (A_t). One time step later, the agent receives feedback corresponding to a numerical reward of its action R_{t+1} , the environment proceeds to a new state (S_{t+1}), and the process repeats [\[32\]](#). When the environment is not completely known by the agent, the scenario is termed a Partially Observable Markov Decision Process (POMDP) [\[37\]](#). As one can imagine, it is impractical to store and compute all of the possible states and actions of a POMDP, even when the possible states (eg. waypoints corresponding to sensor deployments). Planning in a gridworld (eg. Cartesian grid in \mathbb{Z}^n) or in a continuous space faces an even larger explosion in the size of the state-space (see the remark on the next page). Many approaches that build from the POMDP formulation use tree and graph search algorithms to find rewarding paths through the decision space, such as Monte-Carlo Tree Search (MCTS) [\[38\]](#) [\[39\]](#).

In [\[40\]](#), the authors partition approaches to planning under uncertainty into three main categories: *simulation-based* approaches, *infinite-horizon* strategies, and *receding-horizon* strategies. In their taxonomy, simulation-based approaches generate a few candidate paths and simulate their performance relative to some information metric (eg. [\[41\]](#)), receding-horizon approaches compute a policy over a finite number of control actions into the future (eg. EKF-SLAM in [\[42\]](#)), and infinite-horizon approaches compute a policy over the entire planning space of the vehicle (which may be constrained by a movement budget), with IPP approaches falling under this last category.

A finite MDP is generally defined by a 5-tuple (S, A, T, R) , comprising:

- S : A finite set of states (the *state space*).
- A : A finite set of actions available to the agent (the *action space*).
- T : A function $T : S \times A \times S \rightarrow [0, 1]$ that defines the probabilities of transitioning between states, i.e., $T(s, a, s') = P(s'|s, a)$ (the *state transition probability function*).
- R : A function $R : S \times A \rightarrow \mathbb{R}$ that assigns a scalar reward for taking action a in state s (the *reward function*).

Some formulations also add a term γ to form a 5-tuple, with:

- γ : A scalar in the range $[0, 1]$ that determines the present value of future rewards (the *discount factor*).

A POMDP extends this into a 7-tuple $(S, A, T, R, \Omega, O, \gamma)$, adding:

- Ω A finite set of observations $o \in \Omega$ (the *observation space*).
- O A function $O(o \mid s', a)$ that establishes the likelihood of observing a particular state given the new environmental state and the previous action (the *observation model*).

The advantages of an MDP formulation is that we can craft robust strategies to guide a robot at any moment. Like a game of chess, we only need to know the current state of the board in order to reason about the next best move. However like a game of chess, there are far too many combinations of moves and future game states to calculate what the *best* next-move is. Unlike chess, the dynamics of a robot and its environment mean that there is no guarantee that a expected state will actually be achieved after a given action. Also unlike chess, we often do not have visibility of the entire state of the world when deciding where to go. This “partial-observability” adds an additional layer of stochasticity to the inputs of our planning algorithms.

2.4.3 Single-robot IPP

IPP methods that focus on **modeling hotspots** encompasses objectives that aim to understand regions where a spatial field deviates significantly from average values. For example, in a temperature field a literal *hot spot* is a region where the temperature is unusually higher than the rest of the field. Hot spots do not have to be regions where the expected value differs from the norm, it can also include regions with unusually high variability. In [38], a planning algorithm is designed to maximize the amount of time a USV remains in a hot-spot region, defined as a zone above some threshold.

A special case describes efforts where the hotspot region is one where the spatial field exceeds (or falls below) some threshold value. Here, the goal is to acquire information about this “front” (the boundary region where the transition occurs). This task can include: determining if a front is present, modeling where it is, and dynamically tracking its movement and evolution [43] [44].

Sometimes, the hotspot corresponds to the physical processes of a plume of a high-concentration substance that advects and diffuses in a fluid medium (liquid or gas). In these scenarios, it may be desirable to obtain the source of the plume and/or its extent. A large body of works leverage the physical equations

that govern the evolution of fluid plumes to identify “plume source” and/or a “plume characterization” based on measurements from an autonomous robot [45] [46] [47]. Some of these efforts leverage biologically-inspired algorithms [48] or very simplistic heuristic behaviors [49].

2.4.4 Adaptive sampling

In [40], the authors categorized IPP methods as planning within an infinite-horizon, or more accurately the entire time horizon that the robot is able to operate within. These approaches can involve the calculation one path plan for the entire surveying task, or one path plan per episode, where the surveying occurs over several episodes. In this later case, there is an opportunity to incorporate the information learned in the previous surveying runs and *adapt* future surveying runs based on an updated prior. These **adaptive sampling** planners can include algorithms that update after each measurement or those that employ an episodic planning strategy that alternates between planning, execution and re-planning (in a receding horizon). Adaptive sampling methods are limited to robotics path planners and [50] provides a review of different strategies used for Kriging interpolation. More recently, reinforcement learning approaches have been applied toward adaptive IPP [51] [52].

Another scenario where adaptive sampling has been well-studied is multi-robot sampling. In these scenarios, each robot can adapt to the presence of and/or the observations from another robot in a surveying deployment. In the next section, we will explore multi-robot IPP in more detail, however as a domain of study path planning for multi-robot mapping has been investigated [53] even before the IPP problem emerged as a common motif in the literature [54]. A common scenario aims to establish strategies to partition an environment among a set of sensing vehicles. These *coverage control* planning algorithms may direct robots toward a static sampling location, achieving something akin to a space-filling Voronoi partitioning of the environment [20] or it may establish paths within a partitioning [55] [56]. Coverage algorithms may operate in free space or over a topology informed by the structure of the environmental parameter of interest [57].

2.4.5 Multi-robot surveying

IPP with multiple sensing agents has been accomplished with *sequential-allocation*, where the path planning algorithm is executed k times for k -number of robots [20]. Other approaches include merging observations from different robots to form a distributed consensus [58] and splitting the sensing space into

tours [59] or Voronoi partitions [60, 61, 56]. Planning approaches for different sensing modalities typically distinguish between the area that can be observed by one measurement [62]. Altitude-dependent sensor and noise models have been incorporated into IPP approaches for unmanned aerial vehicles [63]. IPP can also be considered over networks of static sensing nodes, where movement is performed between discrete locations [20, 64, 21, 65]. Different modeling objectives also guide different IPP approaches. For example, cross-entropy-based optimization techniques have been used to identify the most informative trajectory *en route* to a fixed goal [66]. Myopic planning routines have been used for robots tasked with identifying the source and characterizing chemical plumes using the Gaussian plume model [67]. In aquatic environments, modeling approaches often involve locating and/or characterizing hot spots or frontal regions of interest (not to be confused with *frontiers*) [44]. Finally, adaptive planners are able to improve the environmental model and have been used to update the informative planner during the course of a survey [68], [52], [69].



FIGURE 2.5: The author (foreground) engaged in a traditional environmental survey task assisted by a colleague and a robot (background).

3

Principal Concepts and Theory

In this work, we consider the problem of reconstructing an environmental phenomenon given a limited number of observations, collected along a path collected by a mobile sensing agent. The phenomenon is represented as a value that varies in space (a scalar field) and/or space and time (scalar or vector field). *The objective is to produce an algorithm that can solve this problem, optimizing for the best reconstruction of the environmental field, with the lowest costs incurred while traveling.

It necessary to specify a few assumptions about objective and operating environment as these are foundational to the methodologies employed in this thesis. For the environmental field:

- "Modeling" the environmental field is the same as being able to reconstruct the entire field at arbitrary locations in our region of interest.
- While the environmental phenomena being modeled may arise from deterministic, physical processes, the number of interactions are so numerous and complex that it is impossible or impractical to produce an analytic solution that can reliably calculate what can be expected at a given location in space and time.
- This complexity makes it appear that the environmental phenomenon is a stochastic process. That is, it can be modeled as randomly varying in space and/or time. Therefore, we can use statistical methods to arrive at a "likelihood" for the field given a set of input parameters.

The robotic path planner is tasked with producing a path through an environment for a mobile robot with a sensor to travel and collect observations. This is yet another instance of the exploration-exploitation dilemma [70], where at any give moment a decision must be made to incur resources traveling and exploring a domain, or adopt a *satisficing* strategy and use the existing set of observations for the modeling task. [Section 3.3](#) will present a formal

mathematical description of this task. Now, let us consider the modeling objective.

3.1 Kriging: as a framework for modeling environmental fields

In geostatistics, practitioners are often faced with interpolating between sparse measurements obtained over a spatial domain. For example, a prospector may want to predict a mineral fraction of a soil (say, % clay) from a set of n samples obtained randomly across a plot of land. A reasonable (albeit naïve) approach would be to apply a simple regression to the data, with parameters chosen by minimizing the sum of squared deviations between the regression and the measured values. For example, the prospector may perform a bilinear interpolation, (linear interpolation in the x and y component axes), using the method of ordinary least squares (OLS) to estimate the values of the linear model [71].

However, there exist better-performing models that have better predictive accuracy due to their ability to leverage knowledge that we know about the environment (see: Tobler’s First Law in [section 2.3](#)). For example, OLS does not take into account spatial correlations—if we find a streak of high-clay corresponding to an ancient alluvial deposit, then we would expect all of the measurements within this deposit to also have a high clay content. However a simple linear interpolation may smooth out these “outliers” caused by natural features.

Kriging is a method of spatial interpolation that overcomes many of the shortcomings of these least-square methods. It is a multi step process, where samples of the environment are used to create a distance-dependent correlation function, and then this function is applied to make predictions. From [72], we will establish the Kriging paradigm. Following from the framing in the introduction of this chapter, our environmental phenomenon $\mathbf{Z} \equiv (Z(\mathbf{x}_1), \dots, Z(\mathbf{x}_n))^T$ is thought as arising from a process with a significant random (stochastic) component. The process Z is *modeled* as follows:

$$Z(\mathbf{x}) = \mathbf{m}(\mathbf{x})^T \boldsymbol{\beta} + \varepsilon(\mathbf{x}), \quad (3.1)$$

where:

- \mathbf{m} is a smoothly-varying deterministic function that models the phenomenon Z . It is parameterized by $\boldsymbol{\beta}$, a vector of p unknown coefficients and has values in \mathbb{R}^p .
- $\varepsilon(\mathbf{x})$ is a random component with a known covariance structure that represents small-scale processes that are unobserved or not quantified. This term can also include observation errors and is often represented as a Gaussian noise that is independent of the random vector $\boldsymbol{\beta}$

We observe $\mathbf{Z} = (Z(\mathbf{x}_1), \dots, Z(\mathbf{x}_n))^T$ and want to make a prediction for $Z(\mathbf{x}_0)$. If we know $\boldsymbol{\beta}$ (the coefficients of the model), then we can perform a straightforward evaluation of Equation 3.1 to obtain the expected value for Z . If we do not know $\boldsymbol{\beta}$, then we are tasked with finding a linear predictor, usually of the form $\lambda_0 + \boldsymbol{\lambda}^T \mathbf{Z}$, where λ_0 is the intercept term and $\boldsymbol{\lambda}^T \mathbf{Z}$ is a vector of coefficients corresponding to the parameters of $\boldsymbol{\beta}$. The goal then is to minimize $E \left\{ Z(\mathbf{x}_0) - \boldsymbol{\lambda}^T \mathbf{Z} \right\}^2$ subject to an unbiasedness constraint. If we solve this constrained optimization for $\boldsymbol{\lambda}$, we obtain $\boldsymbol{\lambda}^T \mathbf{Z}$, which is called a best linear unbiased predictor (BLUP) for $Z(\mathbf{x}_0)$. It is also known as the *kriging estimator* in ordinary kriging or the *spatial BLUP* and its form depends on the model \mathbf{m} adopted for the random variable Z . These weights can also be obtained empirically by fitting a variogram model to observed data (see [9] for an extended history and treatment of variogram analysis in spatial statistics).

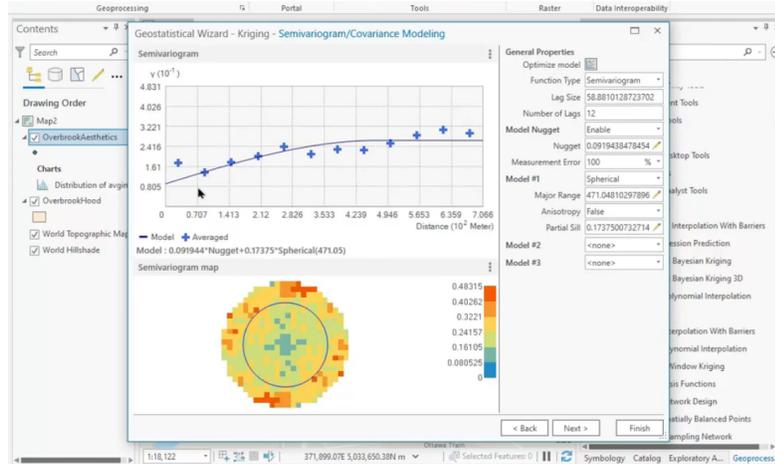


FIGURE 3.1: Screenshot from a Kriging interpolation model construction in Arc Pro (Image Credit: @TheGeomatician).

Kriging is named after the technique first used by South African mining engineer Danie G. Krige in his 1951 Master's thesis and formalized by French mathematician Georges Matheron in 1963 (who is known as the "Father of

Geostatistics”) [73] [74]. For a complete treatment of the theory and history, the reader is highly encouraged to read this short 2004 article on Matheron and the history of spatial statistics [74] and this short 1990 article on *The Origins of Kriging* [75]. The reader is also encouraged to examine *Geostatistics for Natural Resources Evaluation*, which presents a rigorous yet accessible treatment of Kriging and all of its variants through a context of the types of data characteristic to the earth sciences, and *Interpolation of Spatial Data, Some Theory for Kriging* which unifies the mathematical basis of Kriging and Gaussian process regression, while situating some of the techniques commonly used by geostatistics practitioners within a broader statistical theory [72].

Independently, various scientific disciplines have found the need for a principled approach to model refinement and have “discovered” variations of the BLUP (eg. *objective analysis* and *optimum interpolation* in meteorology, see [75]). Within the context of optimal design, the technique emerged as a general regression methodology, especially for time-series analysis [76]. With the emergence of various machine learning methods for the analysis of large data sets, so came techniques and frameworks that are well-suited for the operations and the data structures used by computers. [77].

3.2 Gaussian process regression: as a basis to inform sampling

The BLUP has a Bayesian interpretation (see [9] Chapter 3.4.4 for an extended treatment of *Bayesian Kriging*) that becomes especially elegant if we introduce a couple of additional assumptions:

- The random field ε is Gaussian and independent of the random vector β .
- The realization stochastic process $\mathbf{Z} = f(\mathbf{x})$ consists of a collection of random variables $\mathbf{Z} = (Z(\mathbf{x}_1), \dots, Z(\mathbf{x}_n))^T$ which have a *joint Gaussian* distribution.
- This allows both the random term ε and the model term *mathbf{m}* to be fully specified by a mean and a variance, or in the case column vectors, a *mean function* $m(\mathbf{x})$ and a *covariance matrix* $k(\mathbf{x}, \mathbf{x}')$.

With a slight change of notation, let us define these terms:

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})] \\ k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))] \end{aligned} \tag{3.2}$$

If a function can be defined by the composition of these two functions, we can call it a **Gaussian process**, using the following representation:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \quad (3.3)$$

Given the same objective of making a prediction at $Z(\mathbf{x}_0) = f(\mathbf{x}_*)$, we obtain the conditional distribution of this joint Gaussian as:

$$\mathbf{f}_* | \mathbf{f}, \mathbf{x}, \mathbf{x}_* \sim \mathcal{N}(\mu_{f_*}, \Sigma_{f_*})$$

This stems from the definition of the marginalization property of a multivariate Gaussian distribution, and will be revisited in greater detail in the next section. The important thing to note here is that with access to the full *distribution*, we are able understand both the average-case value of this posterior and we have access to the covariances Σ , which specifies the variability of the prediction of the value of f at \mathbf{x}_* . If we are able to evaluate this modeled variance anywhere in our survey area, then we can quantify how this modeled variance changes, depending on what observations are obtained. This can be used to guide a sampling strategy for an environmental modeling task.

3.2.1 Gaussian process regression

Gaussian processes are a collections of random variables, which have useful statistical properties that allows for supervised machine learning. They are non-parametric insofar as they do not explicitly specify a functional relation between the input and the output, and instead describe a space of functions that reflect the observed data.

Let $\mathcal{X} \subset \mathbb{R}^2$ be the environment where measurements are taken. We wish to describe the spatial distribution of an unknown parameter, modeled as a function f that is continuous in this environment: $f: \mathcal{X} \rightarrow \mathbb{R}$. This function f describes our observed data points y_i plus some measurement noise ϵ_i in the following relation: $y_i = f(x_i) + \epsilon_i$, where we assume that this noise follows an i.i.d. Gaussian distribution with zero mean and variance σ_n^2 : $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$. The Gaussian process assumption is to model f as a random probability distribution over a set of functions, and that the value of f for arbitrary inputs x and x' ($f(x)$ and $f(x')$ respectively) has a jointly-Gaussian distribution.

We assume that f is a realization of a Gaussian process, which is completely defined by a mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x}, \mathbf{x}')$ with input vector x : [77]

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \quad (3.4)$$

The joint distribution of observations (the explanatory variable) \mathbf{y} , $\{f(x_1) + \varepsilon_1, \dots, f(x_n) + \varepsilon_n\}$ at a set of inputs \mathbf{X} , $\{x_1, \dots, x_n\}$ and function value (the response variable) at an arbitrary test input x_* , \mathbf{f} , $\{f_*, \dots, f_*^n\}$ can be written as:

$$\begin{pmatrix} f(x_1) \\ \vdots \\ f(x_N) \\ f(x_*) \end{pmatrix} \sim \mathcal{N} \left(\begin{pmatrix} m(x_1) \\ \vdots \\ m(x_N) \\ m(x_*) \end{pmatrix}, \begin{pmatrix} k(x_1, x_1) + \sigma_1^2 & \cdots & k(x_1, x_N) & k(x_1, x_*) \\ \vdots & \ddots & \vdots & \vdots \\ k(x_N, x_1) & \cdots & k(x_N, x_N) & k(x_N, x_*) \\ k(x_*, x_1) & \cdots & k(x_*, x_N) & k(x_*, x_*) \end{pmatrix} \right) \quad (3.5)$$

Typically, it is assumed that the mean function is equal to zero: $m(\mathbf{x}) = \mathbf{0}$. This is for notational simplicity and to make these usually-didactic formulations interpretable. Alternative mean functions include a scalar, linear or nonlinear offset from the mean e.g.: $m(\mathbf{x}) = \mathbf{c}^\top \cdot \mathbf{x}$, where \mathbf{c} is some predetermined vector of numbers.

(3.5) can be re-written in block form as:

$$\begin{bmatrix} \mathbf{y} \\ f(x_*) \end{bmatrix} \sim \mathcal{N} \left(0, \begin{bmatrix} k(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}_n & k(\mathbf{X}, x_*) \\ k(x_*, \mathbf{X}) & k(x_*, x_*) \end{bmatrix} \right) \quad (3.6)$$

where \mathbf{y} is a column vector of scalar outputs y , from a training set \mathcal{D} of n observations, $\mathcal{D} = (\mathbf{X}, \mathbf{y}) = \{(\mathbf{x}_i, y_i) \mid i = 1, \dots, n\}$. k is the covariance function, σ_n^2 is the variance of the observation noise, and input vectors \mathbf{x} and query points x_* of dimension D are aggregated in the $D \times n$ design matrices \mathbf{X} and \mathbf{X}_* respectively.

Through the marginalization of jointly Gaussian distributions, we can derive the following predictive conditional distribution at a single query point $f_* \mid \mathcal{D}, x_* \sim \mathcal{N}(\mathbb{E}[f_*], \mathbb{V}[f_*])$ as [77]:

$$\mu = \mathbb{E}[f_*] = k(x_*, \mathbf{X}) \left[k(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}_n \right]^{-1} \mathbf{y} \quad (3.7)$$

$$\sigma = \mathbb{V}[f_*] = k(x_*, x_*) - k(x_*, \mathbf{X}) \times \left[k(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}_n \right]^{-1} k(\mathbf{X}, x_*) \quad (3.8)$$

where $k(\mathbf{X}, \mathbf{X})$ is a matrix containing the joint prior distribution of covariances of the function f at inputs \mathbf{X} and $k(x_*, \mathbf{X})$ is a matrix containing the covariances between the function at query points and training inputs.

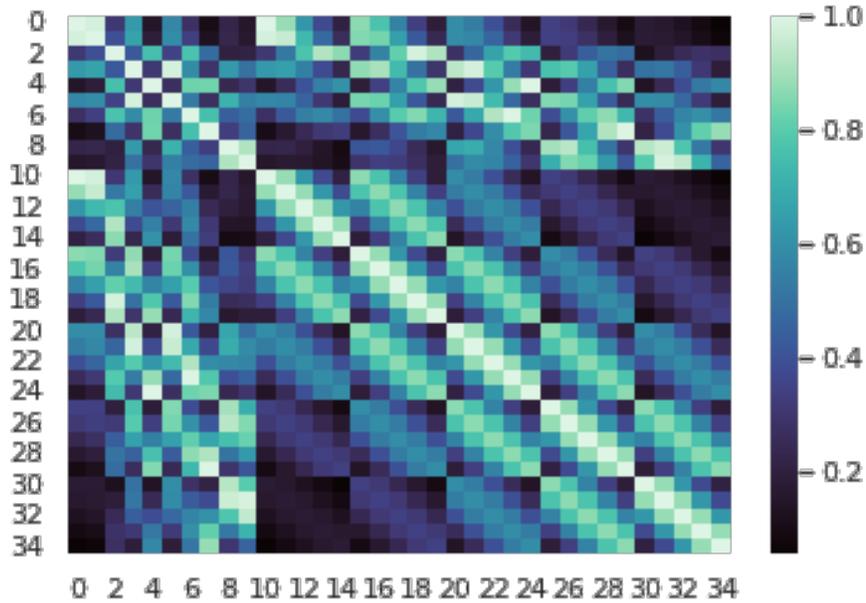


FIGURE 3.2: Visualization of the model posterior covariance matrix. In this example, 10 training points are used to make inference across a 5x5 grid of 25 total test (or query) points. Note that values range from $[0, 1]$.

The covariance function k (or kernel) captures prior knowledge about the function of interest, including properties such as stationarity and smoothness. This can be expressed in both the form of the function and the parameters of the function. In essence, this kernel encodes a prior that can be updated through subsequent observations.

It is important to note that the variance in (3.8) depends only on the inputs \mathbf{X} as opposed to the value of the underlying function \mathbf{y} . This allows us to reason about uncertainty at unobserved locations. If we model a spatial phenomenon as a Gaussian process, this allows us to optimize the locations of where observations are collected (i.e. sampling locations) *before* traveling to collect the observation.

3.2.2 Spatial prior

As mentioned earlier, the kernel k establishes a prior likelihood over the space of functions that can fit observed data in the regression task. Throughout this work, we will use the Matérn 3/2 kernel, which is a finitely-differentiable

function with broad use in the geostatistical literature for modeling physical processes. The Matérn covariance function takes the form [72]:

$$K_{\text{Matern}}(X, X_*) = \sigma^2 \frac{2^{1-v}}{\Gamma(v)} \left(\frac{\sqrt{2v}}{l} r \right)^v K_v \left(\frac{\sqrt{2v}}{l} r \right) \quad (3.9)$$

where K_v is a modified Bessel function, $\Gamma(\cdot)$ is the Gamma function, and r is the Euclidean distance between input points X and X_* . $v > 0$, $l > 0$, and $\sigma^2 > 0$ are hyperparameters representing smoothness, lengthscale, and observation variance respectively. $v = 3/2$ is a shape parameter for the Matérn 3/2 kernel that regulates the smoothness and differentiability of the random process. The kernel function and hyperparameters contain *a priori* knowledge about the distribution of y and a variety of different functions can be substituted for (3.9) depending on the modeled system. See Stein [72] for discussion of why the Matérn kernel is preferred for geostatistics and modeling physical processes. The set of hyperparameters $\theta = \{\sigma^2, \sigma_n^2, l\}$ are set to fixed values in this preliminary work, however they can be trained using various optimization methods to match the properties of environment, however care must be taken to avoid overfitting [77].

3.3 Problem statement: Informative Path Planning

So far, we have explored methods for constructing spatial models of the environment through regression. In particular, we have focused on Gaussian process regression, which is a technique that can be used to both create models of a variable of interest and a measure of the expected uncertainty of the model for the variable of interest. We will now briefly explore how this can be used to guide where a robot should travel for sample collection in an autonomous surveying objective.

This information-gathering task can be formulated as a constrained optimization problem, where information quantity is to be maximized subject to an observation cost. In [78], the task of *informative path planning* (IPP) is specified follows:

$$\mathcal{P}^* = \underset{\mathcal{P} \in \Psi}{\operatorname{argmax}} I(\mathcal{P}) \text{ s.t. } c(\mathcal{P}) \leq B \quad (3.10)$$

where \mathcal{P}^* is an optimal trajectory found in the space of possible trajectories Ψ , for an individual or set of mobile agents such that the cost of executing the trajectory $c(\mathcal{P})$ does not exceed the movement budget, B . $I(\mathcal{P})$ is the information gathered along the trajectory \mathcal{P} and the movement budget can

be any cost that constrains the effort used to collect observations (e.g., fuel, distance, time).

From the perspective of our surveying objective, the goal of IPP is to produce a trajectory for a surveying robot that minimizes the error between the model generated by the collected observations and the true values of the modeled variable.

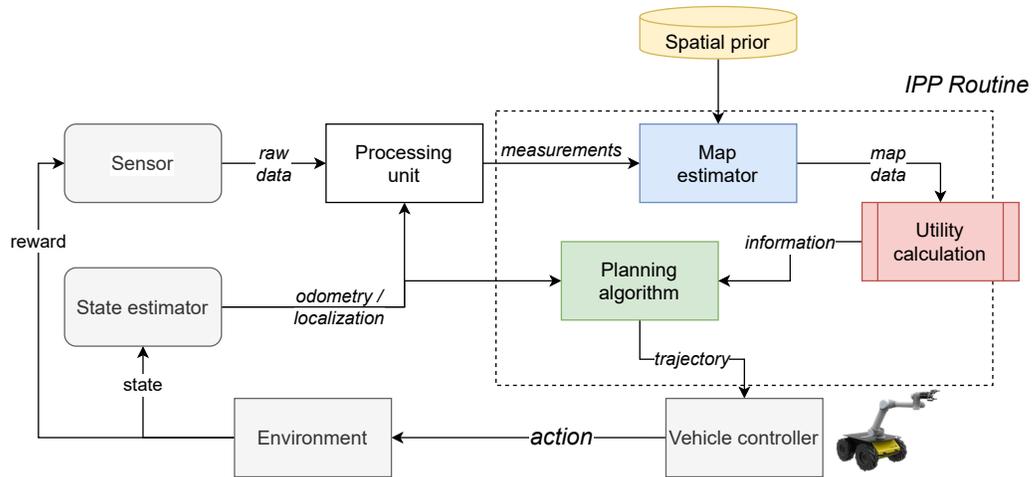


FIGURE 3.3: Schematic overview of a complete surveying system.

3.4 Sampling-based path planning

In order to accomplish the aforementioned objective, we first need a strategy for producing a series of movements for the mobile robot to move from a starting location to a goal destination, while satisfying basic safety criteria such as avoiding collision with known obstacles.

Depending on the context, this task is known as *motion planning* or *path planning*. The most simple applications can be found in the planning algorithms present on toy drones, that simply compute the shortest straight-line trajectory to a goal waypoint, even if a tree happens to be in the way. They can be as sophisticated as the algorithms present in self-driving cars, which consider kinodynamic constraints such as a vehicle's turning radius and motor torque limits, along with environmental constraints such as the traction available to a vehicle's wheels in a given road condition, and safety constraints such as forces experienced by occupants (for comfort) and anticipating the movement of mobile obstacles (other vehicles, pedestrians, and animals).

There are a variety of efficient and optimal path planning algorithms typically used when a navigation system is tasked with finding the shortest obstacle-free path to a destination. However, as discussed in [section 3.3](#), the goal is not to find the *shortest* path to a goal destination, but to find the most *informative* path for sample collection. Arguably, there does not even need to be a particular goal destination in the first place. Therefore, an ideal path planner would generate candidate trajectories for *exploration* that is not necessarily goal-oriented. [Chapter 5](#), explores a goal-oriented objective in that selects among pre-defined waypoints, including an origin waypoint and a destination waypoint. [Chapter 6](#) and [Chapter 7](#) focus on an exploratory objective, that uses a sampling-based path planner as the basis of path planning and we will briefly cover the theory behind the planner in the next section.

3.4.1 Rapidly-exploring Information Gathering (RIG) algorithms

In [chapter 6](#) and [chapter 7](#), the IPP task (3.10) is accomplished with a sampling-based planner with an information-theoretic utility function and convergence criterion. The planner is derived from the family of Rapidly-exploring Information Gathering (RIG) algorithms, which maximize an information quality metric subject to a travel constraint [78]. In turn, RIG inherits the asymptotic cost-optimality of the RRT*, RRG, and PRM* algorithms [79] and a conservative pruning strategy from the branch and bound technique [21]. RRT*, RRG, and PRM* optimize for a cost objective subject to a local movement constraint (avoiding obstacles). RIG instead optimizes for an information

objective, subject to a cost constraint [78]. When not given a goal destination, the RRT algorithm produces a space-filling random tree, with the probability of covering every reachable point in the movement space approaching unity as $t \rightarrow \infty$. When not given a goal destination, the RIG algorithm will similarly add nodes to a space-filling tree until stopped or until there is no free space to expand into. The graph-variant (building upon RRG [80]) is outlined in [Algorithm 1](#). It is worth highlighting a details of this algorithm and a few practical implementation notes:

- Like the RRT-family algorithms, RIG plans in a continuous space. In practice, we model the environment as a grid-world, where the environmental phenomenon is defined on a unit grid. This aligns with how most physical models of the environment are developed. That is, there is some finite simulation resolution for which we subdivide the world into spatially and temporally when computing things like fluid dynamics models. For example, in the atmospheric sciences, “microscale” meteorological phenomena have < 2 km horizontal extent. If the environmental model has a 2 km spatial resolution, then samples would be indexed to the nearest 2-km grid cell, and all nodes in the movement tree could be indexed to the nearest cell. In this scenario, the RRT would expand not $t \rightarrow \infty$, but it would expand until $|V| = |G|$, where $|V|$ is the cardinality of the random tree and $|G|$ is the number of cells in the modeled world where movement is permitted.
- RIG is designed to be run in an “anytime fashion”, where the paths discover approach the optimal path as $t \rightarrow \infty$ in the continuous case, or as $|V| \rightarrow |G|$ in the practical case, with a discretely-defined finite environment. If we are able to specify some optimality threshold, then we can interrupt the algorithm once that condition is reached (see [Equation 3.15](#)).
- A termination criteria can also be set based on proximity to a goal.
- In a manner similar to the “re-wiring” procedure of RRT^* , RIG performs a pruning operation, based on a partial ordering for co-located nodes. During the tree/graph expansion step, a new node is considered to be “co-located” with an existing node if it exists within a step size of the $\text{Near}()$ function. These co-located nodes are ranked according to the partial ordering $n_a > n_b$ defined in [\(3.11\)](#):

$$[H]n_a > n_b \Rightarrow I(p_a^g) + I(n_a) > I(p_b^g) + I(n_b) \quad (3.11)$$

The authors in [78] prove that RIG-Graph is asymptotically optimal if the partial ordering condition holds.

RIG inherits some assumptions of prior sampling-based motion planning literature [78], [80] namely:

1. Robots are modeled using discrete time dynamics
2. A trajectory is deterministic given the environment and control inputs
3. The cost function is strictly positive, monotonically-increasing, bounded and additive.

While different planners consider information functions that are modular, time-varying modular, or submodular, we concentrate on the class of submodular information functions. Submodularity is a property that encapsulates a notion of diminishing returns and is a useful property of functions for optimization problems. In [81] the properties of submodular information functions are discussed at length and in [82] the property is leveraged for a near-optimal sensor placement algorithm using mutual information. For these objective functions, the information gathered along a trajectory is dependent on the prior trajectories. Hollinger and Sukhatme [78] show a sampling-based informative planner is asymptotically optimal for the three aforementioned classes of information functions. From these assumptions, it follows that the cost and information functions are known *a priori*.

Two implementation details remain:

1. How do we compute the information content of a node (what is the function I ?)
2. How do we determine when to stop expanding the RIG graph?

For the answer to question 1, we use an information function that quantifies the value of a new proposed observation at the location of n_{new} during the tree expansion step (L21 in [Algorithm 1](#)). This information function will be covered in detail in the following section. For the answer to question 2, in [chapter 5](#), we explore two approaches, one based on a heuristic stopping criterion developed by the authors in [83] and one based on the expected accuracy of a model constructed from observations at locations proposed by the planner.

3.4.2 Mutual information as a utility function for informative planning

To establish if a trajectory contains informative sampling locations, we employ a utility function that optimizes for a reduction in the posterior variance of the GP

Algorithm 1 Rapidly exploring information gathering graph (RIG-graph)

Input: Step size Δ , Budget B , Workspace \mathcal{X}_{all} , Free space $\mathcal{X}_{\text{free}}$, Environment \mathcal{E} , Start configuration $\mathbf{x}_{\text{start}}$, Near radius R

- 1: \triangleright Initialize cost, information, starting node, node list, edge list, and graph \triangleleft
- 2: $I_{\text{init}} \leftarrow \text{InitialInformation}(\mathbf{x}_{\text{start}}, \mathcal{E}), C_{\text{init}} \leftarrow 0,$
- 3: $n \leftarrow \langle \mathbf{x}_{\text{start}}, C_{\text{init}}, I_{\text{init}} \rangle$
- 4: $V \leftarrow \{n\}, V_{\text{closed}} \leftarrow \emptyset, E \leftarrow \emptyset$
- 5: **while** not terminated **do do**
- 6: \triangleright Sample configuration space of vehicle and find nearest node \triangleleft
- 7: $\mathbf{x}_{\text{samp}} \leftarrow \text{Sample}(\mathcal{X}_{\text{all}}), n_{\text{nearest}} \leftarrow$
- 8: $\text{Nearest}(\mathbf{x}_{\text{samp}}, V \setminus V_{\text{closed}})$
- 9: $\mathbf{x}_{\text{feasible}} \leftarrow \text{Steer}(\mathbf{x}_{n_{\text{nearest}}}, \mathbf{x}_{\text{samp}}, \Delta)$
- 10: \triangleright Find near points to be extended \triangleleft
- 11: $N_{\text{near}} \leftarrow \text{Near}(\mathbf{x}_{\text{feasible}}, V \setminus V_{\text{closed}}, R)$
- 12: **for all** $n_{\text{near}} \in N_{\text{near}}$ **do do**
- 13: \triangleright Extend towards new point \triangleleft
- 14: $\mathbf{x}_{\text{new}} \leftarrow \text{Steer}(\mathbf{x}_{n_{\text{near}}}, \mathbf{x}_{\text{feasible}}, \Delta)$
- 15: **if** $\text{NoCollision}(\mathbf{x}_{n_{\text{near}}}, \mathbf{x}_{\text{new}}, \mathcal{X}_{\text{free}})$ **then**
- 16: $Q \leftarrow \{\mathbf{x}_{\text{new}}\}$
- 17: **while** $Q \neq \emptyset$ **do**
- 18: $\mathbf{x}_q \leftarrow \text{Pop}(Q), N_{\text{qnear}} \leftarrow \text{Near}(\mathbf{x}_q, V \setminus$
- 19: $V_{\text{closed}}, R)$
- 20: **for all** $n_{\text{qnear}} \in N_{\text{qnear}}$ **do do**
- 21: $I_q \leftarrow \text{Information}(I_{n_{\text{qnear}}}, \mathbf{x}_q, \mathcal{E}),$
- 22: $c(\mathbf{x}_q) \leftarrow \text{EvaluateCost}(\mathbf{x}_{n_{\text{qnear}}}, \mathbf{x}_q)$
- 23: $C_q \leftarrow C_{n_{\text{qnear}}} + c(\mathbf{x}_q), n_{\text{added}} \leftarrow \langle \mathbf{x}_q, C_q, I_q \rangle$
- 24: **if** $\text{PRUNE}(n_{\text{added}})$ **then**
- 25: | Delete n_{added}
- 26: **else**
- 27: \triangleright Add edges and node \triangleleft
- 28: $E \leftarrow E \cup \{(n_{\text{qnear}}, n_{\text{added}})\}, V \leftarrow V \cup$
- 29: $\{n_{\text{added}}\}$
- 30: \triangleright Add to closed list if budget exceeded \triangleleft
- 31: **if** $C_q > B$ **then**
- 32: | $V_{\text{closed}} \leftarrow V_{\text{closed}} \cup \{n_{\text{added}}\}$
- 33: **else**
- 34: | $Q \leftarrow Q \cup \{n_{\text{added}}\}$

return $G = (V, E)$

used to model the environment. This follows from considering the information gain of an observation as a reduction of map entropy. In [53], the authors present an approach for quantifying the information content of a map M as its entropy

H and the information content of a new observation Z as the *mutual information* between M and Z , denoted as $I(M; Z)$, defined as follows:

$$I(M; Z) = H(M) - H(M | Z) \quad (3.12)$$

If the map is modeled as a Gaussian Process where each map point (or query point) is a Gaussian random variable, we can approximate mutual entropy with differential entropy. For a Gaussian random vector of dimension n , the differential entropy can be derived as $h(X) = \frac{1}{2} \log((2\pi e)^n |\Sigma|)$. If we let $X \sim \mathcal{N}(\mu_X, \Sigma_X)$ and $X | Z \sim \mathcal{N}(\mu_{X|Z}, \Sigma_{X|Z})$ be the prior and posterior distribution of the random vector X , before and after incorporating observation Z , then the mutual information becomes:

$$I(X; Z) = \frac{1}{2} \left[\log(|\Sigma_X|) - \log(|\Sigma_{X|Z}|) \right] \quad (3.13)$$

where Σ is the full covariance matrix. Again, note that the information content of a new observation depends only on the posterior variance, not the expected value of the random vector. To calculate the information gain, we use the posterior variance of the Gaussian process trained on waypoints in the RIG-Tree graph in a procedure based on Algorithm 5 in [83]. We modify the procedure (Algorithm 2) to consider the posterior variance of the entire modeled environment, where \mathcal{M} contains the set of query (test) points defined in the GP and the set of training points established as waypoints in the RIG-Tree node list. Algorithm 5 details the procedure for updating a node's information content. In lines 6-8, the location of a future measurement z at pose p , is added to the set of past observations (training points) from the entire node graph. This is used to create a new map state containing the previous training points plus the new measurement and the preexisting query points where the GP is evaluated. Next, the posterior variance is calculated (lines 10-14) and the information content of the entire posterior map is updated accordingly.

It is also possible to approximate the mutual information using marginal variances of the Gaussian random variables by leveraging the marginalization property of normal distributions. (3.13) becomes:

$$\hat{I}(X; Z) = \sum_{i=1}^n \frac{1}{2} \left[\log(\sigma_{X_i}) - \log(\sigma_{X_i|Z}) \right] \quad (3.14)$$

which can be calculated from the trace of the GP covariance matrix. See [83] for a derivation and [84] for an example where a similar method is used to approximate Rényi entropy. Borrowing terminology from optimal design we

Algorithm 2 InformationGPVR()**Require:**

Robot pose or desired location p , current map/state estimate \mathcal{M} , covariance function $k(\cdot, \cdot)$, sensor noise σ_n^2 , prior map variance σ , near node information I_{near} ;

- 1: $\bar{\sigma} \leftarrow \sigma$ \triangleright Initialize updated map variance as the current map variance
- 2: **if** I_{near} is not empty **then** \triangleright Initialize information gain
- 3: | $I \leftarrow I_{\text{near}}$
- 4: **else**
- 5: | $I \leftarrow 0$
- 6: $z \leftarrow$ Predict a future measurement at location p and map \mathcal{M}
- 7: \triangleright Construct new map state using z and p \triangleleft
- 8: $\mathcal{M}_{\mathcal{D}} \leftarrow \mathcal{M} \cup \{x_z\}$
- 9: \triangleright Calculate self-covariance and cross-covariance matrices \triangleleft
- 10: $C \leftarrow K(X, X), C_* \leftarrow K(X, X_*) // X$ and $X_* \in \mathcal{M}_{\mathcal{D}}$
- 11: \triangleright Calculate posterior map variance at training and query points \triangleleft
- 12: $c_{**} \leftarrow \text{diag}(K(X_w, X_*))$
- 13: $L \leftarrow \text{Cholesky}(C + \sigma_n^2 I), V \leftarrow L \setminus C_*$
- 14: $v \leftarrow c_{**} - \text{dot}(V, V)^T$ \triangleright dot product
- 15: **for all** $i \in \mathcal{M}_{\mathcal{D}}$ **do**
- 16: | $I \leftarrow I + \log(\sigma^{[i]}) - \log(\bar{\sigma}^{[i]})$
- 17: **return** I (total information gain), $\bar{\sigma}$ (updated map variance)

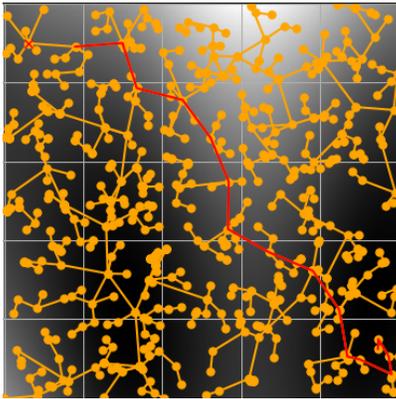
distinguish between the utility formulation that uses the full covariance matrix (d-optimal) and the covariance matrix trace (a-optimal) in the following section.

3.4.3 Comparison of sampling routines

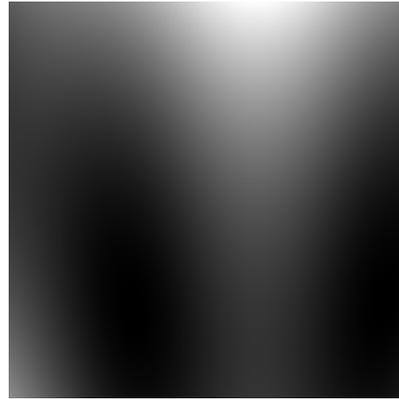
Our objective is to model a continuously-distributed spatial phenomenon, such as the surface of a landscape or a conservative, spatially-distributed process at steady state. As an analogue, we use the Branin-Hoo function, a continuous smooth function of class C^∞ that is among many test functions used for Kriging prediction or other optimization benchmarks [50]. The function is applied over a 500×500 map-unit environment, to comprise the ground truth of a synthetic scalar field.

parameter	value
expand_distance	70.0
path_resolution	20.0
search_radius	200.0
max_iterations	500
model_scale	1:25

TABLE 3.1: RIG parameters used for scalar field experiments. Note that the first three parameters are inherited from RRG/RRT*.



(A) Example graph and path



(B) Branin function scalar field

FIGURE 3.4: Experimental setup for informative planning over a 2D scalar field. The panel on the left shows an example graph and final path selected using the longest-path selection criterion. The right panel shows the underlying scalar field, used as the modeling target.

The spatial model is evaluated using a GP regression model at a 1/25 scale resolution, for a total of 400 query points (20 × 20). Posterior variance is also calculated at the same query points as the final map prediction. The model uses a constant mean function, Matérn ($\nu = 3/2$) covariance function and fixed hyperparameters. Training observations are obtained from a point sensor model, where the a “sample” is obtained by the simulated agent querying the ground-truth scalar field at nodes established in the final trajectory. The RIG planner was based on a RRT implementation by [85] and the GP regression was performed using the `GaussianProcessRegressor` class in scikit-learn [86]. Parameters used by the RIG planner are summarized in Table 3.1

A survey is emulated in three steps: a planning state, where the planner generates a trajectory as a series of waypoints, a sampling stage, where the vehicle collects a point sample of the environmental field at each waypoint, and an inference stage, where a prediction is made for the entire field at a uniform grid of query points. Accuracy of the resulting model is compared to the ground

truth via the Root Mean Squared Error statistic (RMSE), where lower values indicate a more accurate model (see [Table 3.2](#)).

Due to the stochastic nature of RRT-family planners, trajectory statistics are summarized for combinations of different planners, information functions, path selection criteria, and movement budget. $n = 25$ executions of each approach were performed, and the information gain, cost, and final map RMSE for the collection are summarized in [Table 3.2](#). Note that since the trace is not invariant to the scale of the measured parameter, the value of the reward is not directly comparable between the a- and d-optimal functions. From the movement graph returned from the planner, three strategies are compared: a strategy that greedily selects the trajectory with the highest cumulative information gain (greedy), a strategy that selects the trajectory with the highest cumulative path length (longest), and a strategy employed by [\[83\]](#) that balances between exploration and exploitation with a vote-based heuristic. All sampling based planners are allowed to run for 500 sample-iterations for the scenarios with unlimited movement budgets.

approach	size	cost	reward	rmse
gpvr_a (greedy)	8 (2)	390 (127)	2.33 (0.44)	0.264 (0.05)
gpvr_a (longest)	10 (2)	600 (74.1)	1.08 (0.93)	0.235 (0.05)
gpvr_a (vote)	8 (3)	370 (156)	2.01 (1)	0.283 (0.05)
gpvr_d (greedy)	8 (3)	420 (146)	5.46 (1.5)	0.26 (0.06)
gpvr_d (longest)	10 (1)	580 (71.5)	2.1 (1.3)	0.238 (0.05)
gpvr_d (vote)	10 (3)	460 (143)	3.53 (1.4)	0.248 (0.05)

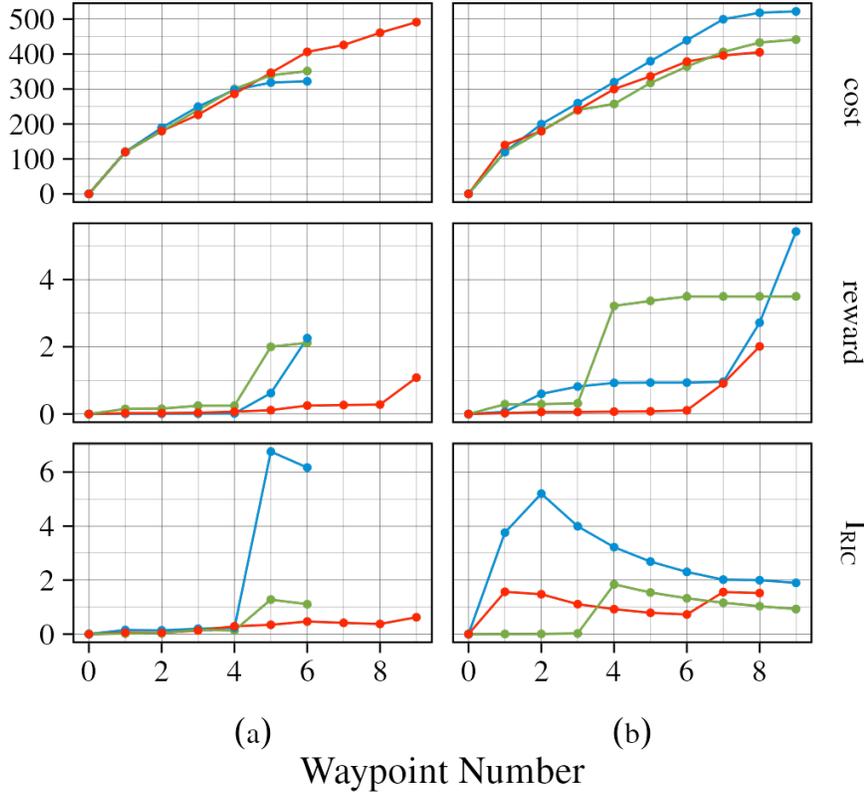
TABLE 3.2: Aggregated ($n = 25$) performance of different planning strategies run without budget constraints, reported as "mean (SD)". Cost and reward are represented as the values for the last waypoint in the final trajectory. Best values are bolded.

Out of 25 repetitions, the execution that is closest to the mean ending reward ([Table 3.2](#)) is displayed in [Figure 3.5](#). While the greedy path selector tended to produce trajectories with the highest cumulative information gain, the overall map accuracy was not as high as the algorithm that selected the longest paths with more observations on average. The goal of the vote-based selection heuristic was to balance between information gain and travel distance and in practice it produced trajectories that were more informative than the longest paths, with a superior final map accuracy.

As another basis for comparison, we calculate the relative information contribution (RIC) ([3.15](#)) criterion that defines the amount of information gain relative to a neighboring vertex in the RIG graph. This was introduced in [\[83\]](#), where it was used as part of a heuristic stopping criterion for information

gathering. In this benchmark demonstration, it is shown in 3.5 as an efficiency metric, to establish the usefulness of acquiring an additional observation.

$$RIC \triangleq \frac{I_{\text{new}}}{I_{\text{near}}} - 1 \quad (3.15)$$



Path Function — greedy — longest — vote

FIGURE 3.5: A comparison of cumulative cost and information gain for the: (a) a-optimal information function and the (b) d-optimal information function along with different path selection strategies. I_{RIC} is the penalized relative information contribution, which expresses a tradeoff between information gain (here, reward) and path length.

In scenarios with a limited movement budget (Table 3.3), trajectories from the informed planner are compared with a naive sampling scheme typical of many environmental surveys—a uniform coverage of sampling locations. The movement budget is defined as the maximum path length (in map units) that the vehicle is allowed to travel for all waypoints in a trajectory. For the coverage planner, this budget is the only varied parameter, as it does not make use of any other information or path selection criteria. For the sampling-based planners, the final trajectory may exceed the movement budget by RRT expansion distance, which is a consequence of how the budget is used by the

RIG planning routine. Specifically, nodes are marked as "closed" when their cost-to-arrive exceeds the movement budget. In every scenario the informed planners produced higher accuracy models than a uniform sampling plan given the same movement budget.

approach	size	cost	reward	rmse
coverage (coverage)	4 (0)	500 (0)	NA	0.272 (0.00)
gpvr_a (vote)	8 (2.1)	460 (133)	1.65 (0.8)	0.249 (0.05)
gpvr_d (vote)	9 (2.2)	490 (140)	3.62 (2)	0.248 (0.04)

TABLE 3.3: Aggregated ($n = 25$) performance of different planning strategies run with a movement budget of 500, reported as "mean (SD)". Cost and reward are represented as the values for the last waypoint in the final trajectory.

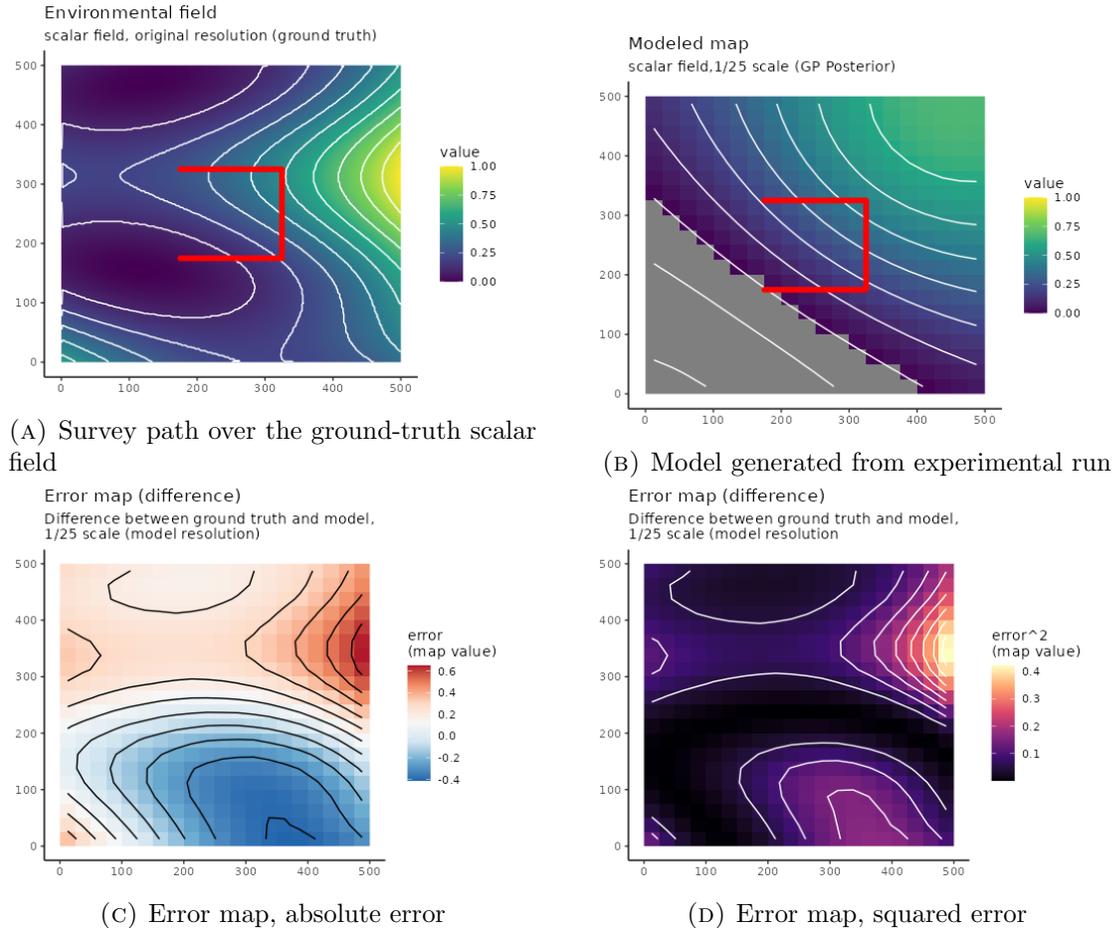


FIGURE 3.6: Example survey generated by the baseline coverage path planner with a survey path length set to 500 map units of distance.

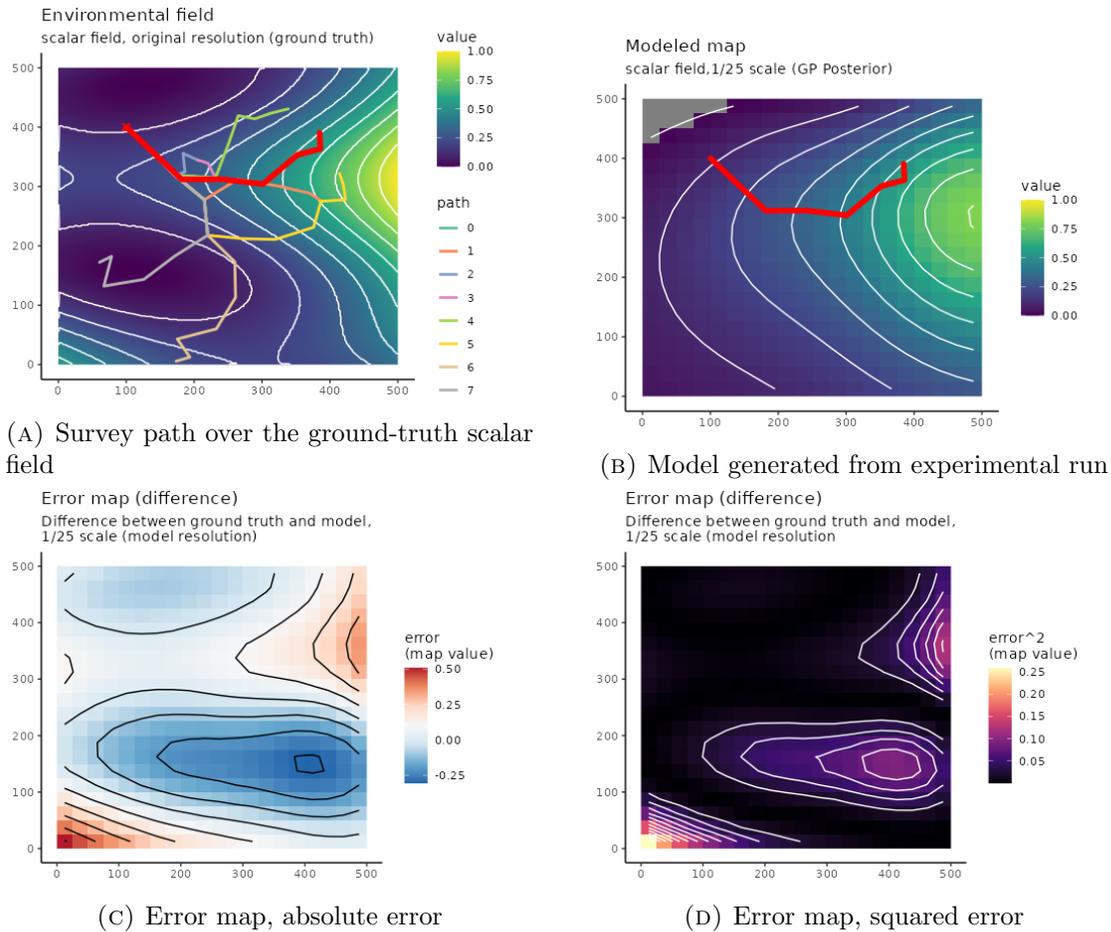


FIGURE 3.7: Example survey generated by informative planner using the a-optimal utility function. The path length of the final path is ≈ 500 map units. The experiment in this figure was one that produced the lowest total model RMSE out of 25 executions.

For an understanding of how well this procedure might work in a real-world scenario, we shall now consider an example where the objective is reconstructing a bathymetric map of a reservoir.

Sonar observations were collected along a manually-defined coverage surveying pattern with a SensePlatypus Lutra differential-propellor unmanned surface vehicle (USV) in Fall of 2019. A navigable perimeter was established from water boundaries obtained from OpenStreetMap, and the planner generated a trajectory within this area [87]. Measurements from the field survey were stored using a kd-tree data structure and training points were obtained by querying the nearest measurement to a sampling location proposed by the planner. Map error was calculated between the posterior field reconstruction and the subset of ground-truth measurements that fell within the rectangular extent of the final planned trajectory. The example pictured in 3 has a RMSE of

70.1 over the entire extent of the lake and 3.6 over the bounding box containing the surveyed path. A large map error outside of this region is to be expected, as the GP quickly regresses to the mean of zero for locations x far away from observations x_j .



FIGURE 3.8: Input dataset for lake experiment, collected with data from UC Merced capstone team in 2019 with a SensePlatypus Lutra USV.

Most of the computational complexity lies in the calculation of the posterior variance of the GP regression, which is $\mathcal{O}(n^3)$ in runtime complexity (for naive Gaussian elimination) due to matrix inversions, where n is the number of training points (L17, Algorithm 2). The a-optimal information function is desirable for its computational efficiency, as it avoids computing the determinant of a matrix of size N , where N is the number of training plus query points. For our d-optimal information function, computing the determinant via LU Decomposition also has a complexity of $\mathcal{O}(n^3)$ [88] [89] (although more efficient approaches exist). For small values of n , this second step can result in a significantly longer computation. In these scenarios, the trace is an attractive approximation of the information content of the posterior model covariance. However, unlike the determinant the trace is not invariant to the scale of the parameters (in this case, the measured values) [90]. Further work can characterize the degree to which this discrepancy is practically meaningful for generating informative sampling trajectories for field reconstruction. For example, in our experiments using the a-optimal information approximation

resulted in generally worse final map predictions, however the difference was small.

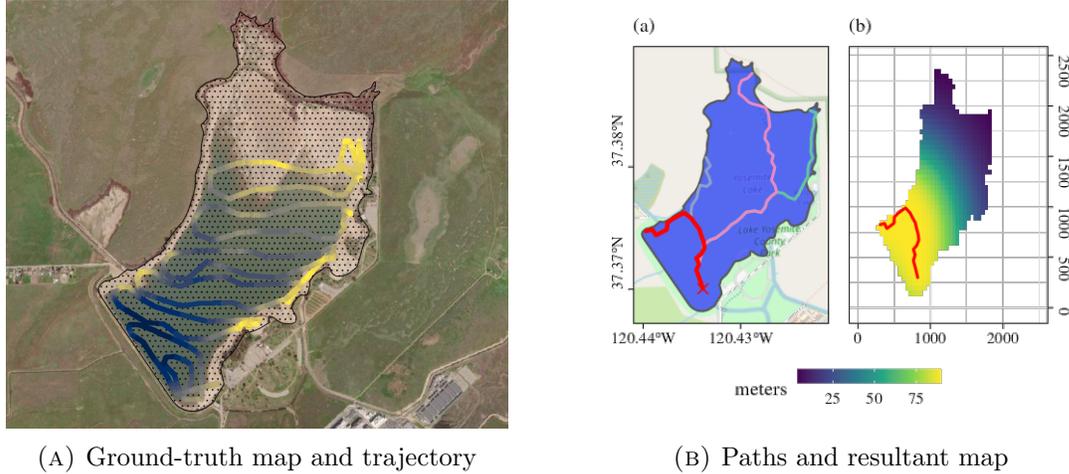


FIGURE 3.9: Right subfigure: An overview of the candidate paths produced by the a-optimal information function, with the highest-ranked final path in red (a) overlain on OpenStreetMap imagery. The resulting GP posterior after executing the trajectory in red is shown (b) with color fill units in meters of elevation from the vertical geodesic datum. All map units are in UTM meters. Note the regression to the zero-mean for locations distant from surveyed depths.

3.5 Conclusion

In this chapter, we reviewed a state-of-the-art approach to informative path planning using a sampling-based planner and we explored how different approximation approaches and path selection criteria affect the accuracy of the final survey product. Due to the improved performance with respect to map accuracy, subsequent chapters will use the vote-based heuristic for path selection.

The main constraint in the practical application of the utility function in [Algorithm 2](#) is the runtime-performance of the posterior variance calculation. To address this, we examined a few approximations and their effect on sample selection and map accuracy. Reducing the complexity of this inference step would allow planning over higher-dimensional belief spaces (including time). This will be explored in [chapter 5](#).

It is also worth considering how the utility function could be applied to a multi-vehicle surveying objectives and whether any modifications would have to be applied to enable distributed coordination between sensing vehicles. This is the topic of the next chapter.

Acknowledgments

Special thanks to Spencer Cole, Javier Chaname, Felix Solano Flores, and Melissa Tanner for their assistance in data collection as part of their senior capstone project. I would also like to thank the UC Merced VICE Lab for equipment use and Christopher Tomaszewski for technical assistance with the SensePlatypus USV over the years.

4

Distributed estimation of scalar fields with implicit coordination

4.1 Background

In this chapter we consider the problem of estimating a scalar field using **a team of collaborating robots**. This problem is motivated by our ongoing research in precision agriculture, where it is often necessary to estimate the spatial distribution of parameters such as soil moisture, nitrates, or carbon dioxide flux that can be modeled as spatially-varying scalars. As with the previous chapter, we model this underlying field, using a Gaussian process (GP) [77], for all of the practical benefits described in [chapter 3](#).

In agricultural applications one is often faced with the problem of estimating quantities over very large domains, therefore the use of multiple robots allows for quicker coverage of the region of interest. In these conditions, robots have to plan their motions with multiple objectives. When an exhaustive, systematic coverage of the entire region is not feasible, it is important to collect samples at the most informative places. It is also necessary to be aware of the limited distance that can be covered before the robot must be refueled or recharged. Therefore robots have to plan paths that will eventually terminate at a designated end point before they run out of energy or fuel. Likewise, it is also important to consider that in real world applications the energy consumed to move between two locations is not deterministically known upfront; rather, it is a random variable whose realization is only known at run-time. For example, a robot may need to take a detour to reach a certain place, or it may move through a muddy area causing wheel slippage, etc. Finally, in rural regions communication infrastructures are often lacking or limited and therefore robots can not assume the availability of broadband communication channels.

With these motivations in mind, in this section we present an algorithm to solve this estimation problem with a team of robots. Coordination between the agents is obtained with minimal information exchange and by leveraging the

mathematical properties of GPs to promote dispersion. The approach is fully distributed and each robot just broadcasts to the rest of the team the limited information consisting of the locations where it has collected data, the value it measured, and its unique identifier—no more than a handful of bytes at a very low frequency. No other communication is required and robots never exchange their individual plans or models. In addition, each robot uses a refinement of our recently developed planner for stochastic orienteering to ensure that it reaches the final location before it runs out energy. Through extensive simulations we will observe that this approach ensures robots collect samples in areas leading to a more accurate reconstructions of the underlying unknown scalar field.

Previous efforts: The problem considered in this chapter is related to the *orienteering* combinatorial optimization problem where one has to plan a path through a weighted graph to gather the maximum sum of vertex rewards while ensuring that the length of the path does not exceed a preassigned travel budget. Recently, we have extensively studied this problem in agricultural settings, both for single agents [28] and multiple agents [91], and we also considered its stochastic variants [39]. In all these works however, rewards associated with vertices were static and assigned upfront, while in this section rewards associated with vertices are iteratively re-estimated based on the gathered samples. Moreover, our former multi-robot solution [91] was centralized, while we here propose a fully distributed approach.

The use of GPs for estimating scalar fields with robots has also been explored in the past. The authors in [59] proposed an algorithm to reduce in minimal time the variance of the scalar field being estimated with a GP. Their solution uses a team of robots, but does not consider travel budgets, i.e., robots can travel as much as needed. Similarly, [92] propose an estimation algorithm for GPs aiming at reducing uncertainty under time and communication constraints. However, their solution does not consider a travel budget.

This section is also related to Informative path planning (IPP) where the emphasis is on planning paths to collect the most informative samples. Examples of additional previous efforts are given in [chapter 2](#). In IPP, however, the set of candidate sample points is not given, but is rather determined by the algorithm, and the travel budget is typically not explicitly considered.

The rest of the chapter is organized as follows. The problem formulation is introduced in [section 4.2](#) and our methods are discussed in [section 4.3](#). In [section 4.4](#) we present extensive simulations to evaluate our proposal and in [section 4.5](#) we draw the conclusions ¹.

¹*This chapter is based on the accepted version of the following published article:* Booth, L., Carpin, S. (2024). Distributed Estimation of Scalar Fields with Implicit Coordination.

4.2 Problem Statement

In this section we consider the problem of estimating a scalar function $f : \mathcal{X} \rightarrow \mathbb{R}$ defined over a bounded region of interest \mathcal{X} given a limited number of observations collected by multiple robots. The goal is to determine where robots should collect these samples. As commonly done in this domain, a graph structure is used to model the navigable environment. We assume that observations of the underlying scalar function can be collected at a limited set of sampling locations, denoted as the set of vertices V in the graph. This assumption holds in a variety of real-world agricultural applications, where specific points of interest (e.g., sentinel trees that serve as early-indicators of ecosystem health) have been pre-identified, or when the robots sense the environment by leveraging pre-deployed infrastructure (such as soil sensors implanted in the ground). This assumption is *not* restrictive; when prior sensing locations are not specified, one can choose V arbitrarily, e.g. as a set of equally-spaced points covering the region of interest, as is typical in *naïve* surveying schemes. We consider this navigable environment as a complete graph, with edge set $E = V \times V$. To each edge $e \in E$ we assign a random variable $c(e)$ representing the movement cost (e.g. energy spent) when the robot traverses the edge. We assume that the density functions characterizing these random variables are known.

All robots must begin at an assigned start vertex v_s and end at an assigned goal vertex v_g before they run out of energy. Each robot r_i starts with a travel budget B_i . When the robot traverses an edge e , its budget decreases by the random value $c(e)$. For simplicity, we assume that all B_i s are the same, but this is not a strict requirement. Each time the robot visits a location $v \in V$, using its onboard sensor(s) it collects a sample of the underlying function f , obtaining a noisy observation $y_v = f(v) + \varepsilon$, where $\varepsilon \sim N(0, \sigma_m^2)$ is measurement noise, Gaussian-distributed with zero mean and variance σ_m .

With regard to communication, we make the following two assumptions: When robots are collecting data, they can only anonymously broadcast packets of the type (v, y_v, n) indicating that they collected observation y_v at vertex v . The last component n is a unique id assigned to each robot—its use will be presented in section [section 4.3](#). At the end of the mission, after the robots have converged at the goal vertex v_g and are in proximity, they can exchange all the data they have gathered during the mission. However, at that point data collection is concluded and they cannot return to the field and acquire more data to improve the estimate. These communication assumptions are consistent with

contemporary technology used by robots in agricultural domain. In particular, LoRa [93] offers the capability of streaming limited amounts of data at long distances and is compatible with the assumptions we made when robots are out in the field. When robots terminate their mission and are in close proximity data can be instead be exchanged using onboard WiFi.

To reconstruct the scalar field we use Gaussian process (GP) regression, as detailed in section [section 4.3](#). Throughout the mission, using the available data (either collected or communicated) robots can make predictions about the value of f at arbitrary locations in \mathcal{X} . We indicate such predictions as \hat{f} . The overall objective is to collect the set of observations providing the most accurate reconstruction of the underlying scalar field. As common in estimation literature, in this section our metric for accuracy is the mean squared error (MSE) defined as

$$MSE = \frac{1}{|\mathcal{X}|} \int_{\mathcal{X}} (f(\psi) - \hat{f}(\psi))^2 d\psi.$$

4.3 Methods

4.3.1 Spatial prior

The modeling approach used in this chapter is Gaussian process regression and was implemented identically to the previous chapter (see [section 3.2](#)).

The kernel k establishes a prior likelihood over the space of functions that can fit observed data in the regression task. Kernel selection and tuning is a key component in GP regression tasks. In machine learning the radial basis function (RBF) kernel is often used. However, in this paper, we use the Matérn kernel with $\nu = 3/2$ which is a finitely-differentiable function. Our choice of this kernel is motivated by its broad use in the geostatistical literature for modeling physical processes [72] like those motivating this research. The Matérn covariance function takes the form:

$$K_{\text{Matern}}(X, X_*) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}}{l} r \right)^\nu K_\nu \left(\frac{\sqrt{2\nu}}{l} r \right) \quad (4.1)$$

where K_ν is a modified Bessel function, $\Gamma(\cdot)$ is the Gamma function, and r is the Euclidean distance between input points X and X_* . The hyperparameters $\nu > 0$, $l > 0$, and $\sigma^2 > 0$ represent smoothness, lengthscale, and observation variance respectively. As common in GP inference, to account for the measurement noise, the kernel we use is the sum of two kernels, namely the Matérn kernel and a

noise term, i.e., the kernel we use is

$$K(X, X_\star) = K_{\text{Matern}}(X, X_\star) + \sigma_n^2 \mathbf{I}$$

where \mathbf{I} is the identity matrix and the term σ_n^2 models the measurement noise σ_m^2 . While we keep ν fixed at $3/2$, the other hyperparameters $\theta = \{\sigma^2, \sigma_n^2, l\}$ can be trained using various optimization methods using the marginal likelihood to match the properties of environment and the sampled data [77]. In particular, the length scale l is related to the lag parameter of the *variogram*, a function used in geostatistics that establishes how quickly the variance increases as a function of separation distance between pairs of observations in space [72]. In the GP kernel, smaller values of l imply that variance quickly grows with distance, while with larger values the variance grows less.

As we will see in the next subsection, by putting constraints on the range of possible values of l one can implicitly encourage dispersion between the robots, thus promoting the collection of samples in different areas of the environment.

4.3.2 Exploration

In this section we present the planning algorithm executed by each robot in the team. No global data structure is shared among the agents, and all the quantities described in the following are local to each robot. Let $G = (V, E)$ be the graph of possible sampling locations and let $D = (v_i, y_i) \ i = 1 \dots n$ the set of collected samples (vertices and values). All robots start from the same start vertex v_s and must end at the same goal vertex v_g . D is initialized as an empty set, but then grows as the robot collects more data or receives data from other agents. Each robot is given a unique numerical identifier n_i , but the robots need not to know how many agents are in the team. At each iteration the robot assigns a reward function to each of the vertices in V , i.e., it computes a function $r : V \rightarrow \mathbb{R}$ assigning a value to each possible sampling location. Different options for the function r will be discussed in the next subsection.

Algorithm 3 SOPCC

```

start vertex  $v, B$ ;
1: Initialize tree  $\mathcal{T}$  with root equal to  $v$ 
2: for  $K$  iterations do
3:    $v_j \leftarrow \text{UCTF}(v)$ 
4:   add  $v_j$  to the tree if not present
5:   for  $S$  iterations do
6:      $\phi \leftarrow \text{SampleTraverseTime}(v, v_j)$ 
7:      $B \leftarrow B - \phi$ 
8:      $path \leftarrow \text{rollout}(v_j, B)$ 
9:     compute  $Q[v_j]$  and  $F[v_j]$  based on the  $S$  paths
10:    Backup( $v_j, Q[v_j], F[v_j]$ )
11: return ActionSelection( $root(\mathcal{T})$ )

```

Once the function r has been computed, the robot is faced with an instance of the stochastic orienteering problem, i.e., it has a graph $G = (V, E)$ with known deterministic rewards r associated to vertices and stochastic costs c associated to edges, as well as a residual budget B . At this point the robot executes the algorithm presented in [39] to solve the stochastic orienteering problem (SOP). Because of the intrinsic computational complexity of the orienteering problem, the SOP algorithm uses a Monte Carlo tree search informed by an heuristic aiming at identifying vertices with high value r , low travel cost, and from which the robot can still reach v_g with high probability (the reader is referred to [39] details). The SOP algorithm returns the next vertex v_a to visit. The robot then moves to v_a , collects an observation y_a , updates D , and broadcasts the packet (v_a, y_a, n_i) to all other agents, where n_i is the unique id of the robot. This process continues until the SOP algorithm returns v_g , in which case the agent moves to the goal vertex v_g and terminates. Throughout the process the robot keeps listening for possible packets broadcast by other agents, and when they are received the location and sampled values are added to D .

Algorithm 4 Adaptive planning with SOPCC and GP-variance

```

given robot identifier  $id$ , observations  $\mathcal{D}$ ;
1: while  $B > 0$  and  $v \neq v_g$  do
2:    $f_* \leftarrow \text{update\_gp}(\mathcal{D})$ 
3:    $v_{next} \leftarrow \text{SOPCC}(v, B, f_*)$ 
4:   move to  $v_{next}$  and measure time spent as  $c$ 
5:    $B \leftarrow B - c$ 
6:    $v \leftarrow v_{next}$ 
7:    $y_v \leftarrow \text{sample\_env}(v)$ 
8:    $\text{broadcast}(v, y_v, id)$ 

```

Algorithm 5 UPDATE_REWARD

```

given GP  $(f(x))$ , current location  $v_c$ , proposed location  $v$ ;
1:  $r(v) = \mathbb{V}(f(v))$ 
2: for all robots,  $j$  do
3:   if  $d(v_j, v) < d(v_c, v)$  then
4:      $r'(v) = r(v) \cdot \frac{d(v_j, v)}{d(v_c, v)}$ 
5: return  $r'(v)$ 

```

As the SOP algorithm was developed for the single robot case, in this work we added a minor modification to account for the presence of multiple robots. The change is as follows: When considering the reward of a vertex $r(v)$, the robot considers for all other agents, the last packet they transmitted (if it exists). Then, if it determines that another agent is closer to v than itself, it discounts the reward $r(v)$. More precisely, let v be the vertex whose utility is being evaluated, and $r(v)$ its reward. Let v_c be the location of the current robot, and assume that it determines that robot j has broadcast a packet indicating it collected a sample at vertex v_j . If v_j is closer to v than v_c , then $r(v)$ is updated as $r'(v) = r(v)d(v_j, v)/d(v_c, v)$ where d is the Euclidean distance between the vertices. The rationale for this update is that if another robot is closer to v , then it is more likely to reach v than the former robot, so the utility of v is decreased for the former robot to prevent having both robots visiting v , as this would be a replicated effort wasting resources. However, the utility is not set to zero because robots do not communicate with each other and do not know their individual intentions. Also, since each robot maintains its own set of GP hyperparameters (see discussion below) and these will be different from each other, robots cannot make absolute predictions about the intentions of other robots in the team.

Remark: one could imagine that after a robot has determined which vertex v it will visit next, it could broadcast this information to other agents so that they do not consider it anymore among their choices. However, this is not done for two reasons. First, such additional communication would almost double the amount of transmitted data, thus going against our effort to keep exchanged information at a minimum. Second, because of the stochastic nature of the environment there is no guarantee that a robot electing to visit a certain vertex will eventually reach it and collect a sample. Hence we opt for the current approach where robots share measured data only after they have reached and sampled a location.

4.3.3 Vertex quality computation

Key to the presented approach is the reward function $r : V \rightarrow \mathbb{R}$ used by the SOP algorithm to decide which vertex to visit next. Ideally, the function should identify *instrumentally good* vertices to visit, where good in this case means vertices that will yield a reduction of the MSE metric. Different metrics have been proposed in literature. One obvious choice is to use Eq. (3.8) to predict the variance of vertices in V and set $r(v) = \sigma^2(v)$. In this case, the objective is to assign high values to vertices with high uncertainty in the estimate. In [94] the authors instead propose to use a linear combination of the mean and standard deviation predicted by Eqs. (3.7) and (3.8). Their approach aims at discovering the extrema of an unknown function. As in our application we are interested in the entire function, and not just its peaks, we could set $r(v) = |\mu(v)| + \beta\sigma(v)$. Finally, in [95] the authors propose an algorithm to compute the mutual information for vertices (prior to and after being added to the movement graph) using predictions for mean and variance. After having implemented these three alternatives, preliminary experiments did not outline significant differences between them. However, setting $r(v) = \sigma^2(v)$ has the advantage of not requiring the tuning of additional parameters, as it is instead necessary for the other two methods. Therefore, informed by these preliminary findings, in our implementation each robot assigns the predicted variance as the value of a vertex. Note that for vertices already in D the algorithm sets $r(v) = 0$, so that robots never consider again vertices that have been already sampled at least once.

The kernel we use to make predictions about the variance depends on three hyperparameters $\theta = \{\sigma^2, \sigma_n^2, l\}$ that can be tuned to best fit the data in D . As pointed out in [77] Ch.5, to obtain better results it is possible to repeat the optimization process multiple times, with random restarts to avoid getting stuck in suboptimal local minima. In our approach, before assigning values to the vertices each robot executes the optimization locally with ten restarts, but never

communicates the hyperparameters of its internal model θ to the other team members. Each agent then operates with its separate set of hyperparameters θ that are unlikely to match the others, due to the random restarts of the optimizer. This difference will further decrease the likelihood that multiple agents will select the same vertices to sample, because even with identical sets \mathcal{D} the variance predicted by the GP will be different. However, during the optimization process each agent uses the same lower bound l_0 for the length scale l . This choice encourages robots to disperse because the variance of vertices in V near to vertices already inserted in D is lower than the variance of vertices far from D and thereby the reward associated to vertices near to already sampled locations is lower.

4.4 Experimental Evaluation and Discussion

To assess merits and limitations of the proposed approach, we perform simulations on test cases while varying the different parameters related to the planning and surveying objectives. Due to the limitation of space, we examine the task of reconstructing two scalar fields. The first is a synthetic scalar field with a periodic trend depicted in [Figure 4.1](#). The second, displayed in [Figure 4.1](#), shows the soil moisture distribution measured in Summer 2018 in a commercial vineyard located Central California. This second scalar field was used as benchmark in previous publications [29]. To ease the comparisons between the two cases, both fields were rescaled to the same size, although the amplitude of the underlying values are different. GP predictions of each respective field were made with a Matérn kernel with $\nu = 3/2$. It should be noted that this kernel is commonly used in geostatistical applications and is more appropriate for the soil moisture dataset. Here, the periodic synthetic field serves as a pathological example, with a mismatched spatial prior. In fact, the use of periodic kernels could lead to better results for the synthetic field. Future work will examine online adaptive kernel selection through Bayesian optimization.

Our algorithm, indicated as *Coord* in the following discussion, is compared with two baseline alternatives:

- The random waypoint selection algorithm (RWP), which selects the next vertex to visit at random among those still to be visited. Due to the nature of the selection process, the ability to communicate during the sampling process is immaterial. The RWP algorithm is often considered as a baseline comparison in this type of tasks (see e.g., [96]).
- A non-coordinated (NC) approach, which selects the next sampling point using the same criteria used by our proposed algorithm, but does not

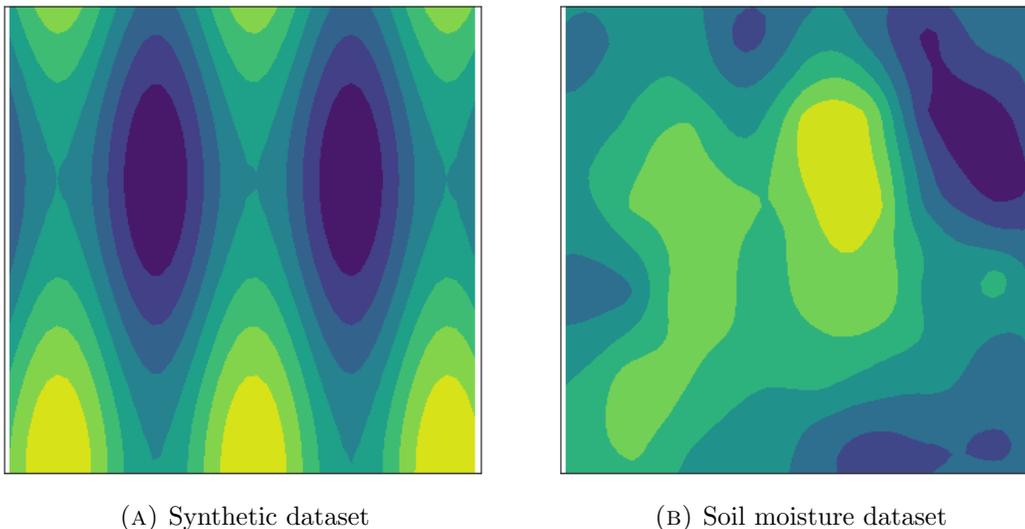


FIGURE 4.1: Benchmark scalar fields to be estimated. With reference to the travel budget B , the length of the side edge is 5. In both instances the start vertex v_s is in the lower left corner and the goal vertex v_g is in the top right corner.

exchange any information during the sample process, i.e., during the selection process each agent only considers the samples it collected, but not those collected by the other agents.

At the end of the mission, when all robots have reached v_g , both RWP and NC share all collected sensor observations and the MSE is computed after fitting the GP using all data collected by all robots. This step is not necessary for Coord because data is exchanged on the go, but it ensures that the MSE evaluation is done fairly among the three algorithms. After the algorithm has selected the next point to visit, all algorithms, including Coord, use the same planning engine. Finally, both NC and Coord do refit of the GP and update the parameters θ before computing r , while RWP does not do this step because it does not use the current estimate for the selection of the next point to visit.

All procedures were executed single-threaded in Python running on an Apple M1 processor. All computations related to GP fitting and processing use the scikit-learn library [86]. For both scalar fields considered in the tests, we varied the number of agents (3,5,7,9), the budget B (10,15,20), and the parameter l_0 (0.1,0.5,1). For each combination of parameters, twenty independent simulations were performed, for a total of about 3,500 executions.

Figure 4.3 show the average MSE as a function of the number robots for all algorithms. As expected, the trend is decreasing (i.e. improved prediction accuracy) with diminishing returns as the number of robot grows. We can observe that the proposed algorithm outperforms the others. Note that the range

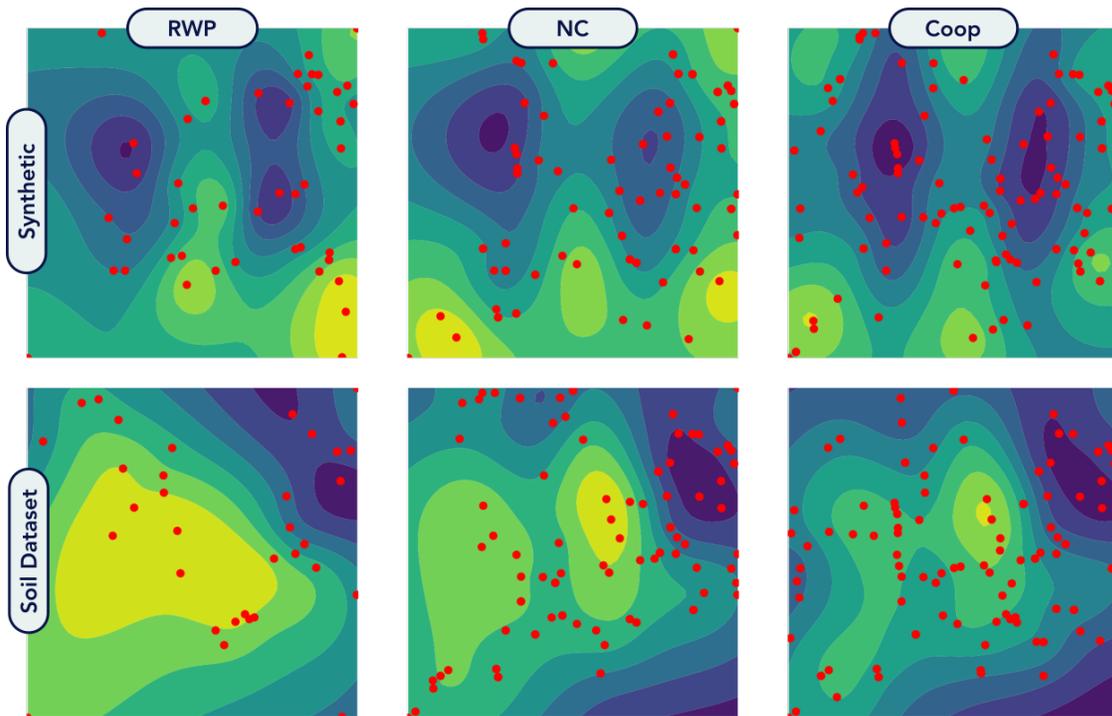


FIGURE 4.2: Examples of reconstructed scalar fields obtained with a combination of different planners and scenarios. All experiments shown in this figure were run with $n = 5$ robots with a path budget of $B = 20$. Algorithms compared are: Random-waypoint (RWP), non-coordinated planning based on variance reduction (NC) and coordinated planning based on variance reduction with shared knowledge of robot positions (Coord).

of values for MSE in the two test cases is different because of the different values in the underlying scalar fields.

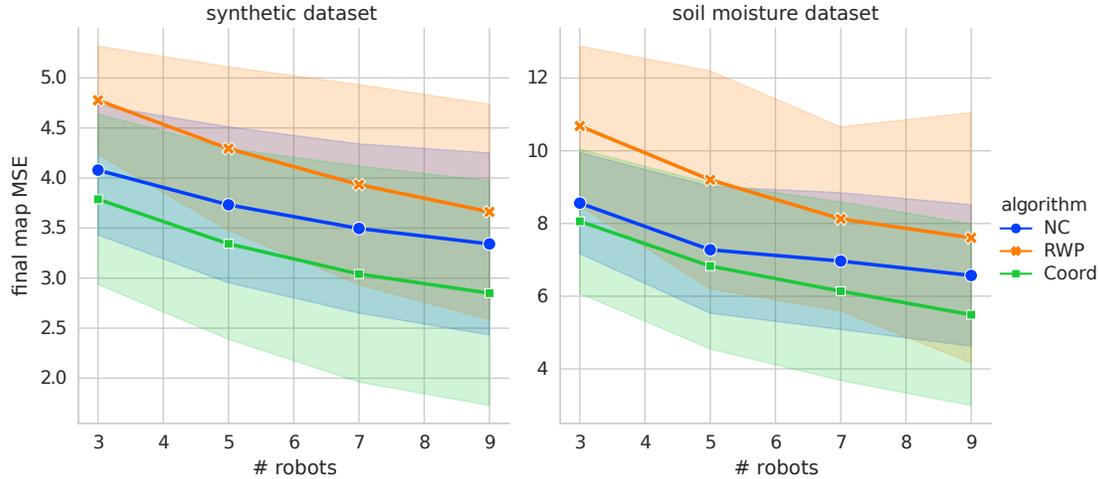


FIGURE 4.3: Average final map MSE for $n = 20$ trials per algorithm. Error bars show \pm one standard deviation.

Next, in [Figure 4.2](#) we show the reconstructed scalar field for the three algorithms with a budget of 20 and 5 robots. The red dots show the locations where the samples were collected. Due to the random selection process, RWP ends up collecting less samples before exhausting the budget and this leads to an inferior estimate. NC and Coord, instead, collect more samples, but we can see how Coord spreads them more uniformly and ultimately leads to a more accurate estimate (see [Figure 4.1](#) for the ground truth). Similar results are observed for the synthetic map, but are not displayed for lack of space. [Table 4.1](#) provides a more detailed numeric analysis of the performance of the three algorithms. Specifically, we look at the number of unvisited locations as well as the number of locations visited by more than one robot. These are two proxies for the MSE metric, and lower values are desired in both cases. When the travel budget is 10, the number of unvisited locations is similar for the three algorithms because with limited budget the set of available choices before the budget is used is limited. As the budget grows, we see that the Coord algorithm emerges as the algorithm with less unvisited vertices, thus substantiating the claim that agents spread in the environment in a more coordinated fashion. For the number of revisited locations, RWP (as expected) always has the lowest number of revisited locations, due to the completely random nature of the selection. However, when comparing NC with Coord we see that the latter has always a lower number, again showing that the agents better spread in the

map	synthetic dataset			soil moisture dataset		
budget	10	15	20	10	15	20
RWP	179/6	164/8	155/10	179/6	164/9	155/10
NC	164/17	139/31	122/42	166/16	140/30	122/40
Coord	163/12	129/18	104/21	166/11	133/17	106/21

TABLE 4.1: Average number of unvisited and re-visited waypoints, from an experimental setting of 200 candidate sampling locations. X/Y means that there were on average X unvisited vertices and Y revisited vertices.

environment avoiding to revisit the same places, thus ensuring that coordination leads to a better use of the robots are mobile sensors.

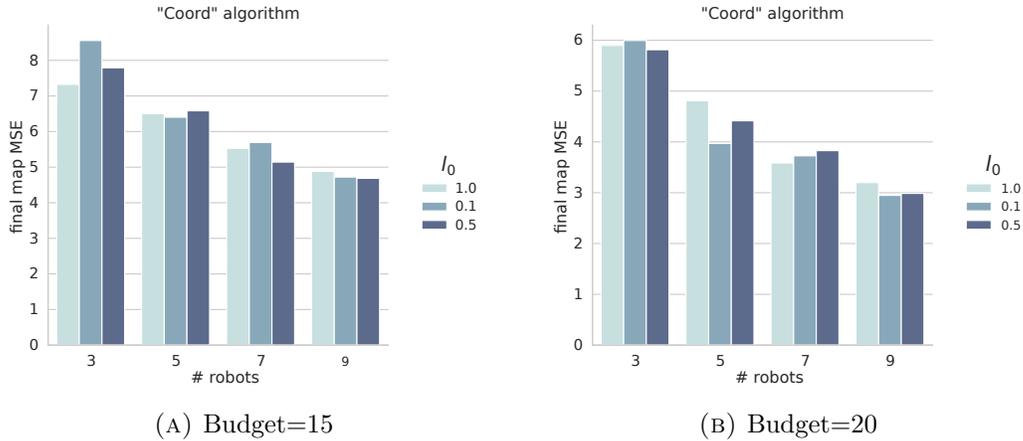


FIGURE 4.4: Average MSE for different values of l_0 and number of robots.

Finally, in [Figure 4.4](#) we display how the choice of l_0 , the lower-bound for the length scale parameter l , impact the value of the MSE metric. The two panels correspond to a budget of 15 and 20 respectively, and group the results for different numbers of surveying robots. For 9 robots the impact is marginal, and this is explained by the fact that with this many agents the team manages to cover most of the environment during the mission. However, for budget of 15 and 3 robots, a value of $l_0 = 1$ gives a clearly better result. Likewise, for budget of 20 and 5 robots, a value of $l_0 = 0.5$ is best. These results show that by tuning l_0 it is possible to implicitly promote better dispersion in the team and then lower values for the MSE. An outstanding question to be investigated is how to select this value in a general setting. Nevertheless, these results confirm our hypothesis that by constraining the GP kernel parameters being optimized, one can enforce different behaviors on the team members.

4.5 Conclusion

This chapter presented an approach to reconstruct a scalar field using a team of robots performing distributed GP estimation. By exploiting the underlying properties of GPs, robots implicitly manage to disperse thorough the domain and collect samples at locations leading to a more accurate estimation. Furthermore, this is all enabled through limited communication, providing a feasible pathway for multi-robot coordination through limited-bandwidth communication protocols such as LoRa.

Similar to the premise of optimal sensor placement explored in [chapter 2](#), this problem framing is well-suited toward agricultural environments. For example, farm managers and agronomists may choose to closely monitor the presence of plant pests and pathogens in selected “sentinel trees” [\[97\]](#) as a proxy for a wider area of interest. In farms without the infrastructure for sensor networks, a mobile robot may also be used to visit and “read-out” the value of non-networked sensors. The remaining chapters will explore methods to model and reconstruct time-varying environmental phenomena with a single robot ([chapter 5](#)) and with teams of robots ([chapter 6](#)).

5

Informative path planning for scalar dynamic reconstruction using coregionalized Gaussian processes and a spatiotemporal kernel

5.1 Introduction

In this chapter, we turn our efforts toward monitoring and modeling **spatiotemporal processes**. The emergence of small, inexpensive mobile platforms points to a future where mobile sensors will be rapidly dispatched to model a dynamic phenomenon. However, to the best of our knowledge there have been limited investigations of informative planners that consider the *temporal dimension* of information content, especially in an online planning approach. This is necessary to produce faithful representations of dynamic environments, as observations made early in the course of a survey may no longer represent the state of the system at the location at the end of the survey. Additionally, it may be desirable to infer the state of the system at arbitrary points in time, or into the future.

To address this issue, we propose a novel sampling-based IPP framework that considers the information content of sensing locations in space and time. An overview of the framework is shown in [Figure 5.1](#) and in the accompanying video. Inspired by the asymptotic optimality of IPP methods based on random trees [78] [83] and advancements in large-scale, multiple-output Gaussian process modeling [98], our method combines an information-theoretic sampling-based planner with a spatiotemporal covariance function implemented as a separable kernel to access the information gain from the locations of candidate sensing locations both in space *and* time. This also allows for both inference of the state and inference of model uncertainty for unexplored

parts of the system and establishes a criterion for revisiting already-observed locations that no longer meaningfully reduce uncertainty of the system's current state.

The contributions of this chapter are:

- A framework for reasoning about the information content of observations in arbitrary dimensions reconciled to a metric appropriate for path planning
- The integration of this spatiotemporal information function in a novel time-aware informative planner for terrestrial monitoring
- Validation of the approach in the context of spatial and temporal priors with simulated and real-world dynamic scenarios inspired by common environmental dispersion processes
- Exploration of interactions between the parameters governing the planner and the model

Previous efforts: As discussed in [chapter 2](#), this section draws from a rich body of literature, surrounding the task of collecting observations by an autonomous agent for modeling the distribution of a variable of interest in the environment.

Most IPP approaches consider the spatial phenomenon to be static or at steady-state, or they assume that the phenomenon does not change meaningfully during the duration of the survey and there has been a limited number of efforts devoted to planning for time-varying spatial phenomena [38].

As shown in [chapter 3](#), the asymptotic optimality of rapidly-exploring random trees (RRT) has been leveraged to solve IPP tasks in a computationally tractable manner, including exploration applications where the robot is tasked with monitoring an unknown parameter of interest [80]. Rapidly-exploring information gathering (RIG) algorithms approach the IPP task using incremental sampling with branch and bound optimization [78]. This chapter builds on [83], which extended RIG with an information-theoretic utility function and a related stopping criterion.

The remainder of this chapter is organized as follows: The problem formulation is introduced in [section 5.2](#) and our methods are discussed in [section 5.3](#). In [section 5.4](#) we experimentally evaluate our proposal and conclude in [section 5.5](#)¹.

¹*This chapter is based on the accepted version of the following published article: L. Booth and S. Carpin, "Informative Path Planning for Scalar Dynamic Reconstruction Using Coregionalized Gaussian Processes and a Spatiotemporal Kernel," 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 2023, pp. 8112-8119, doi: 10.1109/IROS55552.2023.10341858*

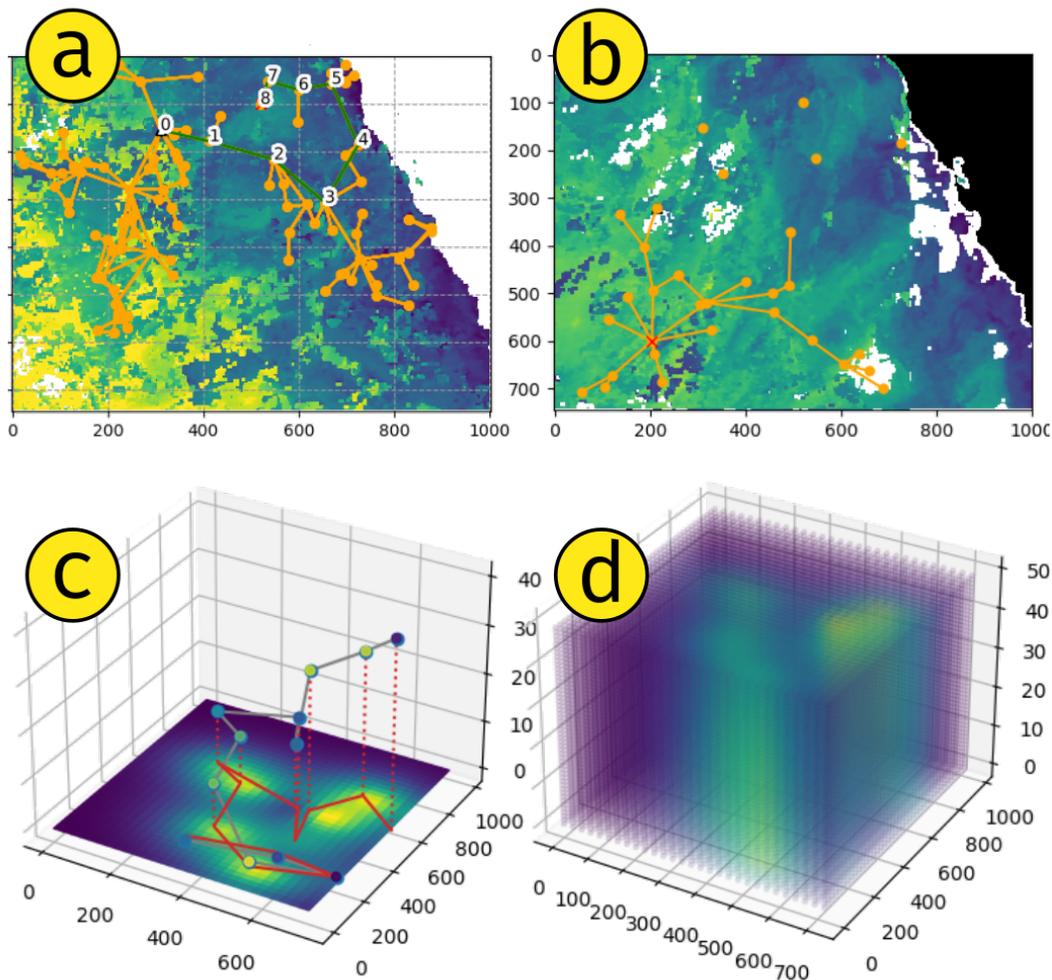


FIGURE 5.1: An overview of our evaluation methodology. (a) shows the ground truth, and the vehicle in the replanning stage, with observation history enumerated. (b) shows the environment during the planning stage with the locations of previous observations. (c) Samples can be visualized along a path in a temporal dimension and (d) displays the final map estimate at all inducing points in the Gaussian process.

5.2 Problem Formulation

In this work, we consider the problem of reconstructing a dynamic scalar field given a limited number of observations, collected along a path. Paths are generated using a receding-horizon approach, alternating between planning and execution of the plan until the traveled distance exceeds the budget B or a prediction window t_{max} . The task can be formulated as a constrained optimization problem, where information quantity is to be maximized subject to an observation cost. In [78], the task is specified follows:

$$\mathcal{P}^* = \operatorname{argmax}_{\mathcal{P} \in \Psi} I(\mathcal{P}) \text{ s.t. } c(\mathcal{P}) \leq B \quad (5.1)$$

where \mathcal{P}^* is an optimal trajectory found in the space of possible trajectories Ψ , for an individual or set of mobile agents such that the cost of executing the trajectory $c(\mathcal{P})$ does not exceed an assigned motion budget, B . $I(\mathcal{P})$ is the information gathered along the trajectory \mathcal{P} , and the movement budget can be any cost that constrains the effort used to collect observations (e.g., fuel, distance, time, etc.)

This paper inherits the assumptions of the original RIG formulation and of prior sampling-based motion planning literature (see [subsection 3.4.1](#)) [78], [80] and adds the following assumptions with respect to time:

1. The state of the robots and the environment are modeled using discrete time dynamics
2. Movement of the sampling agent is anisotropic in the time dimension (see: [section 5.4](#))

To quantify the information content of a trajectory, we employ a utility function that optimizes for a reduction in the posterior variance of the GP used to model the environment. This follows from framing the information gain of an observation as a reduction of map entropy or uncertainty. In [53], the authors present an approach for quantifying the information content of a map M as its entropy H and the information content of a new observation Z as the *mutual information* between M and Z , denoted as $I(M; Z)$ and defined as follows:

$$I(M; Z) = H(M) - H(M | Z) \quad (5.2)$$

We take advantage of the submodularity of mutual information; that is, the information gained by adding an observation to a smaller set is more useful than adding the same observation to a larger (super-) set (See [81] for an analysis of the benefit of submodular information functions for informative sensing applications and [82] for the submodularity of mutual information.)

From the perspective of the environmental modeling task, a useful survey is one that produces the most accurate representation of the environment, minimizing the expected error given field observations. This follows from equations [Equation 5.1](#) and [Equation 5.2](#). This assumption holds when the model is *well-calibrated* with respect to the priors embodied in the model parameters ². Our approach can be extended to an *adaptive planning* scenario,

²Refer to [section 5.4](#) and [Figure 5.3](#) for discussion of the consequences when this assumption does not hold

where model hyperparameters are updated based on new measurements and future path plans leverage the updated model. In previous work, we have demonstrated how model priors can encode modeler intuition, resulting in sampling strategies that vary in the degree of exploration [99].

5.3 Methods

5.3.1 Environmental Model

Let's now briefly review the environmental model: We describe the spatial distribution of an unknown stochastic, *dynamic* environmental process occurring in a region $\xi \subset \mathbb{R}^2$ as a function $f: \mathcal{X} \rightarrow \mathbb{R}$ that is sampled and modeled at the discrete grid, $\mathcal{X} \subset \mathbb{R}^{N_t \times N_{x,y}}$. Here $N_{x,y}$ is a discretization of the spatial domain ξ , while N_t is the temporal domain in which the spatial process evolves.

As in previous chapters, the environmental map comprises this function f that describes our observations y_i , plus some additive measurement noise ε_i , i.e., $y_i = f(x_i) + \varepsilon_i$, where we assume that this noise follows an i.i.d. Gaussian distribution with zero mean and variance σ_n^2 : $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$. We assume that f is a realization of a Gaussian process, represented as a probability distribution over a space of functions. Through marginalization, we can obtain the conditional density $f \mid y = \mathcal{N}(\mu_{f|y}, \Sigma_{f|y})$. The joint distribution of observations \mathbf{y} , $\{f(x_1) + \varepsilon_1, \dots, f(x_n) + \varepsilon_n\}$ and predictions \mathbf{f} , $\{f_*, \dots, f_{*n}\}$ at indices \mathbf{X}_i, \mathbf{t} , $\{x_{1,1}^{(st)}, \dots, x_{m,n}^{(st)}\}$ becomes:

$$\begin{bmatrix} \mathbf{y} \\ f(x_*) \end{bmatrix} \sim \mathcal{N} \left(0, \begin{bmatrix} [cc]k(\mathbf{X}, \mathbf{X}) + \sigma^2 I_N & k(\mathbf{X}, x_*) \\ k(x_*, \mathbf{X}) & k(x_*, x_*) \end{bmatrix} \right) \quad (5.3)$$

where s and t denote spatial and temporal indices respectively. Here, environmental observations y , are drawn from a training set \mathcal{D} of n observations, $\mathcal{D} = (X, \mathbf{y}) = \{(\mathbf{x}_{i,t}, y_{i,t}) \mid i = 1, \dots, n\}$. k is the covariance function (or kernel), σ_n^2 is the variance of the observation noise, and input vectors \mathbf{x} and query points \mathbf{x}_* of dimension D , are aggregated in the $D \times n$ design matrices X and X_* respectively. From the Gaussian process, we can obtain estimations of both the expected value of the environmental field and the variance of each prediction. Noteworthy is the posterior variance, which takes the form:

$$\sigma = \mathbb{V}[f_*] = k(x_*, x_*) - k(x_*, \mathbf{X}) \times \left[k(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}_n \right]^{-1} k(\mathbf{X}, \mathbf{x}_*) \quad (5.4)$$

The differential entropy of a Gaussian random variable is a monotonic function of its variance, and can be used to derive the information content of a proposed measurement. We will show how this can be used to approximate information gain (Equation 5.2) in subsection 5.3.4.

It is important to note that for fixed kernels the variance does not depend on the value of the observation, allowing us to reason about the effectiveness of a proposed observation before traveling to the sampling location [13]. Also notable is the kernel k which establishes a prior over the covariance of any pair of observations. Separate priors can be established in spatial or temporal dimensions, leading to the opportunity to incorporate spatial and/or temporal domain knowledge into the planning process.

5.3.2 Spatiotemporal prior

The modeling effort can be framed as a multi-task (or multi-output) prediction of correlated temporal processes at each spatial discretization $N_{x,y}$. As we only have a finite set of sampling vehicles (one, in fact), we cannot observe all of the spatial "outputs" for a given time, however we can establish a basis upon which they can be correlated [100]. Specifically, the Linear Model of Coregionalization (LMC) has been applied to GP regression where p outputs are expressed as linear combinations of independent random vector-valued functions $f : \mathcal{T} \rightarrow \mathbb{R}^p$. If these input functions are GPs, it follows that the resulting model will also be a GP [101]. The multi-output GP (MOGP) can be described by a vector-valued mean function and a matrix-valued covariance function (see Equation 5.4). A practical limitation of MOGPs has been their computational complexity. For making p predictions with n input observations $y(t_1), \dots, y(t_n) \in \mathbb{R}^p$, the complexity of inference is $\mathcal{O}(n^3 p^3)$ in time and $\mathcal{O}(n^2 p^2)$ in memory [102]. A variety of strategies exist to solve lighter, equivalent inference tasks under simplifying assumptions, such as expressing an output from linear combinations of latent functions that share the same covariance function, but are sampled independently [101]. Since our information function is only dependent on the posterior covariance, we can take advantage fast approximations with complexity $\mathcal{O}(k(n + p \log p))$ (see discussion in subsection 5.3.4).

As mentioned earlier, the kernel k establishes a prior likelihood over the space of functions that can fit observed data in the regression task. For the regression of discretely-indexed spatiotemporal data, where space is indexed by s (eg. latitude/longitude) and time is indexed by t (eg. seconds), we build a composite kernel by multiplying a spatial and temporal kernel:

$$k((s, t), t(s', t')) = k_s(s, s')k_t(t, t') \quad (5.5)$$

While other approaches to kernel composition are possible and encode different environmental priors, constructing a kernel that is separable along input dimensions affords considerable computational advantages. More generally, when $k(\mathbf{x}, \mathbf{x}') = \prod_{d=1}^D k^{(d)}(\mathbf{x}^{(d)}, \mathbf{x}'^{(d)})$, the kernel (Gram) matrix K can be decomposed into smaller matrices $K = K_1 \otimes \cdots \otimes K_D$ which can be computed in $\mathcal{O}(Dn^{\frac{D+1}{D}})$ time (see [103] and [104] for more on kernel composition for multidimensional regression.)

For the spatial relation, we use the Matérn kernel with $\nu = 3/2$ and fixed hyperparameters. Comprehensively described in [72], the Matérn kernel is a finitely-differentiable function with broad use in the geostatistical literature for modeling physical processes due in part to its ability to resist over-smoothing natural phenomena with sharp discontinuities. It takes the form:

$$K_{\text{Matern}}(X, X_*) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}}{l} r \right)^\nu K_\nu \left(\frac{\sqrt{2\nu}}{l} r \right) \quad (5.6)$$

where K_ν is a modified Bessel function, $\Gamma(\cdot)$ is the Gamma function, and r is the Euclidean distance between input points X and X_* . $\nu > 0$, $l > 0$, and $\sigma^2 > 0$ are hyperparameters representing smoothness, lengthscale, and observation variance respectively. We use a radial basis function kernel (RBF or squared-exponential) in the time dimension to smoothly capture diffusive properties that may fade in time. Note that the Matérn kernel approaches the RBF as $\nu \rightarrow \infty$.

5.3.3 Informative Planning

In this work, we present a novel planner IIG-ST to address IPP task defined in equation Equation 5.1. Our planner is built upon IIG-Tree, a sampling-based planner with an information-theoretic utility function and convergence criterion [83] and derived from the family of Rapidly-exploring Information Gathering (RIG) algorithms introduced by Hollinger and Sukhatme [78]. RIG inherits the asymptotic cost-optimality of the RRT*, RRG, and PRM* algorithms [79], a conservative pruning strategy from the branch and bound technique [21], and an information-theoretic convergence criterion (see discussion in subsection 5.3.5). We add routines to consider the time dimension of samples in the tree and combine it with a hybrid covariance function and stopping criterion grounded in map accuracy.

5.3.4 Information Functions

From equation [Equation 5.2](#), we established information gain as the reduction of map entropy H given a new observation Z .

If the map is modeled as a Gaussian Process where each map point (or query point) is a Gaussian random variable, we can approximate mutual entropy with differential entropy. For a Gaussian random vector of dimension n , the differential entropy can be derived as $h(X) = \frac{1}{2} \log((2\pi e)^n |\Sigma|)$. If we let $X \sim \mathcal{N}(\mu_X, \Sigma_X)$ and $X | Z \sim \mathcal{N}(\mu_{X|Z}, \Sigma_{X|Z})$ be the prior and posterior distribution of the random vector X , before and after incorporating observation Z , then the mutual information becomes:

$$I(X; Z) = \frac{1}{2} \left[\log(|\Sigma_X|) - \log(|\Sigma_{X|Z}|) \right] \quad (5.7)$$

where Σ is the full covariance matrix.

For a random vector $\mathbf{X} = (X_1, \dots, X_n)$ with covariance matrix \mathbf{K} , the mutual information between \mathbf{X} and observations Z can be approximated from equation [Equation 5.7](#) as:

$$\hat{I}(X; Z) = \sum_{i=1}^n \frac{1}{2} \left[\log(\sigma_{X_i}) - \log(\sigma_{X_i|Z}) \right] \quad (5.8)$$

Using marginalization, for every X_i , it holds that $\mathbb{V}[X_i] = K^{[i,i]}$. The expression becomes:

$$\hat{I}^{[i]}(X_i; Z) = \frac{1}{2} \left[\log(\sigma_{X_i}) - \log(\sigma_{X_i|Z}) \right] \quad (5.9)$$

and can be computed as the sum of marginal variances at i : $\hat{I}(X; Z) = \sum_{i=1}^n \hat{I}^{[i]}(X_i; Z)$ (see [\[83\]](#) for a derivation).

The main motivation of using marginal variances at evaluation points (Equation [Equation 5.8](#)) is to avoid maintaining and updating (inverting) the full covariance matrix. This is of a particular concern for spatiotemporal modeling, because the number of inducing points grows on the order of $m \times n$ for a spatial domain of m rows and n columns. Alternate GP formulations such as spatio-temporal sparse variational GPs (ST-SVGP) allow for computational scaling that is linear in the number of time steps [\[98\]](#) For computing the posterior variance at GP inducing points, we use LOVE (LanczOs Variance Estimates), for a fast, constant-time approximation of predictive variance [\[105, 106\]](#).

[Algorithm 6](#) details the procedure for updating a node's information content. In lines 6-8, the location of a future measurement z at pose p , is added to the set of past observations (training points) from the entire node graph. This is used

Algorithm 6 Information_GPVR-ST()

Require:

Proposed robot pose or location from RRT/RIG **Steer** p , current map/state estimate $\mathcal{M}_{\mathcal{D}}$, covariance function $k(\cdot, \cdot)$, prior map variance σ , variance of observation noise σ_n^2 , near node information I_{near} ;

- 1: $\bar{\sigma} \leftarrow \sigma$ \triangleright Initialize updated map variance as the current map variance
- 2: **if** I_{near} is not empty **then** \triangleright Initialize information gain
- 3: | $I \leftarrow I_{\text{near}}$
- 4: **else**
- 5: | $I \leftarrow 0$
- 6: $z \leftarrow$ Propose a future measurement at location p and map \mathcal{M} \triangleright Calculate posterior map variance at training and query points
- 7: $\bar{\sigma} \leftarrow \text{LOVE}(X, X_*)$
- 8: **for all** $i \in \mathcal{M}_{\mathcal{D}}$ **do**
- 9: | $I \leftarrow I + 1/2 \left[\log \det \left(\sigma^{[i]} \right) - \log \det \left(\bar{\sigma}^{[i]} \right) \right]$
- 10: **return** I (total information gain), $\bar{\sigma}$ (updated map variance)

to create a new map state containing the previous training points plus the new measurement and the preexisting query points where the GP is evaluated. Next, the posterior variance is calculated (lines 8) using LOVE (LancZOs Variance Estimates) [105, 106] to produce a posterior variance at the proposed locations of training points $X \in \mathcal{M}_{\mathcal{D}}$, query points $X_* \in \mathcal{M}_{\mathcal{D}}$, and the variance of observation noise σ_n^2 . Finally, information content of the entire posterior map is updated and the information gain is returned as a marginal variance (lines 9-11).

5.3.5 Convergence criterion

The closely related Incrementally-exploring Information Gathering (IIG) algorithm modifies RIG with an information-theoretic convergence criterion [83]. Specifically, IIG bases the stopping criterion around a *relative information contribution* (RIC) criterion that describes the marginal information gain of adding a new observation relative to the previous state the RIG tree (see Equation 15 in [83] for a comprehensive discussion of the IIG algorithm and for a definition of the RIC). There, it was used as a tunable parameter that established a planning horizon for information gathering. In this paper, we use posterior map variance as a lower bound for mean-square error (MSE) (Equation 5.10) at a arbitrary test location in the GP, given optimal hyperparameters θ for the GP regression model. We replace the stopping criterion in IIG with a threshold

established by the operator as the lower bound of expected prediction MSE.

$$\text{MSE}(\hat{f}_*) \geq \underbrace{\mathbb{V}[f_*]}_{=\sigma_{*|y}^2(\theta)} \quad (5.10)$$

It is important to note that this inequality holds for the hyperparameters θ that produce an optimal predictor of f (see Result 1 in [107] for a proof of Equation 5.10 using the Bayesian Cramér-Rao Bound (BCRB).) In practice, θ is learned from the data. For approximate (suboptimal) values of θ , the bound of Equation 5.10 will not hold, as additional error is introduced from the unknown model hyperparameters. However, when coupled with adaptive planning techniques to learn θ from observations, then the posterior variance approaches the true lower bound of the MSE. A deeper analysis of the implications of this application is a target of future work.

5.3.6 Path selection and planning

Once the planner terminates (either by the convergence criterion or after a fixed planning horizon), a path must be selected from the graph of possible sampling locations. We use a vote-based heuristic from [83] that ranks paths according to a similarity ratio and biases towards paths that are longer and more informative with a *depth-first search*. In the simulated environment, parameters are set for vehicle speed, sampling frequency, and replanning interval. The vehicle alternates between planning, executing, and replanning in a receding-horizon fashion, such that 2-3 waypoints are visited in each planning interval.

The path selection strategy is independent of the informative path planning algorithm and can be thought of as an orienteering problem within a tree of sampling locations.

5.4 Experimental Evaluation and Discussion

In this section, we contrast our proposed spatiotemporal-informed planner (IIG-ST) against a traditional coverage survey strategy (see Figure 5.2), and an informed planner that does not consider temporal variation (IIG). We evaluate the accuracy of the final map representation at the end of the survey period under varying choices of spatial and temporal priors. We also consider the ancillary objective of making predictions of the state of environment at arbitrary points in time. This can be useful for objectives that wish to reconstruct the dynamics of a system, such as modeling a vector field. However, this is

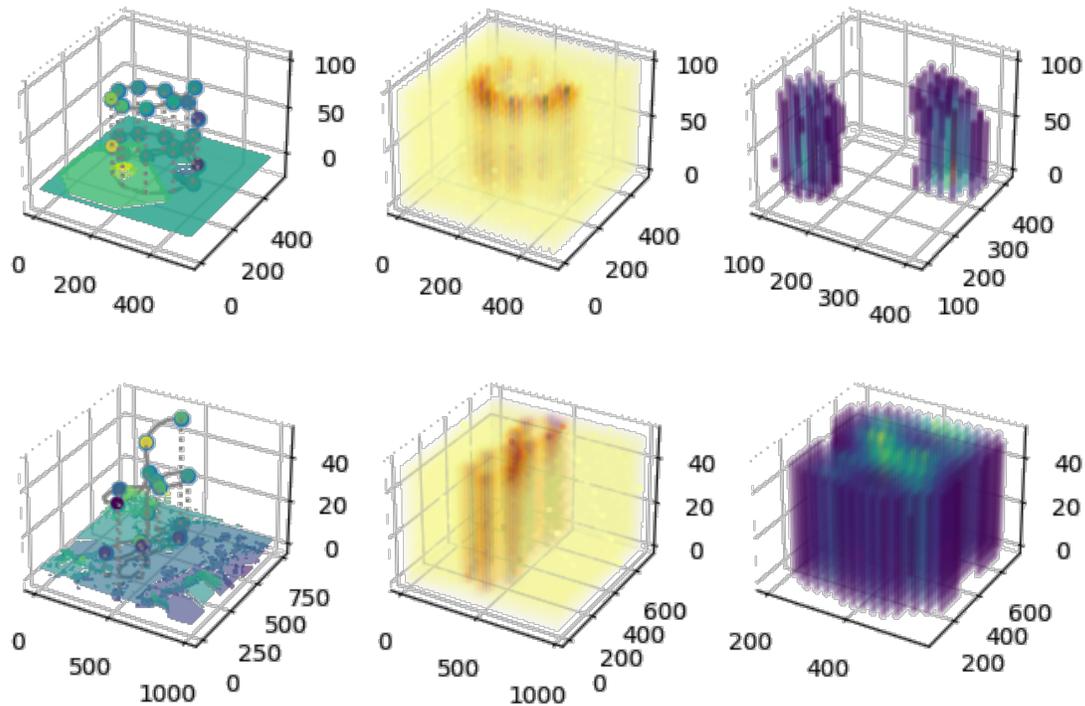


FIGURE 5.2: A visualization of the benchmark (coverage) sampling scenarios (top: fluid simulation, bottom: ocean sampling simulation). The posterior variance is depicted in the second panel, and the posterior mean in the third, with near-zero values filtered show the underlying structure. The coverage planners are given a path budget and node budget equivalent to the median of the equivalent metrics among all runs of the informed planners. Observations are collected on a circular coverage in the synthetic environment and a lemniscatic coverage in the oceanic experiment.

complicated by the fact that the survey envelope is anisotropic in the temporal dimension – the robot and sensor can only travel forward through time.

5.4.1 Experimental setting

Our objective is to model the end-state of a spatial phenomenon that undergoes advection and diffusion in a 2D environment. This can represent the movement of a substance of interest in a fluid, a porous medium such as soil, or any number of similar natural processes. Two fluid parcels are initialized with inversely-proportional velocities, at opposite corners of a 500×500 -unit gridded environment. The fluid parcels advect and diffuse according to the Navier-Stokes equations for an incompressible fluid, implemented as a forward-differencing discretization without boundary conditions.

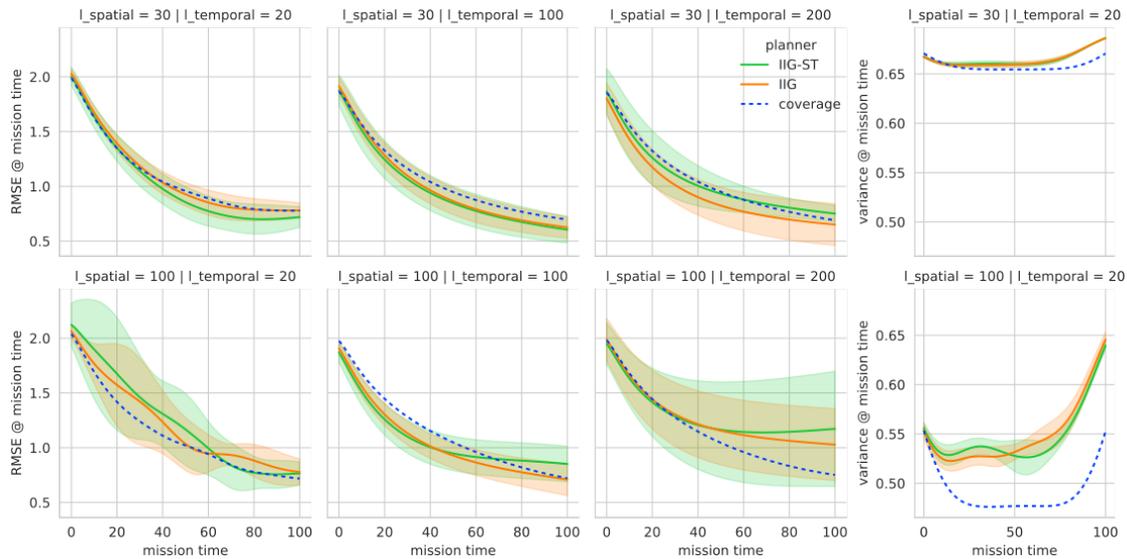


FIGURE 5.3: [Advection/diffusion simulation] A comparison of map error and posterior variance (lower is better) at different locations in the mission time for different spatiotemporal priors. Optimal priors are chosen in the top left panel ($l_t = 20$ and $l_s = 30$) and become increasingly suboptimal in other panels. IIG-ST (our planner) is compared the same planner lacking time information (IIG) and a circular survey strategy. The error metric is expressed across the entire spatial domain at different time indices (denoted on the x-axis), and reflects the error between the estimated map and the state of the environment *at that time*. Y-axis scales are shared between rows.

We initialized the RIG-planner with fixed planning parameters: the vehicle can move a maximum of 100 map-units, every 5 time-units. Replanning is done every 10 time increments, and planning within each increment stops when estimated $\mathbb{V}[f_\star] = 0.15$. Sampling occurs once every 5 time increments. We set the time budget to be 100 units and compute the accuracy of the final representation of the map at $t = 50$ min. Map accuracy at different moments in mission time are presented in [Figure 5.3](#). While the planner was not given a movement budget, the fixed speed of the vehicle and finite time-horizon resulted in consistent numbers of observations ($M = 21.0, SD = 0.2$) and path lengths ($M = 1236, SD = 36$) among the informative planners. The coverage baseline is given a proportional budget (21 observations, 1610 map units traveled). This is sufficient to complete a full tour of the environment with revisitation (see [Figure 5.2](#)). The full table of parameters set for the planner can be found in the accompanying video. We executed the experiments in a GNU/Linux environment on a 3.6 GHz Intel i7-4790 computer with 11 GB of RAM available. All procedures used single-threaded Python implementations for RRT sampling from [\[85\]](#) and multi-threaded posterior variance final map predictions were performed using implementations from GPyTorch [\[106\]](#) without GPU or TPU acceleration so as to simulate the resources available on an embedded system.

5.4.2 Consequences of the temporal prior

To demonstrate the consequences of incorporating a spatiotemporal prior on informative planning in dynamic fields, we use the composite covariance function given in equation [\(5.5\)](#) both in planning and for evaluating the accuracy of the final map representation. This is notable for the baseline comparisons—while the coverage planner follows a deterministic trajectory, different map accuracies and variance reductions are expected depending on the choice of spatiotemporal prior during the construction of the final map model.

For the temporal relation, we use a RBF kernel with length scales of $\ell_t = 20, 100, 200$ time units. The spatial relation comprises a Matérn kernel with $\nu = 3/2$ and length scales of $\ell_s = 100$ distance units. To verify that the robot solves the problem in [section 5.2](#), we evaluate the root-mean squared error between the map representation at $t = 100$ and the state of the field at the same time. As the planner only requires the posterior covariance, it is not necessary to produce continuous estimations of the map state, so the final representation is computed once the simulation has ended. 20 episodes are run for each hyperparameter combination and summaries of average error, average posterior variance and standard deviations are found in table [Table 5.1](#).

		<i>RMSE</i>				\bar{V}	
planner	ℓ_s	ℓ_t	t_{max}	t_{all}	t_{max}	t_{all}	
IIG	30	20	0.781 (0.066)	1.123 (0.072)	0.686 (0.0)	0.664 (0.001)	
		100	0.612 (0.096)	1.035 (0.098)	0.64 (0.005)	0.626 (0.006)	
	100	20	0.762 (0.113)	1.288 (0.179)	0.645 (0.007)	0.547 (0.004)	
		100	0.75 (0.222)	1.093 (0.116)	0.462 (0.02)	0.413 (0.025)	
IIG-ST	30	20	0.733 (0.089)	1.092 (0.064)	0.686 (0.0)	0.665 (0.001)	
		100	0.611 (0.121)	1.028 (0.135)	0.638 (0.005)	0.624 (0.006)	
	100	20	0.768 (0.101)	1.3 (0.238)	0.64 (0.004)	0.547 (0.005)	
		100	0.866 (0.194)	1.114 (0.117)	0.458 (0.014)	0.414 (0.017)	
coverage	30	20	0.777	1.132	0.671	0.658	
		100	0.697	1.099	0.639	0.638	
	100	20	0.718	1.173	0.552	0.491	
		100	0.721	1.19	0.398	0.394	

<i>RMSE</i>			
planner	ℓ_s	t_{max}	t_{all}
IIG	5	6.654 (0.015)	5.499 (0.004)
	40	5.934 (0.382)	4.926 (0.087)
	100	3.835 (0.725)	3.777 (0.17)
IIG-ST	5	6.658 (0.013)	5.499 (0.003)
	40	5.846 (0.305)	4.904 (0.072)
	100	4.1 (0.739)	3.698 (0.252)
coverage	5	3.826	6.238
	40	3.281	5.506
	100	2.909	4.56

TABLE 5.1: (Top) [Advection/diffusion] Aggregated ($n = 20$) map accuracy (RMSE) and posterior variance (mean, *std*) of the spatiotemporal planner (IIG-ST) compared to a spatial-only and deterministic survey strategies for fixed length scales. (Bottom) [Ocean dataset] Aggregated $n = 20$ map accuracy for the ocean water quality experiment ($\ell_t = 100$ for all runs). Lower numbers are better. Note: standard deviation values are not expressed for the deterministic planner.

In [Figure 5.3](#), we examine the choice of kernel hyperparameters on the performance of our planner. Optimal parameters were established offline using the baseline samples and a standard marginal log likelihood function and the Adam optimizer in gpytorch ($\ell_t = 20$ and $\ell_s = 30$). These serve as the basis of comparison in the top-left panel of [Figure 5.3](#) and resulted the spatiotemporal planner outperforming the temporally-naive and baseline planner for on average, throughout the entire mission duration. Large lengthscales imply a greater degree of correlation across space or time, and result a greater reduction of posterior variance. A reduction of model uncertainty should translate to a higher map accuracy, however this is not the case if the spatial priors are unrepresentative. For example, while the coverage planner had lower variance due to a longer path traveled and more dispersed observations, the resulting map accuracy was not better than the informative planners, leading to the conclusion that the spatiotemporal prior did not reflect the variation of the observed process. We want to emphasize that path planning algorithms based around variance reduction should also place the metric within a broader context of the practical objective – map accuracy.

For informative planners, the effect is magnified, as the planner will move toward more dispersive sampling, thus missing high-frequency spatial phenomena entirely. This is demonstrated in the marginally improved accuracy and lower posterior variance for IIG-ST when given a unrepresentative spatial and temporal prior. In worst-case scenarios, a very unrepresentative temporal prior ($\ell_t = 200$) can reduce the performance of the spatiotemporal planner *below* the baseline ([Figure 5.3](#), Col. 2). As the ultimate goal of informed robotic sensing *is* model accuracy and not simply variance reduction, hyperparameter optimization must be a key component for accurate mapping and is a common practice in adaptive planning [50]. Furthermore, a time-varying kernel could be specified and optimized as observations of the environment are gathered. Future work will investigate the effect and performance of updating model priors during the course of a survey mission.

The final map posterior is evaluated with the same spatiotemporal kernel in all cases, regardless of planning method to ensure a fair comparison between the methods. Only the spatiotemporal planner (IIG-ST) is able to make use of temporal variance during replanning. Training observations are obtained from a point sensor model, where the a "sample" is obtained by the simulated agent querying the ground-truth scalar field at a sample location. We use a sparse representation of posterior variance, evaluated at a $1/20$ scale spatial resolution for a total of $25 \times 25 \times 50$ query (inducing) points. Recent advancements in spatiotemporal GPs with separable kernels, enable computational scaling to scale linearly in the temporal dimension, instead of cubic [98]. These and other

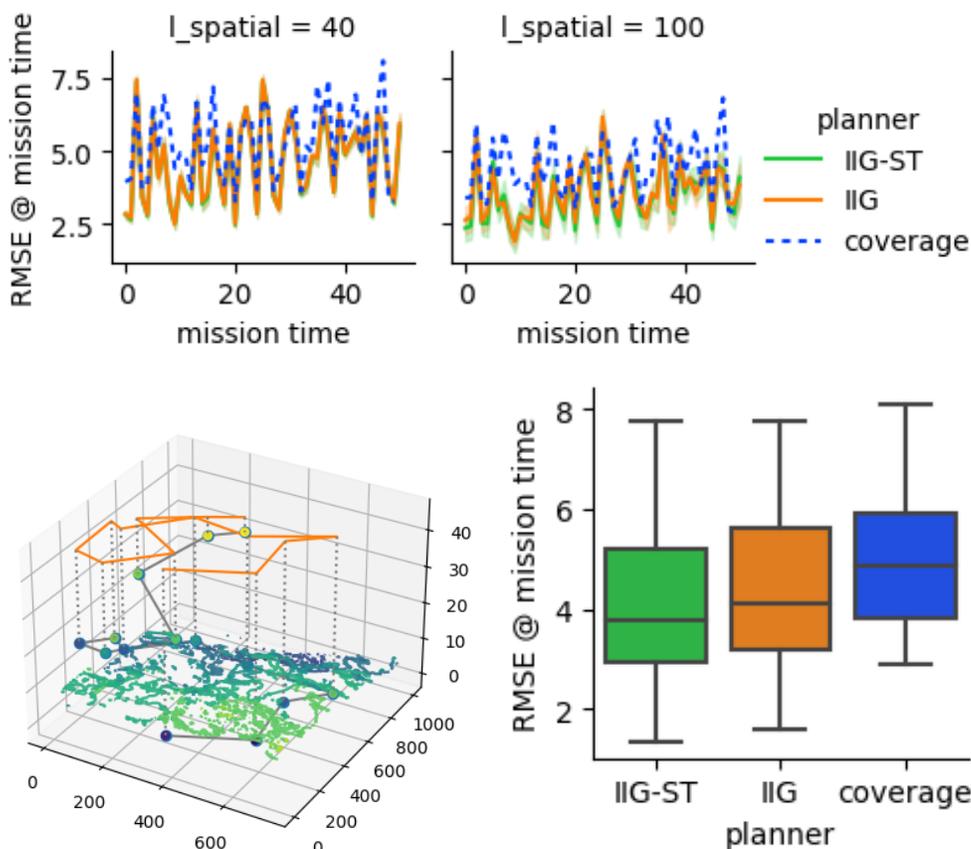


FIGURE 5.4: Example results from the ocean modeling experiments. (Top) Map error as a function of mission time, ($l_t = 100$). (L) Example trajectory, with path trace projected above a representation of the environment at $t = 0$. (R) Aggregated statistics from the figures in the top panel.

recent developments are reducing the computational burden of large GPs and informative planning with spatiotemporal information at a large scale.

5.4.3 Ocean particulate mapping scenario

We demonstrate our spatiotemporal IPP approach in a syoptic-scale simulation using real-world ocean reflectance data. The data was collected in an approximately $1500 \times 1000 \text{ km}$ region off the west coast of California from the Moderate Resolution Imaging Spectroradiometer (MODIS) aboard NASA’s Terra and Aqua earth observation satellites [108]. Rasters of weekly median reflectance from band 9 (443 nm wavelength) were assembled for the calendar year of 2020. Backscattered light in this wavelength band is highly correlated with the concentration of suspended organic and inorganic particles (e.g. sediments) in the water. In terrestrial and oceanic waters, this can be used as

an indicator of water quality, which can guide management decisions related to water diversion and treatment.

We simulated an autonomous aquatic vehicle (AUV) with characteristics similar to the Wave Glider, which is an AUV capable of extended oceanic monitoring campaigns by using oceanic waves for propulsion. Based on the long-mission average speed of 1.5 knots, our simulated vehicle could cover a maximum of 330 km per week. We compare the performance of our informed planner against a fixed lemniscatic coverage pattern. As with the previous section, we evaluate the RMSE of the map representation, both at the final time step and at arbitrary temporal increments in the mission envelope. Summaries of average error, standard deviations, and posterior variance are presented in [Table 5.1](#) and [Figure 5.4](#). As with the previous experiment, posterior variance and map accuracy are evaluated at a 1/20 scale spatial resolution. Also, as with the previous experiment, the performance of IIG-ST is sensitive to the choice of hyperparameters.

5.5 Conclusion

This chapter presented an approach for environmental modeling using a novel spatiotemporally-informed path planner. We presented a framework for quantifying the information gain of sampling locations based on their location and time and quantifying the operative outcome – map accuracy. We show that this informed strategy is computationally tractable with modern computational techniques and can outperform naive and conventional approaches, conditional on an appropriate spatiotemporal prior.

Multiple avenues for future work lead from this effort. Adaptive planning can be used to revise the spatiotemporal prior as measurements are collected between replanning intervals. This approach can be extended to consider time-varying kernels, variable sensor models, and multi-robot systems.

6

Unified adaptive and cooperative planning using multi-task coregionalized Gaussian processes

6.1 Introduction

With the growth in use of autonomous vehicles for agriculture and natural resource management [109], so too grows the burden for distilling actionable information from the observations gathered. In the field of geostatistics, there is a long history of using mathematical methods for modeling an environment through sparse measurements [72]. This is particularly relevant when observations require direct sampling, such as for robotic plant phenotyping [110] and robotic sampling of plant tissues [111].

This chapter considers the task of modeling a dynamic phenomenon in an environment with a team of robots equipped with point sensors. Given a limited movement budget, it is desirable for the robots to visit locations that will result in more accurate signal reconstructions when the surveying mission is complete.

Previous efforts: This task of *informative path planning* (IPP) aims to find obstacle-free trajectories in an environment that maximize the information gathered during traveling. When applied to environmental monitoring tasks, it is closely related to the task of *optimal sensor placement* [82], and can be considered as an optimization problem subject to constraints such as the physical confines of the rows of an vineyard [91] or when observations are constrained to a set of pre-established monitoring locations [99]. When the process to be monitored evolves over time, there is a choice to be made: whether to explore unvisited locations in the environment, or to re-survey previously-visited locations that may have changed. Approaches explored in prior works include a recursive-greedy approach for surveying a time-varying field with a single robot [112] and multi-robot efforts that leverage clustering in order to divide the observation domain among different sampling vehicles. Recent IPP

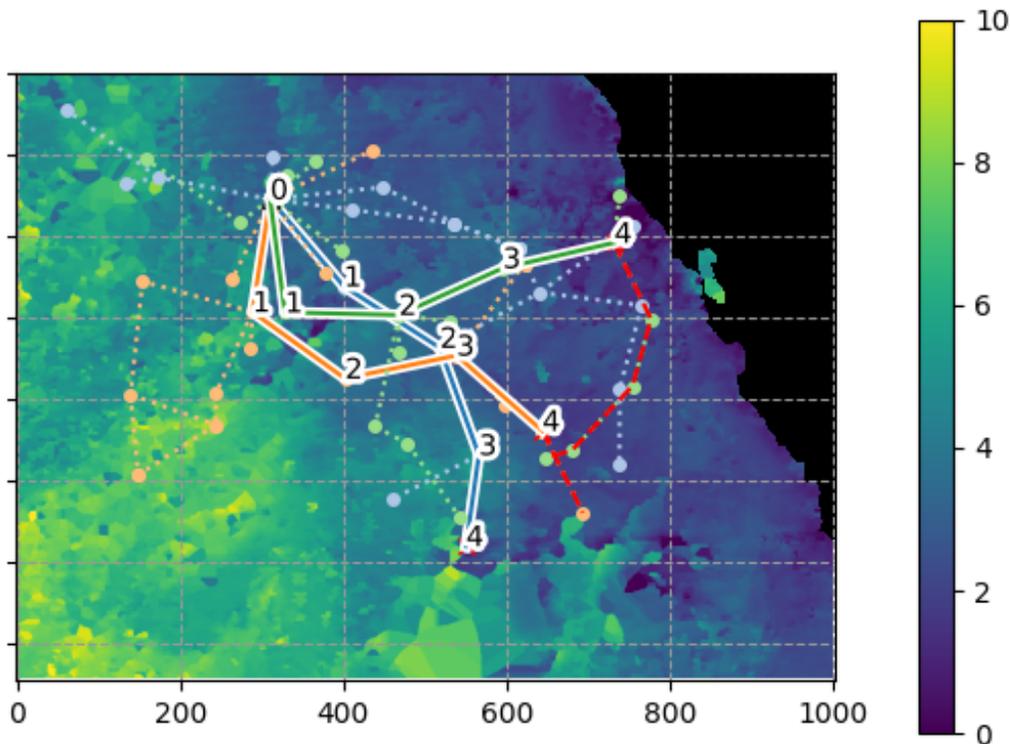


FIGURE 6.1: An example of a planning mission with 3 vehicles in an ocean turbidity monitoring mission. Vehicle movement history is displayed with a white outline. In this image, 3 vehicles have acquired 4 observations each. Candidate paths and sampling locations as proposed by our planner are shown in dashed lines. The planner for the green vehicle has just finished, and proposed the best candidate path, shown in the dotted red line.

approaches consider robotic path planning in response to multiple objectives, such as variable sensor models [84], and multi-modal sensor configurations [113].

When surveying unknown environments, it is often desirable to utilize an *adaptive* sampling scheme, where new samples are targeted based on information collected from previous samples in the surveying mission in order to improve the overall surveying ability of the robotic system (subject to the evaluation criteria) [50]. When extended to robotic surveying, this has been called the adaptive informative path planning (*AIPP*) problem. Recent literature has found learning-based approaches to be particularly suited to *AIPP* across a wide variety of mapping objectives [114] [115] [116] and different approaches have been used to extend the task into multi-robot surveying efforts [117] [118].

In this chapter we present a multi-robot informative path planner *IIG-Cooperative*, that guides robots to gather observations in regions that result in a higher expected improvement of the environmental map. Our approach is

cooperative as it is able to incorporate information from other robots involved in the team surveying task. This algorithm builds upon *IIG*, an informative sampling-based planner [83] that utilized mutual information as the basis for establishing when a proposed sampling location would result in an improved model of an environment. The algorithm in turn builds upon *RIG*, which presented an informative planner based on RRTs [119] find obstacle-free trajectories through an environment that maximize information gathered along the way by leveraging the asymptotic optimality of random-trees. [78]. In a previous effort, we presented *IIG-ST*, which utilizes an information-theoretic stopping criterion and an information function that trades off between exploring new environments and re-surveying previous potentially-stale observations, when monitoring a time-varying spatiotemporal phenomenon [120].

IIG-Cooperative adds an information function that incorporates the location of observations from other sensing vehicles into a robot’s evaluation of the information gain at proposed sampling locations. In [120], the performance of the algorithm was dependent on tuning the information function to the environment through expert selection of spatial and temporal priors. In this work, we overcome the limitation by presenting an adaptive planning framework that alternates between planning, execution, and updating the environmental priors in order to generate an improvement of planning performance as a survey proceeds. The main contributions of this work are as follows:

- *IIG-Cooperative*, an informative path planner that integrates multi-vehicle interactions through a separable kernel, reconciled to the surveying objective
- An open-source ¹ adaptive planning framework that takes advantage of observations gathered to create a best-effort continuous improvement of model hyperparameters and planning optimality
- An experimental validation of the base planner and a spatiotemporal variant under different configurations and communication scenarios

Figure 6.1 shows an example of our cooperative planner used in an ocean monitoring example. Our approach is discussed in section 6.3 and is enabled by the use of a multi-task Gaussian process, where each robot’s observations comprise an input of a multi-input modeling task. We extend this effort in an adaptive planning framework, applied in experiments on simulated and natural environments in section 6.4. Our framework can accommodate a diverse variety

¹<https://github.com/ucmercedrobotics/ipp-RRT-family>

of planning and configuration scenarios, including variable sensor models and variable noise models, all of which will serve as the basis for future investigation (discussed in [section 6.5](#))².

6.2 Problem Formulation

Consider the case where an environmental field is observed by different sensing agents v , comprising a team of N_v agents. The different agents may be equipped with different sensors that may have different noise characteristics, and observations that are unlikely to be coincident.

We can consider three scenarios:

1. Where observations from all sensing agents are shared to form one unified training set, for a unified model that serves as the basis for allocating future sampling locations by a global planner.
2. Where observations from a given sensing agent v are used to build a representation of the environment that is *only used* to inform future plans for an agent v .
3. Where correlations between sensing agents are captured in a covariance function. Each agent independently builds a representation of the environment, assisted with knowledge of observations collected from other sensing agents.

Case 1 can be addressed through a variety of approaches, including partitioning an environment among a set of sensing agents. Case 2 is applicable if we apply any single-robot IPP algorithm to a team of surveying robots, who independently sample and plan without knowledge of the other agents in the team.

This chapter considers Case 3, where each member v of the team of robots is *tasked* with producing an independent, internal representation of the environment to guide its own planning. We denote each task with the letter j . Observations are shared between robots and are used to update each robot's internal representation of map uncertainty.

In our planning framework, robots alternate between: planning, surveying, communicating, updating priors, and re-planning. This occurs in a continuous loop until the end of the allotted survey period, or until the robots consume their

²This section is based on the following manuscript, which is currently in review: Booth, L., Carpin, S. (2025). Unified adaptive and cooperative planning using multi-task coregionalized Gaussian processes

movement budget. At the end, observations from all robots are aggregated to form a final, unified model of the environment.

6.3 Methods

6.3.1 Overview

Our planning framework encodes the intuition that it is more desirable to collect observations where a model is deficient, rather than where a model is sufficient. In the following sections we describe: the form of the environmental model and how we derive model adequacy (subsection 6.3.2); how we encode and update prior knowledge about the environment and the sensing vehicles (subsection 6.3.3); and how this knowledge allow us to determine where to sample (subsection 6.3.4). Implementation details of our planner and an algorithmic overview are found in (subsection 6.3.5)

6.3.2 Environmental model

Congruent with standard approaches in geostatistics, we extend a 2D spatial regression task (Kriging) to consider a 3D regression task of an unknown scalar-valued environmental process that changes over time (e.g., chemical concentration and distribution, soil moisture content, etc.) represented as the function: $f: \mathcal{X} \rightarrow \mathbb{R}$ that is modeled on discrete intervals in space ($N_{x,y}$) and time (N_t): $\mathcal{X} \subset \mathbb{R}^{N_{x,y}} \times \mathbb{R}^{N_t}$. The phenomenon is observed through noisy measurements $y_{i,j}$ made at location $i \in N_{x,y} \cup N_t$ by sensing agent j : $y_{i,j} = f(x_i) + \varepsilon_j$. Noisy perturbations ε_j are modeled with a zero-mean, homoskedastic, additive Gaussian noise model, that is consistent within a given sensor on vehicle v (denoted by task j): $\varepsilon \sim \mathcal{N}(0, \sigma_j^2 I)$.

GP regression

The modeling approach used in this chapter is a multi-dimensional Gaussian process regression and was implemented identically to the previous chapter (see subsection 5.3.1).

6.3.3 Spatiotemporal and task priors

Spatiotemporal prior

The kernel (or, covariance function) k is a function that provides the expected correlation between pairs of data points. While arbitrary functions of input pairs

are not guaranteed to be valid covariance functions, there exists a considerable amount of choice and discretion in choosing a function that is appropriate to the predictive task. Both the choice of kernel and the function hyperparameters encode assumptions about the property which we wish to predict [77].

Following [120], we establish a base kernel, composed of a spatial and temporal kernel:

$$k((s, t), t(s', t')) = k_s(s, s')k_t(t, t') \quad (6.1)$$

where s refers to the spatial index (such as, a geographical coordinate) and t refers to a temporal index (a timestamp). For the spatial dimensions, we use a Matérn kernel with $\nu = 3/2$, chosen in part for its use in the geostatistical literature and its ability to capture discontinuities present in natural phenomena. We use a radial basis function kernel to capture smoothly diffusive process in the time dimension. For additional discussion and details about the kernel choice, readers may refer to [72] and [120].

Task prior

In section 6.2, we describe how each sensing agent is *tasked* with producing a unique model of the environment, to guide its path planning. Following the terminology from [121], we describe our system of models as *multi-task*, where the process f is observed by different sensing agents j . Our model is *multi-output*, and can produce a unique representation of the environment based on observations collected from a particular sensing agent j .

Following the notation of [121] let us consider a training set constructed of data pairs $S_d = (\mathbf{X}_d, \mathbf{Y}_d) = (\mathbf{x}_{d,1}, y_{d,1}), \dots, (\mathbf{x}_{d,N_j}, y_{d,N_j})$ for outputs $D = N_j$ (the number of sensing agents) and number of query points N . From this, we obtain a vector of sampling locations for each sensing vehicle j : $\mathbf{X} = \{\mathbf{X}_j\}_{j=1}^D = \mathbf{X}_1, \dots, \mathbf{X}_D$, where $\mathbf{X}_d = \{\mathbf{x}_{d,n}\}_{n=1}^N$. In this general sense, a separate process f_d can be learned by training set S_d , where: $\mathbf{f}(\mathbf{X}) = (f_1(\mathbf{x}_{1,1}), \dots, f_1(\mathbf{x}_{1,N}), \dots, (f_D(\mathbf{x}_{D,1}), \dots, f_D(\mathbf{x}_{D,N}))$.

We can construct a similar formulation for a vector-valued GP as in Equation 5.3. The vector-valued kernel \mathbf{K} is an $ND \times ND$ with entries: $(\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j))_{d,d'}$, for $i, j = 1, \dots, N$ and $d, d' = 1, \dots, D$.

We can consider a *separable* kernel function, formulated as a sum of products between a kernel function for the *input space alone* (the spatiotemporal kernel in this work) and a *task* kernel function that encodes interactions between the outputs (correlations between the sensor models, in this work). Such a kernel can be defined as the Hadamard product of an input kernel and a task kernel and takes the form:

$$(\mathbf{K}(\mathbf{x}, \mathbf{x}'))_{d,d'} = k(\mathbf{x}, \mathbf{x}')k_T(d, d') \quad (6.2)$$

where k is the input kernel, k_T is the task kernel, both defined over $\mathcal{X} \times \mathcal{X}$ with $\{1, \dots, D\} \times \{1, \dots, D\}$. Equivalently, this can be written as a matrix expression:

$$\mathbf{K}(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') \mathbf{B} \quad (6.3)$$

We establish an index kernel, defined by a lookup table of indices corresponding to the number of tasks. In this chapter, the number of tasks is equal to the number of sensing agents (N_v), that is: $N_v = N_j = D$:

$$k(i, j) = \left(BB^\top + (\mathbf{I}v)_{i,j} \right) \quad (6.4)$$

where B is a low-rank matrix that establishes the variance between tasks and v is a positive constraint on the inter-task variance. Refer to [121] for a detailed treatment of kernel functions for multi-output GPs and [106] for details of the function `pytorch.kernels.IndexKernel` used in our implementation.

It is important to note that the vehicles do not attempt to construct the posterior variance from the perspective of other vehicles in the team. For each vehicle, only the output corresponding to the ego vehicle is used for planning. The other outputs could be used to infer model variance from the perspective of other sensing agents, and could be used to project their probable next-actions. While an interesting area for future work, this consideration is outside of the scope of this study.

6.3.4 Utility Formulation

Understanding that the differential entropy of a Gaussian random variable is a monotonic function of its variance, we can construct a utility function based on the reduction of map entropy H , given a new observation Z . We derive this from the posterior variance, which is obtained from Equation 5.4. Specifically, we evaluate the information gain of a new proposed sampling location, using the mutual information I between the current training set X and a new set X' , containing the observation Z .

For a random vector of observations $\mathbf{X} = (X_1, \dots, X_n)$, for every X_i the mutual information becomes:

$$\hat{I}^{[i]}(X_i; Z) = \frac{1}{2} \left[\log(\sigma_{X_i}) - \log(\sigma_{X_i|Z}) \right] \quad (6.5)$$

and can be calculated as the sum of marginal variances at i : $\hat{I}(X; Z) = \sum_{i=1}^n \hat{I}^{[i]}(X_i; Z)$. Refer to [120] for additional context and see [83] for a derivation. Crucially, we are able to obtain from the covariance matrix produced by the kernel $\mathbb{V}[X_i] = K^{[i,i]}$.

6.3.5 Path selection and planning

General framework

Path planning proceeds according to the procedure described in (IIG, [83]) and chapter 5 (IIG-ST, [120]), using the task-aware covariance function described in the subsection 6.3.3.

Convergence criterion and path selection

We utilize the convergence criterion from [120] in the re-planning stage, to establish when the agent should stop adding new proposed locations to the RIG tree and switch to path generation. Path generation is performed using a vote-based heuristic from [83]. We use posterior map variance as a lower bound for mean-square error, given optimal hyperparameters θ for the kernel function.

$$\text{MSE}(\hat{f}_\star) \geq \underbrace{\mathbb{V}[f_\star]}_{=\sigma_{\star|y}^2(\theta)} \quad (6.6)$$

These kernel hyperparameters θ comprise another component of our spatiotemporal and task-priors. Optimal parameters can be chosen in a standard Bayesian approach, using the marginal log likelihood (MLL) of the GP model, when applied to observed data. We employ this approach to update model hyperparameters, as the sensing agents receive observations during the survey mission.

Adaptive planning

An overview of the complete adaptive planning routine is visualized in Figure 6.2. Each vehicle alternates between collecting observations, updating internal parameters and re-planning at a fixed interval until a time budget is elapsed. Each agent broadcasts the value and location of a sample immediately upon collection, and all agents continuously listen for observations from other robots in the team. Each robot re-plans after every 2nd sample.

In the experiments, it assumed that vehicles have access to a wide-area, low-bandwidth communication link (such as LoRa [93]) and are able to communicate their observations globally. Prior to the next re-planning procedure, each vehicle

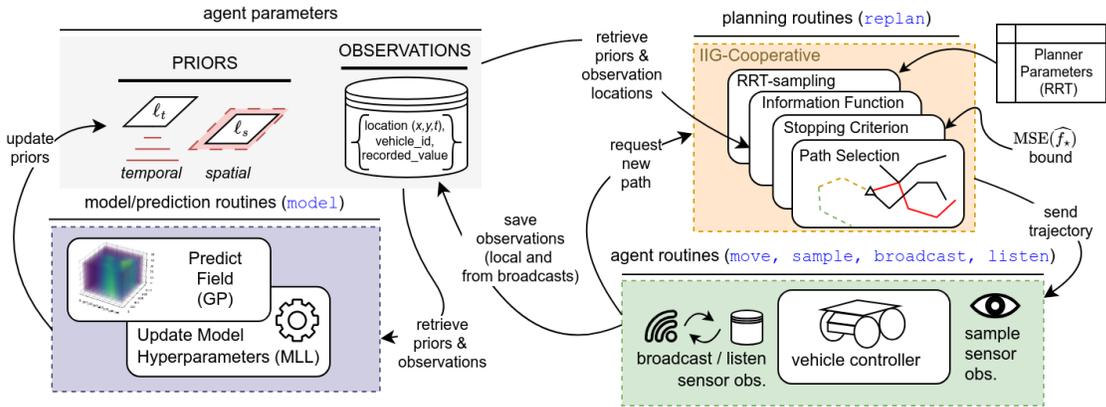


FIGURE 6.2: Schematic overview of the adaptive planning routine. Not pictured: When the vehicle controller has reached its movement budget (time or distance), the modeler is called one last time to produce a final map prediction.

evaluates the map expectation \mathbb{E} (posterior mean) from the multi-task model, incorporating all observations \mathbf{y}_j collected by the agent and communicated by other agents. The model output is used to calculate the marginal log likelihood of the GP with respect to the observations collected, which forms the basis of the hyperparameter optimization routine.

6.4 Experimental Evaluation and Discussion

In this section, we evaluate our coordinated planner *IIG-Cooperative* against the non-cooperative *IIG-ST* introduced in [120] as applied to the task of surveying and modeling a physical phenomenon that changes in an environment over a fixed period of time. In the first scenario, the surveyed phenomenon advects and diffuses in a simulated fluid environment over time, similar to the dynamics of an environmental contaminant in the soil, water, or air. In the second scenario an oceanic turbidity dataset is used as the target for the surveying objective.

We evaluate our planner according to the objective that would be most salient to a surveyor; that is, the accuracy of the final map representation at the end of the survey period ($t = 100$) with a root-mean squared error metric (*RMSE*). We also consider the auxiliary objective of making predictions at arbitrary points in time. This is relevant if the operator wishes to reconstruct the dynamics of the system. However, while our spatiotemporal planners incorporate time into the planning objective, the robot and sensor can obviously only travel forward through the temporal dimension.

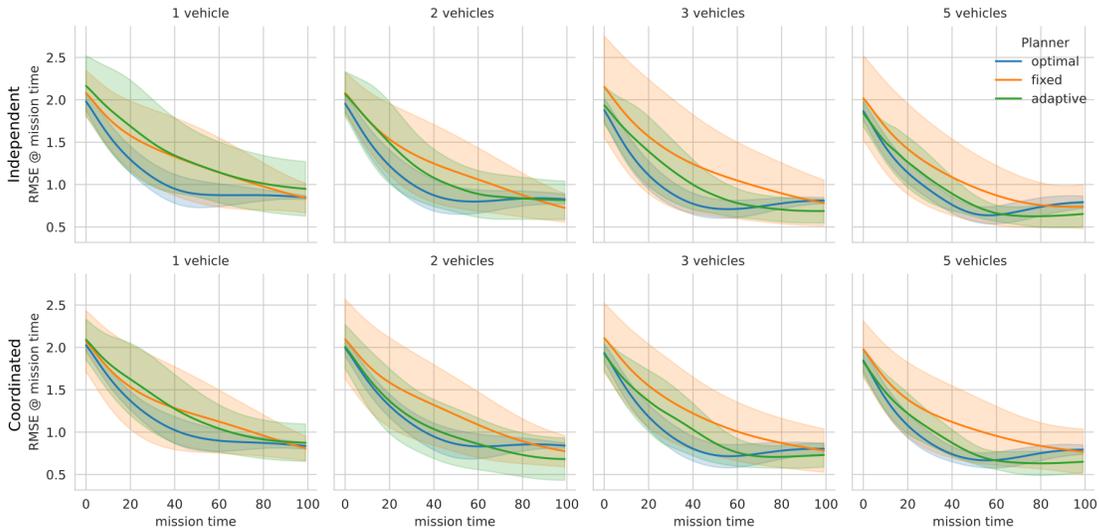


FIGURE 6.3: **Advection/diffusion simulation:** Map error (lower is better) at different moments in the survey mission time for different planning configurations for *IIG – Cooperative* with varying numbers robots deployed in the sensing task. Planners with fixed priors are run in two configurations, fixed-suboptimal ($\ell_t = 50$ and $\ell_s = 100$) and fixed-optimal ($\ell_t = 20$ and $\ell_s = 30$). The adaptive planner starts with the suboptimal priors and continuously updates model hyperparameters throughout the survey mission. All planners are run with global communication (bottom row) and without communication, in an independent planning scenario (top row).

6.4.1 Experimental setting

The simulated world implements advection and diffusion according to the Navier- Stokes equations for an incompressible fluid (forward-differencing discretization). All robots are initialized with the same path planner with fixed planning parameters that are updated independently during the course of a survey. The full table of parameters set for the planner can be found in the accompanying video and the code repository. We executed the experiments in a GNU/Linux environment on a 3.6 GHz Intel i7-4790 computer with 10 GB of RAM available. RRT planning procedures were derived from [85] and GP posterior variance and final map predictions were executed with GPyTorch [106], run without hardware acceleration so as to simulate the resources available on a mobile robot.

6.4.2 Comparison of cooperative and adaptive planning

To assess the effectiveness of the cooperative and adaptive planners, we compare the planners in three configurations: A planner with fixed model lengthscales (ℓ) chosen proportional to the size of the world (“fixed” planner, $\ell_s = 100, \ell_t = 50$); an adaptive planner that starts with the same

hyperparameters as the fixed planner, but is allowed to update as observations are collected; and a fixed, planner given parameters determined through hyperparameter optimization along a dense, coverage path plan (“optimal” planner, $\ell_s = 30, \ell_t = 20$).

A summary of the main results can be found in [Figure 6.3](#), which presents the RMSE between the state of the world at time t and the state of the predicted world at time t , constructed after observations have been compiled by *all sensing vehicles* at the end of the survey mission. This aligns with the typical mode of operation in multi-agent surveys. All scenarios are run with the multi-task planner, which degenerates into a single-task problem when $n = 1$ sensing vehicle, and/or when there is no transfer of information between the vehicles (in other words, there is no information for the other input tasks). This later case is labeled as “independent” planning.

In [Figure 6.3](#), lower error values found on average across nearly the entire mission envelope for the adaptive planner, although it does not produce as accurate a map posterior as if it were given “optimal” hyperparameters. A notable observation is the map accuracy for the final time index of the survey period, when all planners converge to similar performance, with the “optimal” planner producing a slightly less accurate representation at $t = 100$ than the adaptive planner (or even the fixed planner, in some configurations). There are a few possible explanations for this phenomenon: 1. For large numbers of vehicles, by $t = 100$, the environment has been uniformly surveyed, regardless of the direction given by the informative planner. 2. The “optimal” parameters were derived through sampling along a conventional, coverage path, of a distance equal to the average distance traveled by the informed planners. This deterministic route does not guarantee representative samples across the entire spatiotemporal domain (and thus are not strictly optimal). This demonstrates another weakness of traditional surveying procedures. 3. The “optimal” parameters were determined through a maximum-likelihood estimator, optimized for the predictive ability across the *entire* survey envelope, and not *solely* on the final map state. Likely, different results would be obtained for parameters chosen to minimize the error of the final map state. Reconciling these potentially competing objectives within a unified planning framework could be explored in future studies.

A summary of the performance for all planning configurations is presented in [Table 6.1](#), where the two objectives are presented under: t_{100} for the predictive accuracy at $t = 100$ and t_{all} , for the predictive accuracy across the entire survey envelope using the coordinated planner (*IIG-Cooperative*) and the independent planning scenario. We also explore the addition of the spatiotemporal kernel to each configuration (“Function” column). Notably, the adaptive planner often

Planner	Function	N_v	Coordinated		Independent	
			t_{100}	t_{all}	t_{100}	t_{all}
Adaptive	<i>IIG</i>	1	0.888	1.344	0.935	1.367
		2	0.670	1.104	0.827	1.234
		3	0.749	1.109	0.703	1.091
		5	0.664	0.988	0.689	0.999
	<i>IIG-ST</i>	1	0.845	1.263	0.983	1.477
		2	0.703	1.174	0.797	1.181
		3	0.683	1.080	0.650	1.143
		5	0.618	0.986	0.566	1.015
fixed	<i>IIG</i>	1	0.820	1.321	0.839	1.374
		2	0.787	1.372	0.710	1.225
		3	0.806	1.312	0.782	1.374
		5	0.778	1.197	0.791	1.177
	<i>IIG-ST</i>	1	0.782	1.293	0.841	1.294
		2	0.760	1.247	0.749	1.365
		3	0.726	1.203	0.785	1.205
		5	0.745	1.161	0.648	1.149
optimal	<i>IIG</i>	1	0.841	1.162	0.841	1.128
		2	0.839	1.093	0.842	1.069
		3	0.825	1.043	0.802	0.986
		5	0.802	0.946	0.787	0.969
	<i>IIG-ST</i>	1	0.833	1.164	0.851	1.113
		2	0.843	1.143	0.812	1.063
		3	0.783	0.936	0.818	0.984
		5	0.787	0.963	0.796	0.972

TABLE 6.1: Fluid Simulation: Summary of average map error ($RMSE$) produced by observations collected by all vehicles v at the end of a survey mission. Error is represented across entire survey envelope (t_{all}) and for the last time step of the survey mission ($t = 100$). Lowest values within a given configuration are emphasized in bold.

outperforms the non-adaptive configuration, given informed “optimal” priors. This is due to the reasons outlined in the previous paragraph. In multi-robot configurations without coordination, minimal to no gains are found with the spatiotemporally-informed planner across most configurations. This is likely due to the aggregate effect of a more even coverage of the environment that occurs with time-naive robots, at high numbers.

6.4.3 Environmental monitoring scenario

As a proof of concept, we demonstrate our planners in a synoptic-scale ocean monitoring experiment, using ocean reflectance off the west coast of California

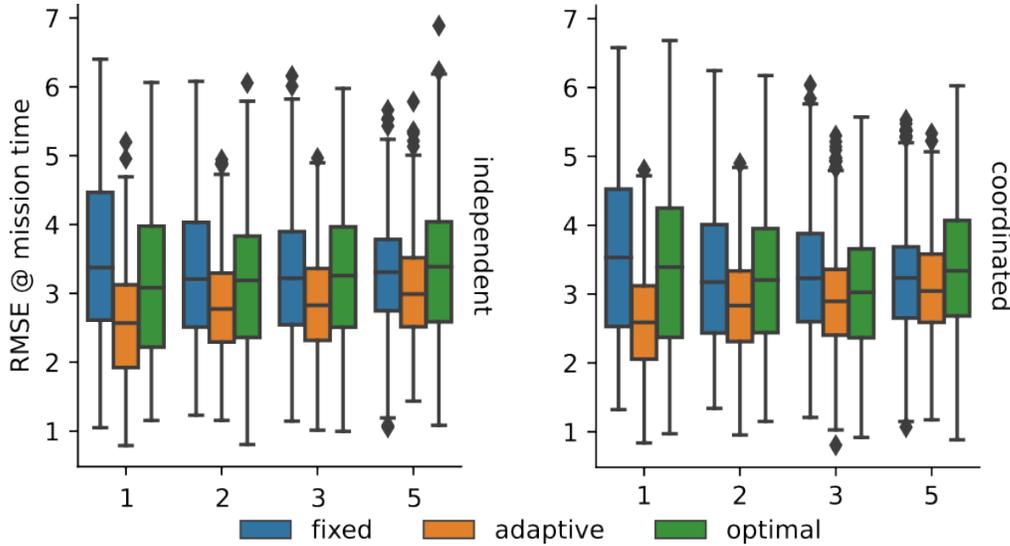


FIGURE 6.4: Ocean turbidity: Summary of average map error produced for different planner configurations and varying number of survey vehicles.

from the Moderate Resolution Imaging Spectroradiometer as a turbidity proxy [108]. Our simulated vehicles are configured with velocities congruent to the Wave Glider autonomous aquatic vehicle, based on the reported long-mission average speed of 1.5 knots (approximately 330 km per week) [122]. Figure 6.4 presents the results for different configuration of the coordinated and independent planners, configured with the spatiotemporal kernel. With a sampling rate of 1/week for a ~ 50 -week interval, inputs were relatively sparse and no distinguishing trends were observed between the performance of the cooperative and independent planners.

As with the synthetic simulation, a basis of comparison for the “optimal” fixed planner was established with hyperparameters collected by a single-vehicle lemniscatic coverage. As with the previous experiments, the adaptive planner results in improved map accuracy for all configurations across the survey envelope.

6.5 Conclusions

This chapter presented a novel integration of multi-vehicle informative path planning, informed by both model uncertainty and information from other sensing agents, which may differ in their contribution toward reducing model uncertainty. We also demonstrated how this approach could be utilized in an adaptive planning framework. Finally, we quantified the effectiveness of our planning approach, grounded in map accuracy, the salient objective for the survey operator.

As discussed in [section 4.5](#), it is reasonable to assume mobile robots will have access to a global, low-bandwidth communication channel (such as LoRa or other satellite communication systems). The consequences of relaxing this assumption is the natural next topic of investigation. In the next chapter, we will explore various dimensions of further study that stem from this and previous chapters.

This chapter illuminates multiple avenues for future work. Given global communication, each robot could directly integrate observations from all other robots, using a distributional or variational kernel. Robots could also attempt to project likely *future* movements by other agents in the team, using a similar utility metric grounded in model uncertainty. Observations collected in these experiments used a point sensor model, where the robot obtained a low-noise observation of the environment directly beneath the vehicle. Multi-task planning scenarios can incorporate variable sensor models, including variable noise, observation windows, and observation frequency.

7

Concluding Remarks

ISO 8373 - programmed actuated mechanism with a degree of autonomy, moving within its environment, to perform intended tasks

IEEE 1872 - An agentive device in a broad sense, purposed to act in the physical world in order to accomplish one or more tasks.

No Lorenzo, a washing machine is not a robot.

– various “definitions” for “robot”

7.1 Main conclusions

As with many dissertations in this domain, this work is motivated by some perceived automation need that can (in theory) be served by robotic methods. Specifically, I make the claim that there are some surveying objectives (especially whose where data collection is expensive) that benefit from *strategic sampling* approaches. Let’s review some practical contributions of this work:

This framework for generating surveys can be used to inform a variety of surveying efforts, including traditional surveys. The roboticist in me thinks that it is nice to envision an end-to-end autonomous surveying robot, where the operator just throws a robot onto the land, tosses a GeoJSON at its feet, and returns the following day with an environmental model with 5 nines of predictive accuracy. In reality, an small consultant might not be able to invest 50,000 USD in an “affordable” robot platform, but they do use a bluetooth GPS antenna and a smartphone to record samples that they take in the field. The planning algorithms described in this thesis could be used as a plugin to a map-based data logging mobile application, where the environmental model could be updated and maintained on a remote computer (read: the cloud), and the program could offer a suggestion about the next location that should be surveyed in order to improve the model’s predictive ability.

The methods used in this work use geostatistical fundamentals. So far, we have gotten to the final chapter without mentioning “artificial intelligence”. In the past decade, the rapid advancements in the use of convolutional neural networks for image recognition tasks have led to excitement regarding applications for earth observation and remote sensing tasks [123]. More recently, the advancements of vision-language models and other data-driven methods (of which, I will refer to broadly as “AI”) has led many to speculate and apply these tools toward remote sensing [124]. However, even before the mainstream success of generative AI, many researchers have identified the challenges of interpreting the results produced by these data-driven algorithms. When the input space and the model parameters are large, it can be very difficult to understand why a system has produced a particular answer, and the system can be likened to a ‘black box’ [125]. Even worse, one of the largest current limitations of generative AI systems are their tendency to confidently produce incorrect outputs for input queries that may have been outside of the training distribution [126]. These “hallucinations” may be inconvenient for a chat-bot, but when modeling earth and resource systems, hallucinated observations are unacceptable and un-scientific.

This dissertation deliberately avoided these “less-scrutable” data-driven approaches (like deep neural nets) and instead utilized well-understood principles derived from geostatistics. Of course, this does come with some limitations:

1. Predictions using GP regression are expensive to compute and the complexity of computing a posterior mean is $\mathcal{O}(n^3)$, since it involves a matrix inversion operation. However, since we only plan based on the posterior variance $\mathcal{O}(n^3)$ and we use additional constant-time iterative approximation methods that are very accurate [105], we are able to use this technique in an operational capacity for path planning that is quick enough to run on an actual robot.
2. GP regression tends to aggressively smooth variation in the underlying data. However, through the correct use of priors and kernels, we can construct a covariance function that is appropriate to the variation that exists in our prediction domain [72]. That said, the kernels used in this dissertation assume *stationarity* and *isotropy*. Therefore, the resultant models are poor at extrapolation, and regress to the mean very quickly when making predictions outside of the spatial area where there are observations. This could be improved upon by using different model and kernel formulations, discussed in the next subsection.

3. Since the planner is informed by the model, if the model has a low predictive ability, this can negatively impact the performance of the planner, which in-turn leads to observations which might not help improve the model's predictive ability. It remains to be seen if this produces a "negative-feedback", where the planner gets stuck in a local sub-optimum, whereupon it's difficult to improve the model. One could argue that *any* observation is better than *no* observation, and that the model will slowly converge toward the truth given enough time. A formal proof of this could be the topic of future work.

But one of the greatest advantages of using GP regression coupled with information-theoretic utility function is the ability to establish bounds on the expected worst-case accuracy of surveys constructed with the path planner.

During my Master's work [127], I set out to develop a model that could predict the amount of required by a crop in a given growing season, partitioned by rainwater and irrigated inputs. In the end, each prediction was accompanied by a sometimes wide margin of error, caused by uncertainties preset in every input that comprised the final calculation. It would be interesting to examine how uncertainty could be propagated through derivative calculations through automated surveying strategies that are built around uncertainty quantification. This leads to the final conclusion:

The utility functions used in this work are based in the qualities that are useful to surveyors and model designers. Some of the early papers in IPP performed planning based on an "information field", where the robot could query how much information is gained at an arbitrary location and derive a utility from that value. Of course, there are different ways that this "information field" could be constructed or derived. Similarly general are the IPP efforts that represent planner performance by the reduction of model variance. But this is a misleading metric— in [chapter 4](#) and [chapter 5](#), we explored how an improper choice of model parameters can lead to models with low variance, but low accuracy in the model predictions. For this reason, we evaluated all of our planners based on this *accuracy* of the final model using an error metric like RMSE. For users of a surveying system, they care more about how *accurate* the resulting model is, not something obtuse like "a reduction of the posterior variance". Granted, if the parameters for the GP model are selected in a way that maximizes the likelihood of the observed data (for example), then a model with low posterior variance should have a high predictive accuracy.

7.2 Future work

The major contribution of this body of work was the creation of a framework for informative path planning that unified the modeling task and the planning task within a shared statistical construct. In the past chapters, we explored several dimensions of how this framework can be applied toward: goal directed planning between waypoints, in a spatio-temporal modeling task, an adaptive planning task, and a multi-vehicle planning task. There are many variations, questions, and extensions that can branch from this main trunk of inquiry. Let us consider a few of these:

Unifying the path selection procedure using an information-theoretic selection criterion. In [chapter 3](#), we explored the use of a sampling-based planning approach to IPP, where the utility of a proposed location was calculated during the expansion step of the random tree. After the stopping criterion is met, we are still faced with the task of choosing a final path through the environment. Also in [chapter 3](#), we compared different path selection routines: an information-greedy approach, a cost-maximizing approach, and a heuristic path selection routine that balances between information gain and path length. A future study could explore and compare other heuristic or information-theoretic approaches at generating an optimal trajectory through this random tree. For example, the SOP-CC algorithm used in [chapter 4](#) could be applied to the tree variant of the RIG planner.

Informative path planning incorporating latent variables. In the geostatistics literature, *cokriging* is an interpolation technique that utilizes multiple correlated (latent) variables to improve the prediction of a primary variable of interest. By establishing a covariance relationship between variables of interest, a planner could construct a path to optimize for the modeling of an unseen variable. For example if a covariance relation is established between CO₂ flux and temperature/humidity, a planner could construct trajectories for measuring temperature/humidity that optimizes the predictive ability of the flux model. Planning for modeling of latent variables could be explored as extensions to any of the chapters discussed earlier.

Informative path planning with an expert prior. Kriging interpolation can incorporate a covariance relation that can be represented as an *empirical semivariogram*, constructed with a prior sampling campaign. This statistical construct could serve as a spatial prior and guide the form or hyperparameters of the covariance function used to calculate the expected information gain at a proposed sampling location. A future study could explore unifying this geostatistical practice with IPP.

Path planning with a variable sensor model. All of the experiments in this dissertation assume that the sensing vehicle travels to a location to make an observation at that location, before proceeding to a new location. This is what one would expect with a sensor that measures a phenomenon at a single point (i.e. probe-based sensor). Many remote sensors (such as UAV-mounted multispectral sensors) obtain multiple measurements over a wide area. This can be modeled as a sensor that obtains an array of point measurements over a horizontal footprint corresponding to the area imaged by the sensor, where the ground sampling distance (GSD) is proportional to the spatial resolution of the sensor. However, both the footprint and the GSD of aerial imaging sensors vary with the distance that the sensor is flown above the ground. An informative path planner could produce 3-dimensional trajectories, taking into consideration the footprint and spatial resolution of the measurements produced by a sensor as a function of its height. Additionally, the sensor might experience different noise characteristics as a function of its distance from the objects under observation. Dimensions of this problem were explored in [63] and can be extended to the sampling based approaches explored in this dissertation.

Path planning for a continuous sensor model. With the exception of Chapter 4, all of the experiments in this dissertation use a sampling based planner that plans in \mathbb{R}^2 . However in practice, observations are “snapped” to a coordinate grid, whose units are defined by the resolution of the spatial model. This is appropriate with point sampling schemes and with area and variable sensor models described in the previous paragraph. However, there are sensors that produce a continuous stream of data that can be discretized according to the surveying approach and objective. For example in towed Transient Electromagnetic Systems (tTEM), the sensing apparatus produces a continuous waveform that is transformed via a transfer function and discretized to produce a 3D reconstruction of subsurface phenomena (eg. soil composition and moisture) [128]. An informative planner could utilize the inversion algorithm (used to transform the waveform into the geo/hydrologic properties) as part of the utility function that guides the surveying platform to regions that result in a more accurate map. Such an IPP approach could use a discretizations of the signal in a variable signal model, or could utilize properties of the continuous signal to update the model covariance relations. There are many dimensions that could be explored for either approach.

The effect of variable communication modules for optimally of independent planning agents. In [chapter 4](#) and [chapter 6](#), we explored basic multi-vehicle scenarios where it is assumed that robots have the means to communicate small packets of positional information (24 bytes comprising two doubles for position and one for a timestamp). While, this may be a reasonable assumption

for low-bandwidth spread-spectrum protocols like LoRa, it would cease to hold for many other common communication protocols. There is a rich corpus of literature considering wireless sensor networks and communication protocols, and the multi-agent methods explored in this dissertation could be extended to consider unreliable communication scenarios. Of particular interest would be the effect on the accuracy of models generated from informed paths with partial communication ability. Additional extensions could consider dual objective planning for communication and information gathering.

Informative path planning for surveying objectives *other than regression.* One can envision an objective where instead of modeling some parameter in an environment, we wish to identify when and where some event of interest occurs. Motivating use cases include: patrol/surveillance tasks where we want to identify and locate a threat actor, species presence/absence modeling for ecological surveys, and source detection for contaminant plumes in pollution control and environmental remediation. Sampling based IPP methods discussed in this dissertation (and the extensions described in the above paragraphs) could be modified into classification tasks by modeling a probability-of-presence as the regressand, and passing the output into a response function such as in linear logistic regression.

Relaxing the isotropy, homoskedasticity and stationarity assumptions in the planning space. Future work can expand the planning task to consider more sophisticated modeling scenarios, such as: if there are hotspots in the planning space (anisotropy), if the parameter being modeled is trending (non-stationary), and/or if there is non-uniform variance in the world (heteroskedasticity).

7.3 Closing thoughts

In [chapter 1](#), we consider task of robotic surveying and frame it within a desire to obtain information about spatiotemporal systems. As I remember from my first spatial analysis and modeling class, almost *all data* taken in our world can have a spatial label attached to it. We are embodied in space and time and there will always be a desire to understand processes that exist on our world (or other “worlds” out there in outer space), and modeling processes is one way of obtaining those insights. So, does that mean “*more data, more better*”? Not really. Also as mentioned in [chapter 1](#), we collect more data presently than we know what to do with, and our insights are mostly limited by analytic ability, not a lack of observations.

Still, I want to make the claim that there are some resource management objectives that are *still* data-constrained—that is, they are constrained by a lack of

high quality data. For instance, while the dream is to be able to fly a sensor over a farm and understand everything that needs to be known about the plants, sometimes you need to be down at eye level to directly observe if there are pests present, in order to make a management decision before it is too late.

Now, this brings us to the realm of what is practical. As with much in the realm of engineering and especially for the domain of robotics, there exists a gulf between what might work in a lab demo and what can be put into practice for a business-critical operation. While this topic has been done to death in many different settings [129] [130], I still observe countless presentations (academic and commercial) that still breathlessly proclaim that industrial automation or agricultural robotics are critical in order to *X the next Y billion people*, while hand-waving away all the barriers to adoption.

Agriculture shares many of the constraints of other forms of physical production, including manufacturing, mining, and energy production. Advanced automation will only be implemented if the unit economics amount to a net benefit for the firm implementing the technology. While there are many interesting research questions that may or may not amount to an applicable method that makes economic sense, we must continue to develop and evaluate new robotic methods and stretch our imaginations to envision how might be applied to solve needs in present-day practices.

.1 Appendix B: Mathematical Notation and Terminology

The idea for this section came after observing that notational differences between texts in different disciplines (geostatistics vs machine learning vs computer science). This section clarifies the matrix terminology used in this thesis (drawn primarily from Rasmussen and Williams GP book [77]).

.1.1 Matrix terminology

From Chapter 3.3 of [131, p.107], with some additions:

- An $n \times \ell$ matrix $[A]$ is an array of $n\ell$ elements arranged in n rows and ℓ columns
 - A_{jk} denotes the k -th element in the j -th row.
 - Unless specified to the contrary, the elements are real numbers
- The transpose $[A^\top]$ of an $n \times \ell$ matrix $[A]$ is an $\ell \times n$ matrix $[B]$ with $B_{kj} = A_{jk}$ for all j, k
- A matrix is square if $n = \ell$ and a square matrix $[A]$ is symmetric if $[A] = [A]^\top$
- If $[A]$ and $[B]$ are each $n \times \ell$ matrices, $[A] + [B]$ is an $n \times \ell$ matrix $[C]$ with $C_{jk} = A_{jk} + B_{jk}$ for all j, k
- If $[A]$ is $n \times \ell$ and $[B]$ is $\ell \times r$, the matrix $[A][B]$ is an $n \times r$ matrix $[C]$ with elements $C_{jk} = \sum_i A_{ji}B_{ik}$
- A vector (or *column vector*) of dimension n is an $n \times 1$ matrix and a row vector of dimension n is a $1 \times n$ matrix
- Since the transpose of a vector is a row vector, we denote a vector a as $(a_1, \dots, a_n)^\top$
- Note about transposing vectors:
 - if a is a (column) vector of dimension n , then:
 - * aa^\top is the *outer product* produces an $n \times n$ matrix
 - * $a^\top a$ is the *inner product* (or dot product) of the two vectors, and produces a number
 - note that this happened when we apply the transpose to the first vector of the product (so the first vector becomes a row vector)

Bibliography

- [1] “Earth Science Missions: Current Fleet - NASA Science,” <https://science.nasa.gov/earth-science/missions/earth-missions-current/>.
- [2] V. Braitenberg, *Vehicles: Experiments in Synthetic Psychology*, 9th ed., ser. Bradford Book Psychology. Cambridge, Mass.: MIT Press, 2004.
- [3] S. M. LaValle, *Planning Algorithms*. Cambridge ; New York: Cambridge University Press, 2006.
- [4] M. J. Kochenderfer, *Decision Making under Uncertainty: Theory and Application*, ser. MIT Lincoln Laboratory Series. Cambridge (Mass.): The MIT press, 2015.
- [5] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, ser. Intelligent Robotics and Autonomous Agents. Cambridge, Mass: MIT Press, 2005.
- [6] D. S. Hochbaum and W. Maass, “Approximation schemes for covering and packing problems in image processing and VLSI,” *J. ACM*, vol. 32, no. 1, pp. 130–136, Jan. 1985.
- [7] H. González-Banos, “A randomized art-gallery algorithm for sensor placement,” in *Proceedings of the Seventeenth Annual Symposium on Computational Geometry*, ser. SCG '01. New York, NY, USA: Association for Computing Machinery, Jun. 2001, pp. 232–240.
- [8] W. R. Tobler, “A Computer Movie Simulating Urban Growth in the Detroit Region,” *Economic Geography*, vol. 46, pp. 234–240, 1970.
- [9] N. A. C. Cressie, *Statistics for Spatial Data*, rev. ed ed., ser. Wiley Series in Probability and Mathematical Statistics. New York: Wiley, 1993.
- [10] M. C. Shewry and H. P. Wynn, “Maximum entropy sampling,” *Journal of Applied Statistics*, vol. 14, no. 2, pp. 165–170, Jan. 1987.
- [11] S. Seo, M. Wallat, T. Graepel, and K. Obermayer, “Gaussian process regression: Active data selection and test point rejection,” in *Proceedings*

- of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, vol. 3, Jul. 2000, pp. 241–246 vol.3.
- [12] W. F. Caselton and J. V. Zidek, “Optimal monitoring network designs,” *Statistics & Probability Letters*, vol. 2, no. 4, pp. 223–227, Aug. 1984.
- [13] A. Krause, A. Singh, and C. Guestrin, “Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies,” *The Journal of Machine Learning Research*, vol. 9, pp. 235–284, Jun. 2008.
- [14] D. J. C. MacKay, “Information-Based Objective Functions for Active Data Selection,” *Neural Computation*, vol. 4, no. 4, pp. 590–604, Jul. 1992.
- [15] N. Ramakrishnan, C. Bailey-Kellogg, S. Tadepalli, and V. N. Pandey, “Gaussian Processes for Active Data Mining of Spatial Aggregates,” in *Proceedings of the 2005 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, Apr. 2005, pp. 427–438.
- [16] O. S. Adedaja, Y. Hamam, B. Khalaf, and R. Sadiku, “A state-of-the-art review of an optimal sensor placement for contaminant warning system in a water distribution network,” *Urban Water Journal*, vol. 15, no. 10, pp. 985–1000, Nov. 2018.
- [17] A. Jabini and E. A. Johnson, “A Deep Reinforcement Learning Approach to Sensor Placement under Uncertainty,” *IFAC-PapersOnLine*, vol. 55, no. 27, pp. 178–183, Jan. 2022.
- [18] A. Krause and C. Guestrin, “Nonmyopic active learning of Gaussian processes: An exploration-exploitation approach,” in *Proceedings of the 24th International Conference on Machine Learning - ICML '07*. Corvallis, Oregon: ACM Press, 2007, pp. 449–456.
- [19] A. Cameron and H. Durrant-Whyte, “A Bayesian Approach to Optimal Sensor Placement:,” *The International Journal of Robotics Research*, Oct. 1990.
- [20] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, “Efficient Informative Sensing using Multiple Robots,” *Journal of Artificial Intelligence Research*, vol. 34, pp. 707–755, Apr. 2009.
- [21] J. Binney and G. S. Sukhatme, “Branch and bound for informative path planning,” in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 2147–2154.

- [22] G. Hollinger and G. Sukhatme, "Sampling-based Motion Planning for Robotic Information Gathering," in *Robotics: Science and Systems IX*. Robotics: Science and Systems Foundation, Jun. 2013.
- [23] P. Vansteenwegen, W. Souffriau, and D. V. Oudheusden, "The orienteering problem: A survey," *European Journal of Operational Research*, vol. 209, no. 1, pp. 1–10, Feb. 2011.
- [24] B. L. Golden, L. Levy, and R. Vohra, "The orienteering problem," *Naval Research Logistics (NRL)*, vol. 34, no. 3, pp. 307–318, 1987.
- [25] I.-M. Chao, B. L. Golden, and E. A. Wasil, "The team orienteering problem," *European Journal of Operational Research*, vol. 88, no. 3, pp. 464–474, Feb. 1996.
- [26] A. M. Campbell, M. Gendreau, and B. W. Thomas, "The orienteering problem with stochastic travel and service times," *Annals of Operations Research*, vol. 186, no. 1, pp. 61–81, Jun. 2011.
- [27] T. C. Thayer and S. Carpin, "An Adaptive Method for the Stochastic Orienteering Problem," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 4185–4192, Apr. 2021.
- [28] —, "A Resolution Adaptive Algorithm for the Stochastic Orienteering Problem with Chance Constraints," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2021, pp. 6411–6418.
- [29] —, "A Fast Algorithm for Stochastic Orienteering with Chance Constraints," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2021, pp. 7961–7968.
- [30] J. Yu, M. Schwager, and D. Rus, "Correlated Orienteering Problem and its application to informative path planning for persistent monitoring tasks," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2014, pp. 342–349.
- [31] J. Yu, J. Aslam, S. Karaman, and D. Rus, "Anytime planning of optimal schedules for a mobile sensing robot," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2015, pp. 5279–5286.
- [32] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, second edition ed., ser. Adaptive Computation and Machine Learning Series. Cambridge, Massachusetts: The MIT Press, 2018.

- [33] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Machine Learning Proceedings 1994*. Elsevier, 1994, pp. 157–163.
- [34] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, Dec. 2018.
- [35] W. B. Powell and S. Meisel, "Tutorial on Stochastic Optimization in Energy I: Modeling and Policies."
- [36] M. Riedmiller, "Neural reinforcement learning to swing-up and balance a real pole," in *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, Oct. 2005, pp. 3191–3196 Vol. 4.
- [37] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, no. 1-2, pp. 99–134, May 1998.
- [38] J. A. Caley and G. A. Hollinger, "Data-driven comparison of spatio-temporal monitoring techniques," in *OCEANS 2015 - MTS/IEEE Washington*, Oct. 2015, pp. 1–7.
- [39] S. Carpin and T. C. Thayer, "Solving Stochastic Orienteering Problems with Chance Constraints Using Monte Carlo Tree Search," in *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, Aug. 2022, pp. 1170–1177.
- [40] V. Indelman, L. Carlone, and F. Dellaert, "Towards Planning in Generalized Belief Space," in *Robotics Research*, M. Inaba and P. Corke, Eds. Cham: Springer International Publishing, 2016, vol. 114, pp. 593–609.
- [41] C. Stachniss, G. Grisetti, and W. Burgard, "Information Gain-based Exploration Using Rao-Blackwellized Particle Filters," in *Robotics: Science and Systems I*. Robotics: Science and Systems Foundation, Jun. 2005.
- [42] S. Huang, N. Kwok, G. Dissanayake, Q. Ha, and G. Fang, "Multi-Step Look-Ahead Trajectory Planning in SLAM: Possibility and Necessity," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Apr. 2005, pp. 1091–1096.

- [43] G. Hitz, A. Gotovos, F. Pomerleau, M.-É. Garneau, C. Pradalier, A. Krause, and R. Y. Siegwart, "Fully autonomous focused exploration for robotic environmental monitoring," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 2658–2664.
- [44] R. N. Smith, P. Cooksey, F. Py, G. S. Sukhatme, and K. Rajan, "Adaptive Path Planning for Tracking Ocean Fronts with an Autonomous Underwater Vehicle," in *The 14th International Symposium on Experimental Robotics*, ser. Springer Tracts in Advanced Robotics. Springer, 2016, pp. 761–775.
- [45] T. Lewis and K. Bhaganagar, "A comprehensive review of plume source detection using unmanned vehicles for environmental sensing," *Science of The Total Environment*, vol. 762, p. 144029, Mar. 2021.
- [46] Y. Sung and P. Tokekar, "A Competitive Algorithm for Online Multi-Robot Exploration of a Translating Plume," *arXiv:1811.02769 [cs]*, Nov. 2018.
- [47] S. Tanaka, Y. Takei, K. Hirasawa, and H. Nanto, "An experimental study of 3D odor plume tracking using multicopter with gas sensor array," in *2015 IEEE SENSORS*, Nov. 2015, pp. 1–4.
- [48] A. J. Rutkowski, S. Edwards, M. A. Willis, R. D. Quinn, and G. C. Causey, "A robotic platform for testing moth-inspired plume tracking strategies," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 4, Apr. 2004, pp. 3319–3324 Vol.4.
- [49] Y. Takei, Y. Shimizu, K. Hirasawa, and H. Nanto, "Braitenberg's vehicle-like odor plume tracking robot," in *2014 IEEE SENSORS*, Nov. 2014, pp. 1276–1279.
- [50] J. N. Fuhg, A. Fau, and U. Nackenhurst, "State-of-the-Art and Comparative Review of Adaptive Sampling Methods for Kriging," *Archives of Computational Methods in Engineering*, vol. 28, no. 4, pp. 2689–2747, Jun. 2021.
- [51] G. Salhotra, C. E. Denniston, D. A. Caron, and G. S. Sukhatme, "Adaptive Sampling using POMDPs with Domain-Specific Considerations," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, May 2021, pp. 2385–2391.
- [52] J. Rückin, L. Jin, and M. Popović, "Adaptive Informative Path Planning Using Deep Reinforcement Learning for UAV-based Active Sensing," in

- 2022 *International Conference on Robotics and Automation (ICRA)*, May 2022, pp. 4473–4479.
- [53] 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*, vol. 1, Sep. 2002, pp. 540–545 vol.1.
- [54] A. Singh, A. R. Krause, C. Guestrin, W. J. Kaiser, and M. Batalin, “Efficient Planning of Informative Paths for Multiple Robots,” in *International Joint Conference on Artificial Intelligence (IJCAI)*, Hyderabad, India, Dec. 2006.
- [55] V. Suryan, “Learning a Spatial Field in Minimum Time With a Team of Robots,” Thesis, Virginia Polytechnic University, Blacksburg, Virginia, Aug. 2019.
- [56] W. Luo and K. Sycara, “Adaptive Sampling and Online Learning in Multi-Robot Sensor Coverage with Mixture of Gaussian Processes,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 6359–6364.
- [57] S. McCammon and G. A. Hollinger, “Topological path planning for autonomous information gathering,” *Autonomous Robots*, vol. 45, no. 6, pp. 821–842, Sep. 2021.
- [58] R. Cui, Y. Li, and W. Yan, “Mutual Information-Based Multi-AUV Path Planning for Scalar Field Sampling Using Multidimensional RRT*,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 7, pp. 993–1004, Jul. 2016.
- [59] V. Suryan and P. Tokekar, “Learning a Spatial Field in Minimum Time With a Team of Robots,” *IEEE Transactions on Robotics*, vol. 36, no. 5, pp. 1562–1576, Oct. 2020.
- [60] S. Kemna, J. G. Rogers, C. Nieto-Granda, S. Young, and G. S. Sukhatme, “Multi-robot coordination through dynamic Voronoi partitioning for informative adaptive sampling in communication-constrained environments,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 2124–2130.
- [61] A. Benevento, M. Santos, G. Notarstefano, K. Paynabar, M. Bloch, and M. Egerstedt, “Multi-Robot Coordination for Estimation and Coverage of Unknown Spatial Fields,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 7740–7746.

- [62] S. Choudhury, N. Gruver, and M. J. Kochenderfer, "Adaptive Informative Path Planning with Multimodal Sensing," *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 30, pp. 57–65, Jun. 2020.
- [63] M. Popović, T. Vidal-Calleja, G. Hitz, I. Sa, R. Siegwart, and J. Nieto, "Multiresolution mapping and informative path planning for UAV-Based terrain monitoring," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 1382–1388.
- [64] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg, "Near-optimal sensor placements: Maximizing information while minimizing communication cost," in *2006 5th International Conference on Information Processing in Sensor Networks*, Apr. 2006, pp. 2–10.
- [65] Y. Wei and R. Zheng, "Informative Path Planning for Mobile Sensing with Reinforcement Learning," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, Jul. 2020, pp. 864–873.
- [66] M. Kobilarov, "Cross-entropy motion planning," *The International Journal of Robotics Research*, vol. 31, no. 7, pp. 855–871, Jun. 2012.
- [67] J. R. Bourne, E. R. Pardyjak, and K. K. Leang, "Coordinated Bayesian-Based Bioinspired Plume Source Term Estimation and Source Seeking for Mobile Robots," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 967–986, Aug. 2019.
- [68] Y. T. Tan, A. Kunapareddy, and M. Kobilarov, "Gaussian Process Adaptive Sampling Using the Cross-Entropy Method for Environmental Sensing and Monitoring," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. Brisbane, QLD: IEEE, May 2018, pp. 6220–6227.
- [69] M. Popović, G. Hitz, J. Nieto, I. Sa, R. Siegwart, and E. Galceran, "Online informative path planning for active classification using UAVs," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 5753–5758.
- [70] O. Berger-Tal, J. Nathan, E. Meron, and D. Saltz, "The Exploration-Exploitation Dilemma: A Multidisciplinary Framework," *PLOS ONE*, vol. 9, no. 4, p. e95693, Apr. 2014.
- [71] E. Pebesma and R. Bivand, *Spatial Data Science: With Applications in R*, 1st ed. New York: Chapman and Hall/CRC, May 2023.

- [72] M. L. Stein, *Interpolation of Spatial Data: Some Theory for Kriging*, ser. Springer Series in Statistics. New York: Springer-Verlag, 1999.
- [73] D. G. Krige, "A statistical approach to some mine valuation and allied problems on the Witwatersrand," Thesis, University of the Witwatersrand, Johannesburg, South Africa, Mar. 1951.
- [74] G. Matheron, "Principles of geostatistics," *Economic Geology*, vol. 58, no. 8, pp. 1246–1266, Dec. 1963.
- [75] N. Cressie, "The origins of kriging," *Mathematical Geology*, vol. 22, no. 3, pp. 239–252, Apr. 1990.
- [76] A. O'Hagan, "Curve Fitting and Optimal Design for Prediction," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 40, no. 1, pp. 1–42, 1978.
- [77] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, ser. Adaptive Computation and Machine Learning. Cambridge, Mass: MIT Press, 2006.
- [78] G. A. Hollinger and G. S. Sukhatme, "Sampling-based robotic information gathering algorithms," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1271–1287, Aug. 2014.
- [79] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, Jun. 2011.
- [80] —, "Incremental Sampling-Based Algorithms for Optimal Motion Planning," May 2010.
- [81] A. Krause and C. Guestrin, "Submodularity and its applications in optimized information gathering," *ACM Transactions on Intelligent Systems and Technology (TIST)*, Jul. 2011.
- [82] C. Guestrin, A. Krause, and A. P. Singh, "Near-optimal sensor placements in Gaussian processes," in *Proceedings of the 22nd International Conference on Machine Learning - ICML '05*. Bonn, Germany: ACM Press, 2005, pp. 265–272.
- [83] M. G. Jadidi, J. V. Miro, and G. Dissanayake, "Sampling-based incremental information gathering with applications to robotic exploration and environmental monitoring:," *The International Journal of Robotics Research*, vol. 38, no. 6, pp. 658–685, Apr. 2019.

- [84] M. Popovic, T. Vidal-Calleja, J. J. Chung, J. Nieto, and R. Siegwart, "Informative Path Planning for Active Field Mapping under Localization Uncertainty," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. Paris, France: IEEE, May 2020, pp. 10 751–10 757.
- [85] A. Sakai, D. Ingram, J. Dinius, K. Chawla, A. Raffin, and A. Paques, "PythonRobotics: A Python code collection of robotics algorithms," Aug. 2018.
- [86] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [87] OpenStreetMap contributors, "Planet dump retrieved from <https://planet.osm.org/>," 2017.
- [88] S. Toledo, "Locality of Reference in LU Decomposition with Partial Pivoting," *SIAM Journal on Matrix Analysis and Applications*, Jul. 2006.
- [89] E. Anderson, Z. Bai, C. Bischof, L. S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, ser. Software, Environments and Tools. Society for Industrial and Applied Mathematics, Jan. 1999.
- [90] R. Sim and N. Roy, "Global A-Optimal Robot Exploration in SLAM," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Apr. 2005, pp. 661–666.
- [91] T. C. Thayer, S. Vougioukas, K. Goldberg, and S. Carpin, "Multirobot Routing Algorithms for Robots Operating in Vineyards," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 3, pp. 1184–1194, Jul. 2020.
- [92] G. A. Di Caro and A. W. Z. Yousaf, "Multi-robot Informative Path Planning using a Leader-Follower Architecture," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, May 2021, pp. 10 045–10 051.
- [93] J. P. Shanmuga Sundaram, W. Du, and Z. Zhao, "A Survey on LoRa Networking: Research Problems, Current Solutions, and Open Issues," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 371–388, 2020.

- [94] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 3250–3265, May 2012.
- [95] E. Contal, V. Perchet, and N. Vayatis, "Gaussian Process Optimization with Mutual Information," in *Proceedings of the 31st International Conference on Machine Learning*. PMLR, Jun. 2014, pp. 253–261.
- [96] G. A. Di Caro and A. W. Ziaullah Yousaf, "Map Learning via Adaptive Region-Based Sampling in Multi-robot Systems," in *Distributed Autonomous Robotic Systems*, F. Matsuno, S.-i. Azuma, and M. Yamamoto, Eds. Cham: Springer International Publishing, 2022, pp. 335–348.
- [97] A. Vettraino, A. Roques, A. Yart, J.-t. Fan, J.-h. Sun, and A. Vannini, "Sentinel Trees as a Tool to Forecast Invasions of Alien Plant Pathogens," *PLOS ONE*, vol. 10, no. 3, p. e0120571, Mar. 2015.
- [98] O. Hamelijncx, W. J. Wilkinson, N. A. Loppi, A. Solin, and T. Damoulas, "Spatio-Temporal Variational Gaussian Processes," in *Advances in Neural Information Processing Systems*, Nov. 2021.
- [99] L. Booth and S. Carpin, "Distributed estimation of scalar fields with implicit coordination," in *Distributed Autonomous Robotic Systems*, ser. Springer Proceedings in Advanced Robotics. Monbellard, FR: Springer International Publishing, 2023.
- [100] P. Goovaerts, *Geostatistics for Natural Resources Evaluation*, ser. Applied Geostatistics. Oxford, New York: Oxford University Press, Sep. 1997.
- [101] M. A. Álvarez and N. D. Lawrence, "Computationally Efficient Convolved Multiple Output Gaussian Processes," *Journal of Machine Learning Research*, vol. 12, no. 41, pp. 1459–1500, 2011.
- [102] W. Bruinsma, E. Perim, W. Tebbutt, S. Hosking, A. Solin, and R. Turner, "Scalable Exact Inference in Multi-Output Gaussian Processes," in *Proceedings of the 37th International Conference on Machine Learning*. PMLR, Nov. 2020, pp. 1190–1201.
- [103] A. G. Wilson, E. Gilboa, A. Nehorai, and J. P. Cunningham, "Fast kernel learning for multidimensional pattern extrapolation," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, Dec. 2014, pp. 3626–3634.

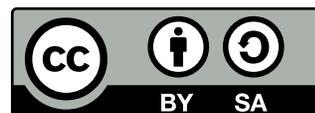
- [104] S. R. Flaxman, "Machine learning in space and time," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, Aug. 2015.
- [105] G. Pleiss, J. Gardner, K. Weinberger, and A. G. Wilson, "Constant-Time Predictive Distributions for Gaussian Processes," in *Proceedings of the 35th International Conference on Machine Learning*. PMLR, Jul. 2018, pp. 4114–4123.
- [106] J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson, "GPpyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS'18. Red Hook, NY, USA: Curran Associates Inc., Dec. 2018, pp. 7587–7597.
- [107] J. Wagberg, D. Zachariah, T. Schon, and P. Stoica, "Prediction Performance After Learning in Gaussian Process Regression," in *Artificial Intelligence and Statistics*. PMLR, Apr. 2017, pp. 1264–1272.
- [108] N. O. B. P. Group, "MODIS-Aqua Level 3 Mapped Particulate Organic Carbon Data Version R2018.0," 2017.
- [109] J. Kim, S. Kim, C. Ju, and H. I. Son, "Unmanned Aerial Vehicles in Agriculture: A Review of Perspective of Platform, Control, and Applications," *IEEE Access*, vol. 7, pp. 105 100–105 115, 2019.
- [110] F. Esser, R. A. Rosu, A. Cornelißen, L. Klingbeil, H. Kuhlmann, and S. Behnke, "Field Robot for High-Throughput and High-Resolution 3D Plant Phenotyping: Towards Efficient and Sustainable Crop Production," *IEEE Robotics & Automation Magazine*, vol. 30, no. 4, pp. 20–29, Dec. 2023.
- [111] M. Campbell, A. Dechemi, and K. Karydis, "An Integrated Actuation-Perception Framework for Robotic Leaf Retrieval: Detection, Localization, and Cutting," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2022, pp. 9210–9216.
- [112] J. Binney, A. Krause, and G. S. Sukhatme, "Optimizing waypoints for monitoring spatiotemporal phenomena," *The International Journal of Robotics Research*, vol. 32, no. 8, pp. 873–888, Jul. 2013.
- [113] J. Ott, E. Balaban, and M. J. Kochenderfer, "Sequential Bayesian Optimization for Adaptive Informative Path Planning with Multimodal Sensing," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, May 2023, pp. 7894–7901.

- [114] M. Popović, J. Ott, J. Rückin, and M. J. Kochenderfer, "Learning-based methods for adaptive informative path planning," *Robotics and Autonomous Systems*, vol. 179, p. 104727, Sep. 2024.
- [115] S. Yanes Luis, M. Perales Esteve, D. Gutiérrez Reina, and S. Toral Marín, "Deep Reinforcement Learning Applied to Multi-agent Informative Path Planning in Environmental Missions," in *Mobile Robot: Motion Control and Path Planning*, A. T. Azar, I. Kasim Ibraheem, and A. Jaleel Humaidi, Eds. Cham: Springer International Publishing, 2023, pp. 31–61.
- [116] G. Hitz, E. Galceran, M.-È. Garneau, F. Pomerleau, and R. Siegwart, "Adaptive continuous-space informative path planning for online environmental monitoring," *Journal of Field Robotics*, vol. 34, no. 8, pp. 1427–1449, 2017.
- [117] Y. Kantaros, B. Schlotfeldt, N. Atanasov, and G. J. Pappas, "Asymptotically Optimal Planning for Non-Myopic Multi-Robot Information Gathering," in *Robotics: Science and Systems XV*. Robotics: Science and Systems Foundation, Jun. 2019.
- [118] M. A. Schack, J. G. Rogers, Q. Han, and N. Dantam, "Optimizing Non-Markovian Information Gain Under Physics-Based Communication Constraints," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4813–4819, Jul. 2021.
- [119] S. M. Lavalle, "Rapidly-Exploring Random Trees: A New Tool for Path Planning," Iowa State University, Computer Science Department, Tech. Rep., Oct. 1998.
- [120] L. A. Booth and S. Carpin, "Informative path planning for scalar dynamic reconstruction using coregionalized Gaussian processes and a spatiotemporal kernel," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2023.
- [121] M. A. Álvarez, L. Rosasco, and N. D. Lawrence, "Kernels for Vector-Valued Functions: A Review," *Foundations and Trends in Machine Learning*, vol. 4, no. 3, pp. 195–266, 2012.
- [122] R. N. Smith, J. Das, G. Hine, W. Anderson, and G. S. Sukhatme, "Predicting Wave Glider speed from environmental measurements," in *OCEANS'11 MTS/IEEE KONA*, Sep. 2011, pp. 1–8.
- [123] T. Kattenborn, J. Leitloff, F. Schiefer, and S. Hinz, "Review on Convolutional Neural Networks (CNN) in vegetation remote sensing,"

- ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 173, pp. 24–49, Mar. 2021.
- [124] B. Janga, G. P. Asamani, Z. Sun, and N. Cristea, “A Review of Practical AI for Remote Sensing in Earth Sciences,” *Remote Sensing*, vol. 15, no. 16, p. 4112, Jan. 2023.
- [125] D. Castelvechi, “Can we open the black box of AI?” *Nature News*, vol. 538, no. 7623, p. 20, Oct. 2016.
- [126] C. S. Smith, “Hallucinations Could Blunt ChatGPT’s Success,” *IEEE Spectrum*, Mar. 2023.
- [127] L. Booth, “Characterizing the spatial-temporal distribution of California’s agricultural water utilization using a water footprint analysis in R,” Master’s thesis, UC Merced, 2018.
- [128] E. Auken, N. Foged, J. J. Larsen, K. V. T. Lassen, P. K. Maurya, S. M. Dath, and T. T. Eiskjær, “tTEM — A towed transient electromagnetic system for detailed 3D imaging of the top 70 m of the subsurface,” *GEOPHYSICS*, vol. 84, no. 1, pp. E13–E22, Jan. 2019.
- [129] B. Vanderborcht, “Robotic Dreams, Robotic Realities: Why Is It So Hard to Build Profitable Robot Companies?” *IEEE Spectrum*, Mar. 2019.
- [130] R. Brooks, “Rodney Brooks’ Three Laws of Robotics – Rodney Brooks,” Jul. 2024.
- [131] R. G. Gallager, *Stochastic Processes: Theory for Applications*. Cambridge, United Kingdom ; New York: Cambridge University Press, 2013.

The format of this thesis is derived from “Masters/Doctoral Thesis LaTeX Template Version 2.1 (2/9/15)”¹

This work is licensed under a [Creative Commons](https://creativecommons.org/licenses/by-sa/4.0/) “Attribution-ShareAlike 4.0 International” license.



¹V2: vel@latextemplates.com, Johannes Böttcher. V1: Steven Gunn, Sunil Patel