

UNIVERSITY OF CALIFORNIA
SANTA CRUZ

**AUTONOMOUS FIELD EXPLORATION USING THE KRIGING
METHOD**

A thesis submitted in partial satisfaction of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

with an emphasis in ROBOTICS AND CONTROL

by

Sargis S Yonan

December 2018

The Dissertation of Sargis S Yonan
is approved:

Professor Gabriel Hugh Elkaim, Chair

Professor Renwick E. Curry

Professor Patrick E. Mantey

Lori Kletzer
Vice Provost and Dean of Graduate Studies

Copyright © by

Sargis S Yonan

2018

Table of Contents

List of Figures	vi
Abstract	1
Dedication	2
Acknowledgments	3
1 Introduction	4
I Background	7
2 Problem Definitions	8
2.1 The Field	8
2.2 The Sensor	9
2.2.1 Real World Sensing Examples	9
3 Previous Work	10
II Spatial Analysis	12
4 Introduction to Spatial Analysis	13
4.0.1 Autocorrelation in a Field	14
4.1 Inverse Distance Weighting	15
4.2 Variography	17
4.2.1 The Variogram	17
4.2.2 The Semivariogram	18
4.2.3 The Empirical Semivariogram	19
4.2.4 Converting a Semivariogram to a Variogram	19
4.2.5 Variogram Models	20
4.2.6 Fitting A Semi-Variogram	21

5 The Kriging Method	24
5.1 Forms of the Kriging Method	24
5.2 Covariance Matrix From A Variogram	25
5.3 The Proximity Vector	25
5.4 The Kriging Weights	26
5.5 The Kriging Prediction Equation	26
5.6 Variance of A Kriging Prediction	28
5.7 The Kriging Error	28
5.8 Procedure For Field Prediction Using The Kriging Method	28
III Autonomous Field Exploration Using The Kriging Method	30
6 Vehicle & Information Gain Model	31
6.1 Exploration Vehicle Model Dynamics	31
6.2 Field Uncertainty Model	32
7 Path Planning	34
7.1 Uncertainty Loss Function	34
7.2 Finding Points of Highest Uncertainty	36
7.3 Next Highest Variance (NHV) Trajectory Finding	37
7.3.1 NHV Path Planning Algorithm	37
7.3.2 Inefficiency in NHV	39
7.4 Monte Carlo Based Trajectory Finding	39
7.4.1 Finding Monte Carlo Trajectory Sets	40
7.4.2 Selecting An Optimal Trajectory (Prediction Feedback)	41
7.4.3 Monte Carlo Path Planning Algorithm	42
7.4.4 N Next Highest Variances Algorithm (Zero Variance MCPP)	44
7.4.5 Benefits in MCPP over NHV	44
7.4.6 Disadvantages of The Monte Carlo Path Planner	45
IV Simulation & Results	47
8 Simulation Framework	48
8.1 Simulated Exploration Vehicle Model Dynamics	49
8.2 Generating a Target Field	49
8.3 Simulation Environment	50
9 Results	54
9.1 Result Comparison Criteria	54
9.1.1 Variance Drop Calculation	55
9.1.2 Simulation Results	56
9.1.3 Simulation Result Parameters	56
9.1.4 Prediction Error Calculation	57

9.2	High Spatial Autocorrelation ($\sigma_{field} = w$) Results	57
9.2.1	10% Maximum Field Scan	58
9.2.2	20% Maximum Field Scan	60
9.2.3	30% Maximum Field Scan	62
9.3	Half Width Spatial Autocorrelation ($\sigma_{field} = 0.5w$) Results	63
9.3.1	10% Maximum Field Scan	64
9.3.2	20% Maximum Field Scan	66
9.3.3	30% Maximum Field Scan	68
9.4	Quarter Width Spatial Autocorrelation ($\sigma_{field} = 0.25w$) Results	69
9.4.1	10% Maximum Field Scan	70
9.4.2	20% Maximum Field Scan	72
9.4.3	30% Maximum Field Scan	74
9.5	Low Spatial Autocorrelation ($\sigma_{field} = 1$) Results	75
9.5.1	10% Maximum Field Scan	76
9.5.2	20% Maximum Field Scan	78
9.5.3	30% Maximum Field Scan	80
9.6	Comparing The Methods	81
End Matter		84
Bibliography		86
A Results (Random Seed - 3)		88
A.1	High Spatial Autocorrelation ($\sigma_{field} = w$) Results	88
A.1.1	10% Maximum Field Scan	89
A.1.2	20% Maximum Field Scan	91
A.1.3	30% Maximum Field Scan	93
A.2	Half Width Spatial Autocorrelation ($\sigma_{field} = 0.5w$) Results	94
A.2.1	10% Maximum Field Scan	95
A.2.2	20% Maximum Field Scan	97
A.2.3	30% Maximum Field Scan	99
A.3	Quarter Width Spatial Autocorrelation ($\sigma_{field} = 0.25w$) Results	100
A.3.1	10% Maximum Field Scan	101
A.3.2	20% Maximum Field Scan	103
A.3.3	30% Maximum Field Scan	105
A.4	Low Spatial Autocorrelation ($\sigma_{field} = 1$) Results	106
A.4.1	10% Maximum Field Scan	107
A.4.2	20% Maximum Field Scan	109
A.4.3	30% Maximum Field Scan	111

List of Figures

4.1	A Gaussian distributed randomly generated spatially autocorrelated field.	15
4.2	An inverse distance weighting predicted field generated from the samples taken of Figure 4.1a at the locations marked in Figure 4.1b.	16
4.3	An empirical semivariogram.	20
4.4	Examples of three different variogram models.	22
4.5	An experimental variogram generated using Equation 4.6 from the samples taken in Figure 4.1b. δ was chosen such that for n observations, a total number of $\left\lfloor \frac{n}{2} \right\rfloor$ points were plotted. A Gaussian statistical model was fit to the experimental variogram. The variogram was fit using <i>fminsearchcon</i> in <i>MATLAB</i>	23
5.1	A Kriging Method predicted field generated from the samples taken of Figure 4.1a at the locations marked in Figure 4.1b.	27
7.1	The Next Highest Variance (N HV) algorithm (Section 7.3.1 terminated after two iterations of the algorithm. The actual field that is being explored is shown in the upper left. The current prediction of the actual field is shown in the upper right. The variance of the current prediction of the field is shown in the lower left. The trace of the exploration vehicle's path taken to the point of termination (in red) is shown in the lower right panel. All distance units are in meters.	38
7.2	Brownian non-deterministic walks (black) surrounding deterministic cords (red). The starting point is indicated in green. For a set of $N = 2$ possible trajectories, 15 random walks around each trajectory with a variance of 10. All distance units in meters.	41

7.3	The Next Highest Variance (NHV) algorithm (Section 7.3.1 terminated after two iterations of the algorithm. The actual field that is being explored is shown in the upper left. The current prediction of the actual field is shown in the upper right. The variance of the current prediction of the field is shown in the lower left. The trace of the exploration vehicle's path taken to the point of termination (in red) is shown in the lower right panel. All distance units are in meters. Here, N , the number of endpoints selected is 3, and the number of random paths per cord calculated is 15.	43
8.1	A field is generated using a zero mean normal distribution with variance 1. Varying degrees of spatial autocorrelation are shown for different values of σ_{field}	50
8.2	Next Highest Variance Path Planning. The actual field (top left), predicted field (with error) (top right), variance field (bottom left), and traversed path (bottom right). The path planner is conducted on the same field with an approximate 3% scan area limit.	51
8.3	Monte Carlo path planner run ($N = 5$, 20 trajectories per path). The actual field (top left), predicted field (with error) (top right), variance field (bottom left), and traversed path (bottom right). The path planner is conducted on the same field with an approximate 3% scan area limit.	52
8.4	Zig-Zag method. The actual field (top left), predicted field (with error) (top right), variance field (bottom left), and traversed path (bottom right). The path planner is conducted on the same field with an approximate 3% scan area limit.	53
9.1	Zig-Zag exploration method set to scan a fixed percentage of a target field. A Kriging prediction and variance calculation is computed after completing the maneuver. The actual field that is being explored is shown in the upper left. The current prediction of the actual field is shown in the upper right. The variance of the current prediction of the field is shown in the lower left. The trace of the exploration vehicle's path taken to the point of termination (in red) is shown in the lower right panel. All distance units are in meters.	55
9.2	A 10% maximum area scan on a field of size 100×100 , $\sigma_{field} = 100$, random seed: 2.	58
9.3	Simulation output for a 10% maximum area scan on a field of size 100×100 , $\sigma_{field} = 100$, random seed: 2.	59
9.4	A 20% maximum area scan on a field of size 100×100 , $\sigma_{field} = 100$, random seed: 2.	60
9.5	Simulation output for a 20% maximum area scan on a field of size 100×100 , $\sigma_{field} = 100$, random seed: 2.	61
9.6	A 30% maximum area scan on a field of size 100×100 , $\sigma_{field} = 100$, random seed: 2.	62
9.7	Simulation output for a 30% maximum area scan on a field of size 100×100 , $\sigma_{field} = 100$, random seed: 2.	63

9.8	A 10% maximum area scan on a field of size 100×100 , $\sigma_{field} = 50$, random seed: 2	64
9.9	Simulation output for a 10% maximum area scan on a field of size 100×100 , $\sigma_{field} = 50$, random seed: 2	65
9.10	A 20% maximum area scan on a field of size 100×100 , $\sigma_{field} = 50$, random seed: 2	66
9.11	Simulation output for a 20% maximum area scan on a field of size 100×100 , $\sigma_{field} = 50$, random seed: 2	67
9.12	A 30% maximum area scan on a field of size 100×100 , $\sigma_{field} = 50$, random seed: 2	68
9.13	Simulation output for a 30% maximum area scan on a field of size 100×100 , $\sigma_{field} = 50$, random seed: 2	69
9.14	A 10% maximum area scan on a field of size 100×100 , $\sigma_{field} = 25$, random seed: 2	70
9.15	Simulation output for a 10% maximum area scan on a field of size 100×100 , $\sigma_{field} = 25$, random seed: 2	71
9.16	A 20% maximum area scan on a field of size 100×100 , $\sigma_{field} = 25$, random seed: 2	72
9.17	Simulation output for a 20% maximum area scan on a field of size 100×100 , $\sigma_{field} = 25$, random seed: 2	73
9.18	A 30% maximum area scan on a field of size 100×100 , $\sigma_{field} = 25$, random seed: 2	74
9.19	Simulation output for a 30% maximum area scan on a field of size 100×100 , $\sigma_{field} = 25$, random seed: 2	75
9.20	A 10% maximum area scan on a field of size 100×100 , $\sigma_{field} = 1$, random seed: 2	76
9.21	Simulation output for a 10% maximum area scan on a field of size 100×100 , $\sigma_{field} = 1$, random seed: 2	77
9.22	A 20% maximum area scan on a field of size 100×100 , $\sigma_{field} = 1$, random seed: 2	78
9.23	Simulation output for a 20% maximum area scan on a field of size 100×100 , $\sigma_{field} = 1$, random seed: 2	79
9.24	A 30% maximum area scan on a field of size 100×100 , $\sigma_{field} = 1$, random seed: 2	80
9.25	Simulation output for a 30% maximum area scan on a field of size 100×100 , $\sigma_{field} = 1$, random seed: 2	81
A.1	A 10% maximum area scan on a field of size 100×100 , $\sigma_{field} = 100$, random seed: 2	89
A.2	Simulation output for a 10% maximum area scan on a field of size 100×100 , $\sigma_{field} = 100$, random seed: 2	90
A.3	A 20% maximum area scan on a field of size 100×100 , $\sigma_{field} = 100$, random seed: 2	91
A.4	Simulation output for a 20% maximum area scan on a field of size 100×100 , $\sigma_{field} = 100$, random seed: 2	92

A.5 A 30% maximum area scan on a field of size 100×100 , $\sigma_{field} = 100$, random seed: 2.	93
A.6 Simulation output for a 30% maximum area scan on a field of size 100×100 , $\sigma_{field} = 100$, random seed: 2.	94
A.7 A 10% maximum area scan on a field of size 100×100 , $\sigma_{field} = 50$, random seed: 2.	95
A.8 Simulation output for a 10% maximum area scan on a field of size 100×100 , $\sigma_{field} = 50$, random seed: 2.	96
A.9 A 20% maximum area scan on a field of size 100×100 , $\sigma_{field} = 50$, random seed: 2.	97
A.10 Simulation output for a 20% maximum area scan on a field of size 100×100 , $\sigma_{field} = 50$, random seed: 2.	98
A.11 A 30% maximum area scan on a field of size 100×100 , $\sigma_{field} = 50$, random seed: 2.	99
A.12 Simulation output for a 30% maximum area scan on a field of size 100×100 , $\sigma_{field} = 50$, random seed: 2.	100
A.13 A 10% maximum area scan on a field of size 100×100 , $\sigma_{field} = 25$, random seed: 2.	101
A.14 Simulation output for a 10% maximum area scan on a field of size 100×100 , $\sigma_{field} = 25$, random seed: 2.	102
A.15 A 20% maximum area scan on a field of size 100×100 , $\sigma_{field} = 25$, random seed: 2.	103
A.16 Simulation output for a 20% maximum area scan on a field of size 100×100 , $\sigma_{field} = 25$, random seed: 2.	104
A.17 A 30% maximum area scan on a field of size 100×100 , $\sigma_{field} = 25$, random seed: 2.	105
A.18 Simulation output for a 30% maximum area scan on a field of size 100×100 , $\sigma_{field} = 25$, random seed: 2.	106
A.19 A 10% maximum area scan on a field of size 100×100 , $\sigma_{field} = 1$, random seed: 2.	107
A.20 Simulation output for a 10% maximum area scan on a field of size 100×100 , $\sigma_{field} = 1$, random seed: 2.	108
A.21 A 20% maximum area scan on a field of size 100×100 , $\sigma_{field} = 1$, random seed: 2.	109
A.22 Simulation output for a 20% maximum area scan on a field of size 100×100 , $\sigma_{field} = 1$, random seed: 2.	110
A.23 A 30% maximum area scan on a field of size 100×100 , $\sigma_{field} = 1$, random seed: 2.	111
A.24 Simulation output for a 30% maximum area scan on a field of size 100×100 , $\sigma_{field} = 1$, random seed: 2.	112

Abstract

Autonomous Field Exploration Using The Kriging Method

by

Sargis S Yonan

A set of methods for the exploration of unknown semi-to-fully-ergodic fields of interest are introduced. Using observations of a single state of interest from an autonomous exploration vehicle with turn rate control, a field of interest can be learned more accurately and efficiently versus preplanned scanning techniques.

The Kriging Method, a *Best Linear Unbiased Predictor* (BLUP) commonly used in the field of Geospatial Analysis, is used to exploit the statistical properties, namely the geospatial autocorrelation, of a target field. The Kriging Method predicts the state of unobserved points from a set of observed points. A prediction and confidence of prediction of the entirety of a given target field can be generated from the method. From the variances associated with the predictions made by the Kriging Method, a set of path planning methods for an autonomous exploration vehicle will be introduced for the purposes of field exploration.

The path planners can be used to reduce the overall uncertainty of field predictions by steering a single vehicle through the field to optimally collect a good set of samples to make a field prediction from. A metric for return on investment of executing a trajectory using feedback from Kriging predictions is introduced as well. The three path planners introduced all aim to suppress the overall uncertainty of a Kriging prediction of an unknown generic target field of interest in order to create a variable quality map.

TODO

Acknowledgments

TODO

Chapter 1

Introduction

Field exploration is a method in which an unknown field (a *target field*) is learned in an attempt to discover traits or track trends about the field. An exploration method is one where a model of the target field phenomena is not necessarily known, but can be learned. Field exploration methods can be useful for tracking the health of crop soil, the size of ice glaciers, generating terrain maps, and a wide variety of scientific and industrial purposes. Furthermore, an exploration technique, versus a patrolling or tracking technique, where a target is tracked or surveilled, does not require a model of the target field dynamics, as they can be learned on-the-fly. This means that a variety of fields can be explored without the knowledge of the initial state of the field, and in the case of an ergodic field, its dynamics.

Using an Unmanned Aerial Vehicle (UAV) system, for example, an unknown field of interest can be scanned within a more reasonable time frame compared to conventional scanning techniques involving satellite and manned-airplane missions. Potentially more nuanced data can be gathered from the UAV made observations because of more desirable

fields of view and more customizable sensors on-board. Using the techniques introduced, a high-quality map can be generated of a previously unknown field of interest. Satellite imagery of Earth has been used for measuring various natural phenomena in the past several decades. Estimating polar ice cap melting rates and exploring the location of an oil spills are among the class of problems solved by this technology. Currently, using a service like the US Forest Services' Moderate Resolution Imaging Spectroradiometer (MODIS) Active Fire Mapping Program, images are updated every 1 to 2 days with a fixed sensor. While this program is helpful for detecting large events with long periods of activity, the sampling rate of this service might not give an emergency response team or a scientist the required resolution and precision in gathered data at their desired rate. The resolution and frequency problem along with the cost associated with building, launching, and maintaining an orbiting Earth satellite might even make some areas of research prohibitive. The use of unmanned aerial vehicles (UAVs) have more recently been used in similar fields of study and in environmental protection. The benefits gained from using UAVs is that of more rapidly acquired data with more easily adjusted accuracy. A UAV can give more nuanced and detailed data on features of a field that are not observable from the distance or field of view of an orbiting satellite. This is because the UAV can be equipped with any compatible sensor and can be deployed from virtually anywhere to fly virtually anywhere.

Presently, a common approach to exploring a field is to conduct a predetermined zig-zag, lawn mower pattern, or other predetermined maneuver on a target field. This task might take longer than needed to collect the required data, and could potentially ineffectively use the flight or drive time of the exploration vehicle which often has a short and limited runtime.

Furthermore, scanning every point in a large unknown field is an unrealistic expectation for vehicles with limited maneuvering capabilities. This is especially a problem if the field as a whole is very large and needs to only be predicted to a small degree of confidence. A scheme for minimal and high-quality scanning via variance motivated path planning would be in the benefit of time for the user(s) of the system, and the scanning equipment as well.

Due to the nature of much of the phenomena one might be interested in scanning in an unknown field, a method that exploits the known stochastic properties of the field could be used to decrease exploration time. A field that exhibits properties of *geospatial autocorrelation* would be more statistically exploitable. In an effort to avoid scanning more points than needed, the unobserved points in the field must therefore be predicted. These fields exhibit properties that make finding patterns in field state distributions simpler to compute. The Kriging Method, a popular interpolation tool, offers a prediction and a variance of prediction for points in a geospatially autocorrelated field. By exploiting the Kriging variances generated by the predictions, variance based path planning methods can be used to steer an exploration vehicle in the areas of maximal uncertainty, while traversing over other areas of low prediction confidence. The methods introduced attempt to help a user of this system explore an unknown field with a known degree of confidence that is configurable through a desired runtime, tuned by the user.

Part I

Background

Chapter 2

Problem Definitions

In an effort to be consistent in naming conventions and parameter definitions throughout this work, the problem space will be defined. The conventions described in Section 2 will be used throughout the rest of the work.

2.1 The Field

The initially unknown field, referred to as the *target field*, will be a rectangular field of height h , and width w , i.e. $Z \in \mathbb{R}^{h \times w}$. The field is made up of square pixel cells, referred to as *vesicles*, of unit area $a \in \mathbb{R}$. The target field is *gridded* in order to discretize the target field as a whole. Each vesicle can be “visited”, or sampled, in order to yield a number in the set of real numbers. Throughout the work presented, a will be constant throughout the field, and the total area of the target field will be $A = ahw$. Throughout this work, a square target field (i.e. $h = w$) will be used.

2.2 The Sensor

For the sake of a simpler introduction to the methods described in this chapter, the basis of the predictions will be observations of interest made using ideal sensors with no measurement noise. The sensors will measure a subset of the area of the entire target field. This area will be referred to as the *sensor footprint*, and will be equal to the size of a single vesicle of the target field, a .

For the methods developed, the locations of the sensor measurements must be known. The locations will be represented as Cartesian coordinates on the field Z . For an arbitrary observation of the field Z , the location of the measurement will be at coordinates $\mathbf{s} \in \mathbb{R}^2$, and the corresponding sensor measurement would be $Z(\mathbf{s})$.

2.2.1 Real World Sensing Examples

If a Global Positioning System (GPS) sensor is used to estimate position, a Haversine Transformation would likely be used to convert Earth longitude and latitude to Cartesian coordinates within the target field. In the case of predicting the current location of a wildfire, for example, an infrared sensor would likely be used to measure the values of interest, thermal output of the field in this case. In the case of terrain mapping, for example, a lidar sensor might be used to sample terrain altitudes of the terrain below, at marked locations using GPS, on a UAV.

Chapter 3

Previous Work

Field scanning techniques, field prediction, and unknown field exploring are active research areas. Several publications in the past decade alone have covered these types of missions, and demonstrated results using UAVs.

Exploration is a subset of the types of missions UAVs have been used for recently. From Section 2 of Nikhil Nigam's *The Multiple Unmanned Air Vehicle Persistent Surveillance Problem: A Review* [10], the various types of missions possible are described. There exist problems of tracking and patrolling which involve following a moving target, or of finding the spread rate and source of an item of interest. The *exploration* mission type is a procedure which runs parallel to the these types of missions.

Exploration missions often do not specify the model of the item of interest being tracked. Knowing the model and kinematics of the item being tracked makes it possible to use an optimal estimation tool such as an Extended Kalman Filter as in Rabinovich et al. *A Methodology For Estimation of Ground Phenomena Propagation* [12] and *Multi-UAV Path*

Coordination Based on Uncertainty Estimation [11] where the velocity and position states of a ground fire are estimated while tracking the points surrounding the periphery of a wildfire.

In the case of tracking a non-dynamic state of interest, or an unknown field model, a statistical model can be first generated, and the hidden states can be predicted from the learned distributions. Often times, exploration missions are not searching for anything in particular, but rather exploring for the sake of discovery. Without a model describing the states of the item of interest being explored, a simple scanning procedure involving random movements or following a predetermined path, like a zig-zag about the field as in [13] are executed, or a zig-zag which incorporates the model dynamics of the vehicle, as in [9].

The Kriging Method has been used in a UAV Contour Tracking problem in Zhang et al. *Oil Spills Boundary Tracking Using Universal Kriging And Model Predictive Control By UAV* [17]. The paper relies on the knowledge of a model of the oil spill, and therefore is not a generic case of an exploration problem.

Using the next expected overall Kriging variance of field, after taking a given path, using the Kriging method has not been directly used. Using Kriging predictions as feedback into a path planner to estimate confidence return for a given trajectory is introduced in this work.

Part II

Spatial Analysis

Chapter 4

Introduction to Spatial Analysis

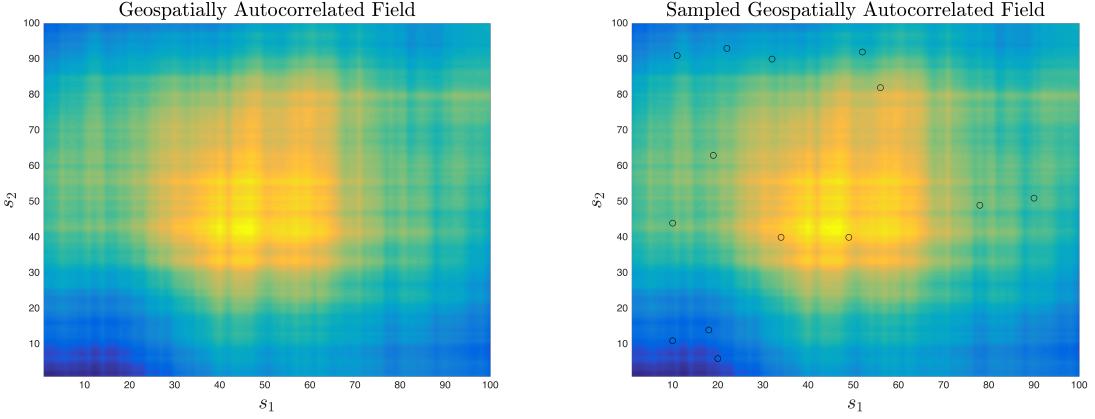
Many of the methods introduced will rely on works developed in the fields of Spatial Statistics, Geostatistics, and Geography. Geo-statisticians have developed much of the work surrounding field predictions in the geospatial domain. In an effort to make the methods introduced be as presentable and contained as possible, a background on the required tools from the field of geostatistics will be reviewed. The Kriging Method, a best linear unbiased predictor (BLUP), and the heart of the method introduced, produces a prediction based on statistical data gathered from samples taken on a field. The method performs a weighted least squares on weights from a covariance matrix created from points sampled on a field. A variance for each prediction computed can also generated as a byproduct of the Kriging prediction, and will be used to calculate information gain in a path finder introduced in this paper. The assumption that the expected value of each point is from a normal distribution, where the variance and expected value of the distribution is a function of geospatial autocorrelation, will be used.

Tobler’s First Law of Geography [15] states, “Everything is related to everything else, but near things are more related than distant things.” Regarding geospatial data, there is a positive correlation between observations with a small difference in distance [8]. This implies the existence of geospatial autocorrelation in many target fields of interest. This implies a positive correlation between elements in the spatial series that are of interest to the introduced technique. Geospatial autocorrelation is the hypothesis that allows naive prediction techniques, like Inverse Distance Weighting (IDW) (Section 4.1), to work.

Furthermore, if the target fields are viewed as an unknown spatial stochastic process, with a generally known underlying Gaussian statistical model (Guassian kernel), more weight in the Inverse Distance Weighting can be placed on the geospatial elements more related to the point that one desires to predict. A best linear unbiased predictor, like The Kriging Method, does just that by first finding the underlying spatial pattern of the field, and then predicting. The methods introduced in this chapter are intended to serve as an introduction to the Kriging Method, and aerial geospatial field interpolation in general.

4.0.1 Autocorrelation in a Field

Positively correlated geospatial autocorrelation in a field implies the existence of a cluster of similar points near one another i.e. relatively small covariances between two spatially similar points. The opposite is true when the overall geospatial autocorrelation of a field is negative. Using Tobler’s First Law of Geography, the assumption that fields measured will contain positive autocorrelation will be used. The degree of geospatial autocorrelation in a field can be measured, and will be discussed in Section 4.2 on Variography.



(a) A randomly generated geospatially autocorrelated field.
(b) Samples at marked locations were taken of the target field.

get field in 4.1a.

Figure 4.1: A Gaussian distributed randomly generated spatially autocorrelated field.

4.1 Inverse Distance Weighting

An inverse distance weighting is a naive interpolation tool where a point is predicted based its distances from a set of observed points. A simple IDW, using Shepard's Method [14], gives a prediction, $\hat{Z}(\mathbf{s}_j)$, of an unobserved point, \mathbf{s}_j , as a function of the $N \in \mathbb{N}$ observed points, $\{Z(\mathbf{s}_1), Z(\mathbf{s}_2), \dots, Z(\mathbf{s}_n)\}$.

$$\hat{Z}(\mathbf{s}_j) = \begin{cases} \frac{\sum_{i=1}^N [w(\mathbf{s}_j, \mathbf{s}_i)] Z(\mathbf{s}_i)}{\sum_{i=1}^N w(\mathbf{s}_j, \mathbf{s}_i)} & \text{if } \forall i \mid d(\mathbf{s}_j, \mathbf{s}_i) \neq 0 \\ Z(\mathbf{s}_j) & \text{if } \exists i \mid d(\mathbf{s}_j, \mathbf{s}_i) = 0 \end{cases} \quad (4.1)$$

$$w(\mathbf{s}_j, \mathbf{s}_i) = \frac{1}{d(\mathbf{s}_j, \mathbf{s}_i)^p} = \|\mathbf{s}_j - \mathbf{s}_i\|_2^{-p} \quad (4.2)$$

where $p \in \mathbb{R}^+$ is the IDW "power parameter". The power parameter, p , controls the emphasis on near and far observations on a prediction. As p increases, the predicted values more closely resemble the closest made observation to the prediction location. Inversely,

as p gets smaller within $(0, 1]$, more emphasis is drawn from observations made further away.

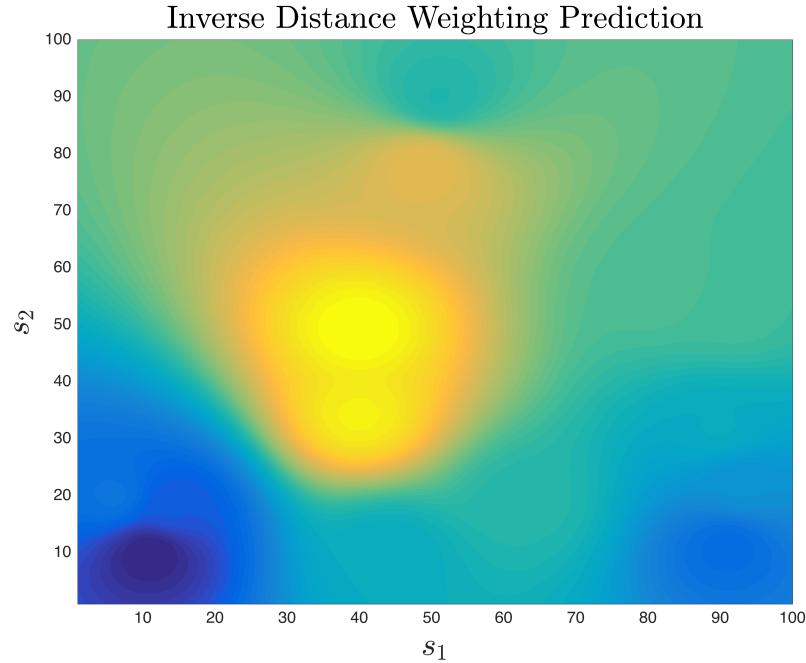


Figure 4.2: An inverse distance weighting predicted field generated from the samples taken of Figure 4.1a at the locations marked in Figure 4.1b.

This method can yield a prediction for all possible s_j points in a field where a set of observations at known locations are made, as done in Figure 4.2. Unfortunately, the method is limited in that it does not take advantage of the underlying stochastic model, and spatial pattern, of the field, Z , being observed to make a more methodical weighted sum prediction. A more statistical approach will be introduced in Section 4.2 on Variography to *learn* the underlying statistical pattern and autocorrelation of a field.

4.2 Variography

Variography is a set of procedures for examining and interpreting spatial dependence and geospatial autocorrelation in a field of observed data. In order to make a more intelligent weighted sum for a prediction, extracting the underlying geospatial autocorrelation function, the *variogram*, of a field will be introduced. The variogram function will be factored into a classical prediction via weighting, yielding a Kriging Weighting.

4.2.1 The Variogram

A variogram quantifies dependence for two disjoint observations separated by some distance, or *lag*, away. The function, in essence, yields a value directly proportional to the covariance between two given points in a stochastic field.

A Variogram is intended to be a continuous function which yields a covariance between two points $Z(\mathbf{s}_i)$, $Z(\mathbf{s}_j)$, which have not necessarily been observed, but known to be a Euclidean distance, or lag, $h_{i,j} \in \mathbb{R}$ apart, where

$$h_{i,j} = \|\mathbf{s}_i - \mathbf{s}_j\|_2 \quad (4.3)$$

Using the assumption on what a point's value on a field is constructed of is made in Equation 2.4.1 of Matheron, 1963 [7]:

$$Z(\mathbf{s}_i) = \mu(\mathbf{s}_i) + \theta(\mathbf{s}_i) \quad (4.4)$$

Where $\theta(\cdot)$ is a zero-mean intrinsically stationary stochastic Wiener process. An assumption that the mean $\mu(\cdot) = \bar{Z}$ is only constant in a reasonably small neighborhood of Z . This becomes relevant when the sample sizes of the field increase where more reliable means will be derived from local neighborhoods. The size of the local neighborhoods to

consider a constant field mean within will later be defined to be a function of the maximum autocorrelation lag, but no cut-off is required by definition.

4.2.2 The Semivariogram

The Semivariogram is defined to be the average squared difference between two points separated by some distance apart. Matheron, 1963 formally defines a variogram in [7] in three-dimensional space. Using the notation used in this paper for a two-dimensional field, the Semivariogram will be defined as:

$$\gamma(h) = \frac{1}{2A} \iint_A [Z(\mathbf{s} + h) - Z(\mathbf{s})]^2 dA \quad (4.5)$$

Where A is a closed area in a field, Z , to consider, $Z(\mathbf{s})$ is the value of a point at location \mathbf{s} on the field Z , and $Z(\mathbf{s} + h)$ is the value of some point a distance h , defined in Equation 4.3, apart from a point \mathbf{s} on the field Z .

It is infeasible to estimate an observation value at each possible point in the field to compute a continuous Semivariogram. Furthermore, the fields observed using these methods are typically gridded, and therefore not continuous by their analytical nature. A discrete model must first be constructed, and will then be fit into a continuous variogram model. This is done by first constructing a discrete variogram model, or *Empirical Semivariogram*, and then fitting a continuous model to it. Fitting a discrete Semivariogram should in turn yield a function close to $\gamma(h)$ defined in Equation 4.5, and should be identical given that every point in the area A is sampled with infinite precision.

4.2.3 The Empirical Semivariogram

An Empirical Semivariogram, or Experimental Variogram, is a discrete function representing the covariance of the observation value difference between two sampled locations that are some distance h apart. By modifying equation 2.4.2, Matheron, 1963 [7] to include an additional boundary to classify a “bin”, for two observations \mathbf{s}_i and \mathbf{s}_j in a stochastic field, Z , the experimental semi-variogram is defined to be:

$$2\hat{\gamma}(h) := \frac{1}{|N(h, \delta)|} \sum_{\forall \mathbf{s}_i, \mathbf{s}_j \in N(h, \delta)} |Z(\mathbf{s}_i) - Z(\mathbf{s}_j)|^2 \quad (4.6)$$

Where $N(h, \delta)$ is the set of all pairs of observed points that are a distance in the interval $[h - \delta, h + \delta]$ away. This distance is referred to as the *lag* between two points. If $\delta = 0$, the semi-variogram is not said to be *binned*.

The experimental variogram conveys the geospatial autocorrelation of a sampled field. As the lag between two given points increases, the covariance also increases. The covariance levels out to a steady value (the *sill*) at some distance in the domain (the *range*). This position in the function marks where the loss of reliable geospatial autocorrelation between two points that are a distance h apart lays.

4.2.4 Converting a Semivariogram to a Variogram

The intent of fitting a statistical model to an experimental variogram is to approximate the continuous covariance for any two points, that have not necessarily been observed, on Z that are at some known lag apart.

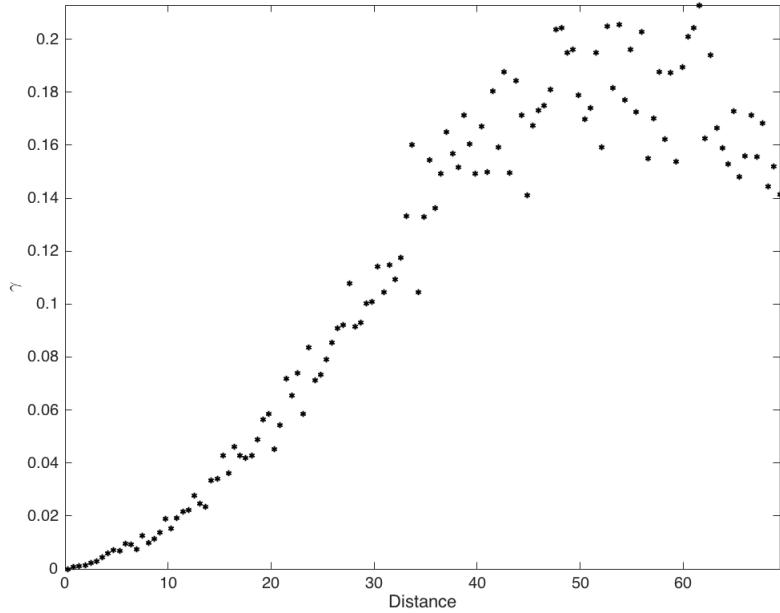


Figure 4.3: An empirical semivariogram.

4.2.5 Variogram Models

The Empirical Semivariogram will be fit to a statistical model, or *kernel*, known as a Variogram Model. There exist some well known models, further discussed in this section. Each model is a scalar function of lag, h , sill, s , and range, a . The term *sill* refers to the point on the codomain where two points at the lag specified are no longer autocorrelated. The sill is therefore the largest value of covariance for two disjoint points on a field that are still considered to be autocorrelated. The corresponding point on the domain for the sill is referred to as the *range* on the variogram. Two points that have a lag larger than the range are not considered to be autocorrelated.

The *nugget* of the variogram is defined to be the variance at zero separation, or $\gamma(0)$. This

value is exactly zero for ideal measurements, but is generally not for real-life measurements.

The value found for the nugget is typically summed with the value yielded by γ , to get the final covariance for a given lag.

The Gaussian Model

$$\gamma_g(h, s, a) = s \left[1 - \exp \left(-\frac{h^2}{a^2} \right) \right] \quad (4.7)$$

The Gaussian model will asymptotically reach its sill. The sill would be at the limit as h approaches infinity. The *practical range* is therefore used to refer the point on the domain where the variogram reaches 95% of its sill.

The Exponential Model

$$\gamma_e(h, s, a) = s \left[1 - \exp \left(-\frac{h}{a} \right) \right] \quad (4.8)$$

The same rules as the Gaussian model apply to the Exponential model.

The Spherical Model

$$\gamma_s(h, s, a) = \frac{s}{2} \left[\frac{3h}{a} - \left(\frac{h}{a} \right)^3 \right] \quad (4.9)$$

The spherical model will reach an exactly zero slope at the sill and range.

4.2.6 Fitting A Semi-Variogram

The kernel function of the range, a , the sill, s , and lag, h is chosen based on the statistical properties of the field being examined. Although there exist no closed form solution for finding an appropriate variogram model for a given field, one can compare a variety of

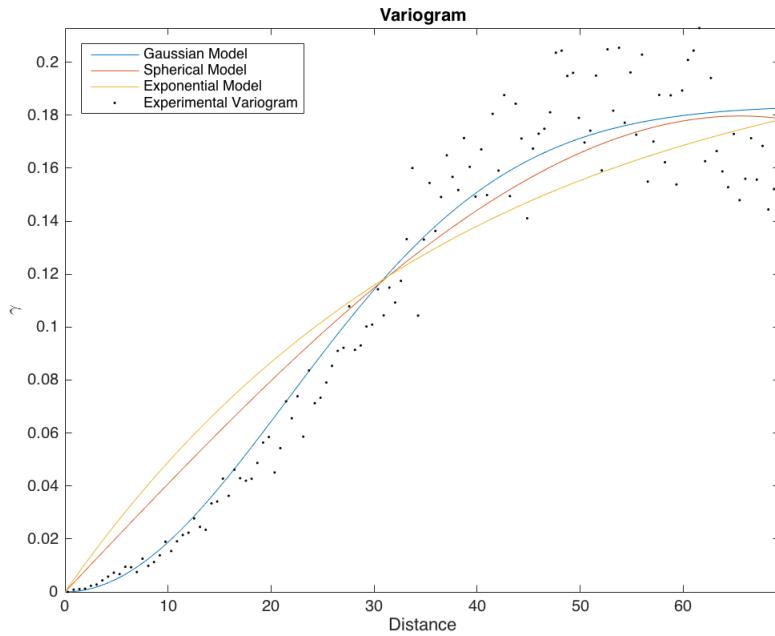


Figure 4.4: Examples of three different variogram models.

different models against one another. Conducting cross-validation tests and comparing root-mean squared prediction errors for different models are common approaches for finding appropriate variogram models.

Fitting a Variogram Using MATLAB

Using a version of the *fminsearch* function in *MATLAB*, a variogram can be fit to the desired objective function from a set of samples, and initial guesses for the range and sill. As the function is used over several iterations of sampling, the fit range and sill values found in the previous iteration can be used as the seed to the next iteration of the fit in an attempt to minimize computation time. The function is defined to “find the minimum of an unconstrained multi-variable function using a derivative-free method”, expressed in

Equation 4.10.

$$\gamma(h) = \min [\gamma_{kernel}(h, s, a) - \hat{\gamma}(h)]^2 \quad (4.10)$$

The function is then modified by specifying bounds of minimization in an attempt to decrease iterations of the function fit, which can be computationally expensive as more samples are taken. This modified version of *fminsearch*, named *fminsearchcon*, can be downloaded from the MathWorks File Exchange.

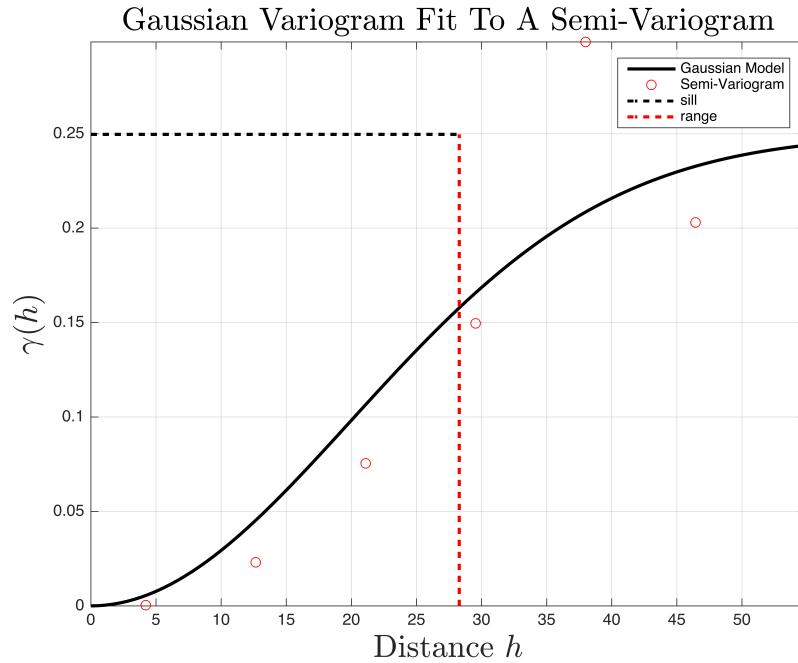


Figure 4.5: An experimental variogram generated using Equation 4.6 from the samples taken in Figure 4.1b. δ was chosen such that for n observations, a total number of $\left\lfloor \frac{n}{2} \right\rfloor$ points were plotted. A Gaussian statistical model was fit to the experimental variogram. The variogram was fit using *fminsearchcon* in *MATLAB*.

Chapter 5

The Kriging Method

The Kriging Method conducts a weighted sum using the continuous variogram model that was fit to the physical observations made. The method can yield a prediction for each vesicle in a target space similar to the Inverse Distance Weighting method described in Section 4.1, but with more statistical robustness.

5.1 Forms of the Kriging Method

There exist three major forms of the Kriging Method. All of which differ primarily in the handling of the mean gathered from observations of a target field. The *Simple Kriging Method* makes the assumption that the mean is known and constant throughout the entirety of an observed field. This is of course not the case for fields that are very large as it does not follow Tobler's First Law. The *Ordinary Kriging Method* can deduce the local mean of a neighborhood from a smaller subset of observations in a larger target field. This is done by classifying the larger field into smaller neighborhoods where the mean is only constant

within those neighborhoods. Ordinary Kriging has the advantage that the mean is not required to be known before running a prediction. The *Universal Kriging Method* can perform similar local mean calculations as the Ordinary Kriging Method, but does so by fitting a polynomial representing a mean trend model and not from a constant mean value representing that neighborhood [16] as seen in Section 4.2.6 on fitting a variogram.

5.2 Covariance Matrix From A Variogram

From the fit variogram which represents our lag covariances, a *covariance matrix* for N observations, $P \in \mathbb{R}^{N \times N}$, will be constructed. The value of the element $P_{i,j}$, will represent the covariance of the lag between the i^{th} and j^{th} observations. If $i = j$, the value of the element, $P_{i,j}$ would be the variance of that observation.

$$P_{i,j} = \text{cov}\{Z(\mathbf{s}_i), Z(\mathbf{s}_j)\} = \gamma(\|\mathbf{s}_i - \mathbf{s}_j\|_2) \quad (5.1)$$

$$P = \begin{bmatrix} \text{var}\{\mathbf{s}_1\} & \text{cov}\{\mathbf{s}_1, \mathbf{s}_2\} & \dots & \text{cov}\{\mathbf{s}_1, \mathbf{s}_N\} \\ \text{cov}\{\mathbf{s}_2, \mathbf{s}_1\} & \text{var}\{\mathbf{s}_2\} & \dots & \text{cov}\{\mathbf{s}_2, \mathbf{s}_N\} \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}\{\mathbf{s}_N, \mathbf{s}_1\} & \text{cov}\{\mathbf{s}_N, \mathbf{s}_2\} & \dots & \text{var}\{\mathbf{s}_N\} \end{bmatrix} \quad (5.2)$$

5.3 The Proximity Vector

For any given point on a field, we can construct a *proximity vector*, $\mathbf{d}_0 \in \mathbb{R}^N$, which contains the covariance of a given point, \mathbf{s}_0 on the field with the N observations made. The k^{th} element of \mathbf{d}_N , would therefore contain the covariance for the lag between point \mathbf{s}_0 and the k^{th} observation made, \mathbf{s}_k .

$$\mathbf{d}_0(k) = \text{cov}\{Z(\mathbf{s}_0), Z(\mathbf{s}_k)\} = \gamma(\|\mathbf{s}_0 - \mathbf{s}_k\|_2)$$

$$\mathbf{d}_0 = \begin{bmatrix} \text{cov}\{Z(\mathbf{s}_0), Z(\mathbf{s}_1)\} \\ \text{cov}\{Z(\mathbf{s}_0), Z(\mathbf{s}_2)\} \\ \vdots \\ \text{cov}\{Z(\mathbf{s}_0), Z(\mathbf{s}_N)\} \end{bmatrix} = \begin{bmatrix} \gamma(\|\mathbf{s}_0 - \mathbf{s}_1\|_2) \\ \gamma(\|\mathbf{s}_0 - \mathbf{s}_2\|_2) \\ \vdots \\ \gamma(\|\mathbf{s}_0 - \mathbf{s}_N\|_2) \end{bmatrix} \quad (5.3)$$

Furthermore, The Kriging Method can be *bounded*. If a to-be-predicted point and a given sample is beyond the range value fit to the variogram model, the corresponding element in the proximity vector is set to the sill. This ensures that points outside of the range of autocorrelation are not weighted anymore than they should be. This method is suggested when the variogram model used is a *bounded function*, e.g. the Spherical Model (Equation 4.9).

5.4 The Kriging Weights

Similarly to the Inverse Distance Weighting method, a set of weights will be computed for each vesicle in the target field. These weights will be referred to as the *Kriging Weights*, or the *error variance vector*. For a given prediction location, \mathbf{s}_0 , the Kriging Weight vector, $\boldsymbol{\lambda}_0$, will be defined as the product of the inverse of the covariance matrix of the field and the proximity vector of the point to predict.

$$\boldsymbol{\lambda}_0 = P^{-1} \mathbf{d}_0 \quad (5.4)$$

5.5 The Kriging Prediction Equation

The Kriging equation will be used to predict the value, $\hat{Z}(\mathbf{s}_0)$ of an unobserved location, \mathbf{s}_0 . The prediction is a function of the Kriging Weights and a vector of N observations.

$$\hat{Z}(\mathbf{s}_0) = [Z(\mathbf{s}_1) \ Z(\mathbf{s}_2) \ \dots \ Z(\mathbf{s}_N)] \boldsymbol{\lambda}_0 \quad (5.5)$$

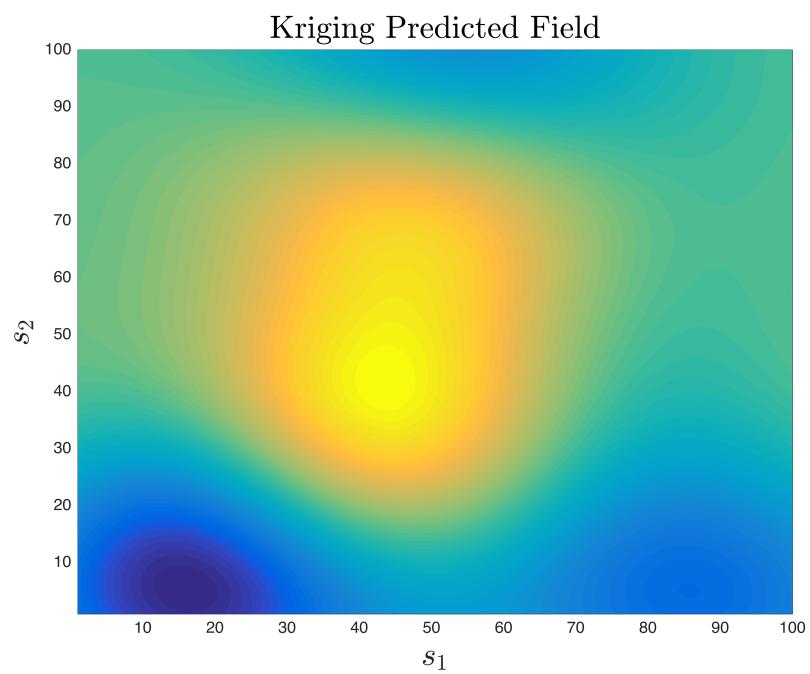


Figure 5.1: A Kriging Method predicted field generated from the samples taken of Figure 4.1a at the locations marked in Figure 4.1b.

5.6 Variance of A Kriging Prediction

The variance of a point predicted on a target field can be calculated using byproduct terms generated along the way of calculating a Kriging prediction. For a predicted point $\hat{Z}(\mathbf{s}_0)$, using the proximity vector, \mathbf{d}_0 , defined in Section 5.3, and the Kriging Weights, $\boldsymbol{\lambda}_0$ defined in Section 5.4 for the predicted point, the variance of the prediction for that point is defined as:

$$\text{var}\{\hat{Z}(\mathbf{s}_0)\} = \mathbf{d}_0 \cdot \boldsymbol{\lambda}_0 = \mathbf{d}_0 \boldsymbol{\lambda}_0^T \quad (5.6)$$

5.7 The Kriging Error

From the method described in Equation 5.5, an error can be computed by comparing the predicted value of a previously observed point, $\hat{Z}(\mathbf{s}_p)$, to the observed value of the point, $Z(\mathbf{s}_p)$.

$$\tilde{Z}(\mathbf{s}_p) = \hat{Z}(\mathbf{s}_p) - Z(\mathbf{s}_p) \quad (5.7)$$

The Kriging Error value will be used to test the unbiasedness of the predictions made, and is factored into confidence of predictions made.

5.8 Procedure For Field Prediction Using The Kriging Method

In order to predict the entirety of a target field from a finite set of N observations and their respective locations, O , the Kriging Prediction is run at every possible unobserved vesicle in the target field, Z . For a single iteration of collecting observations and making predictions, a covariance matrix must first be constructed, and then a the proximity vector

and Kriging Weights are computed for all unobserved vesicles. The Kriging prediction formula is then used to compute the predicted value of each vesicle to predict.

Algorithm 1: Kriging Prediction of Target Field

```

1: procedure KRIGINGPREDICTFIELD( $Z, O$ )
2: Generate Semi-Variogram:
3:    $\forall \mathbf{s}_i, Z(\mathbf{s}_i) \in O$ :
4:      $\hat{\gamma}(h) \leftarrow \mathbf{s}_i, Z(\mathbf{s}_i)$ 
5:
6: Generate Variogram:
7:    $\gamma(h)$  fits to  $\hat{\gamma}(h)$ 
8:
9: Construct Covariance Matrix:
10:   $\forall (\mathbf{s}_i, \mathbf{s}_j) \in O :$ 
11:     $h_{i,j} = \|\mathbf{s}_i - \mathbf{s}_j\|_2$ 
12:     $P_{i,j} = \gamma(h_{i,j})$ 
13:
14: Run Kriging Predictions For Target Field:
15:   $\forall \mathbf{p}_i \in Z$ :
16:     $\mathbf{d}_i = [\gamma(\|\mathbf{s}_1 - \vec{p}_i\|_2) \dots \gamma(\|\mathbf{s}_N - \vec{p}_i\|_2)]^T$ 
17:     $\lambda_i = P^{-1}\mathbf{d}_i$ 
18:     $\hat{Z}(\mathbf{p}_i) = [Z(\mathbf{s}_1) \dots Z(\mathbf{s}_N)] \lambda_i$ 
19:     $\text{var}\{\hat{Z}(\mathbf{p}_i)\} = \mathbf{d}_i \cdot \boldsymbol{\lambda}_i$ 

```

When Algorithm 1 is run on the target field from Figure 4.1a, for the samples taken in Figure 4.1b, a prediction of the entire field can be generated, as seen in Figure 5.1.

Part III

Autonomous Field Exploration Using The Kriging Method

Chapter 6

Vehicle & Information Gain Model

In an effort to formalize the dynamics of the exploration vehicle, and the information gain on the field, the models for vehicle dynamics and field variances will be introduced.

6.1 Exploration Vehicle Model Dynamics

The state vector of the vehicle will be defined as follows:

$$\mathbf{X} = \begin{bmatrix} x \\ y \\ \theta \\ \omega \\ V \end{bmatrix} \quad (6.1)$$

Where x and y are the vehicle's position on a field, θ is the vehicle's heading angle, ω is the vehicle's angular velocity, $\dot{\theta}$, and V is the magnitude of the linear velocity of the vehicle. Both ω and V are control inputs to the vehicle.

The exploration vehicle dynamics will be modeled after a simple forward discrete kinematics model with a constant time step per iteration, ΔT . An iteration of the propagation model will be the sum of the previous iteration, \mathbf{X}_k , the nonlinear vehicle dynamics, $\mathbf{f}(\mathbf{X}_k)$,

and the control input, \mathbf{u}_k .

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \mathbf{f}(\mathbf{X}_{k+1}) + \mathbf{u}_k \quad (6.2)$$

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \begin{bmatrix} V_k \Delta T \cos \theta_k \\ V_k \Delta T \sin \theta_k \\ \omega \Delta T \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \omega_k \\ V_k \end{bmatrix} \quad (6.3)$$

The speed, V , is assumed to be regulated at a constant value of $V = v \frac{\text{m}}{\text{s}}$ for all values of k .

6.2 Field Uncertainty Model

In Section 5.6, a method for calculating the variance of a prediction was defined as a function of the proximity vector and Kriging weights generated for the prediction point.

$$\text{var}\{\hat{Z}(\mathbf{s}_0)\} = \mathbf{d}_0 \cdot \boldsymbol{\lambda}_0 = \mathbf{d}_0 \boldsymbol{\lambda}_0^T \quad (6.4)$$

The variance defined represents the square of the standard deviation of the distribution the expected value of the prediction is sampled from. For points that have been directly measured, the variance is zero, assuming the field has a high level of ergodicity. The uncertainty of the prediction of a point in the target field is therefore directly proportional to the variance of its prediction. The goal of a path planner intending to suppress uncertainty of predictions in a target field would then be to reduce the overall variance of the target field being explored.

A criteria for overall field uncertainty can be defined as the average variance, calculated from a prediction of a target field from a set S , $\hat{Z}(S)$, for all N predictable points on a target field, Z .

$$\Sigma_{\text{var}}(\hat{Z}(S)) = \text{avg}\{\text{var}\{\hat{Z}(S)\}\} = \frac{1}{N} \sum_{\forall i \in Z} \mathbf{d}_i \boldsymbol{\lambda}_i^T \quad (6.5)$$

Where \mathbf{d}_i is the proximity vector of the i^{th} point on the field Z (Section 5.3), and $\boldsymbol{\lambda}_i$ are the Kriging weights of the i^{th} point in Z (Section 5.4).

Chapter 7

Path Planning

The goal of the planner introduced is to assist in the discovery of a field's features with a tunable level of speed and confidence of prediction. The user of such a system could choose to scan more area if fuel is not of high concern. Likewise, if the field is very large, or several fields need to be scanned in a limited amount of time, a quicker scan with a lower degree of prediction certainty can be performed.

7.1 Uncertainty Loss Function

In Section 6.2, a criteria for overall field uncertainty was introduced. Given a set of previously sampled points, S , and a new set of samples, T , a new field uncertainty can be calculated via $\Sigma_{\text{var}}(\hat{Z}(S + T))$, where $S + T$ is the union of the two sets S and T . Given a new set of samples, T , the estimated loss in a target field's prediction uncertainty, $L(T)$, is a metric that should be maximized by a path planner aiming to improve prediction quality for exploration purposes.

$$L(T) = (\Sigma_{\text{var}}(\hat{Z}(S + T)) - \Sigma_{\text{var}}(\hat{Z}(S))) \quad (7.1)$$

The optimal path, O , subject to a limited scanning time constraint, is the path that simultaneously maximizes $L(O)$, and minimizes the length of the path taken (using the assumption of a constant linear velocity from Section 6.1), $l(S+O)$. A method for prediction feedback into a variance calculation, which aims to find an optimal path, is introduced in Section 7.4.2.

$$O = \arg \max_T \beta L(T) - (1 - \beta) l(S + T) \quad (7.2)$$

Where $\beta \in [0, 1]$ is a real number which puts more emphasis on exploration time over prediction quality. It is important to note that as more samples are taken, the overall field prediction variances change. It would be in the benefit of a path planner to batch process a set of points after meeting a predetermined waypoint, or after a threshold number of samples.

Given an endpoint in a single trajectory, in the limit, recalculating O at every sample would optimally shape the trajectory of the exploration vehicle. With no endpoint selected, the exploration vehicle could be found in a repeating state due to being stuck in a global minimum in the variance field. In an effort to avoid the sticking minimum problem, the path planners introduced will be endpoint oriented (where the endpoint of any trajectory is predetermined), and the goal of each path taken is to make it to the endpoint. Furthermore, a new path will only be calculated as a batch process after sampling a trajectory. This is done in an attempt to reduce computation time as the Kriging predictions and variance calculations become more expensive as more samples are taken.

7.2 Finding Points of Highest Uncertainty

Points on a target field with high prediction variances are points on a field that can be sampled first, in turn maximizing L in Equation 7.1. After sampling an initial set of points, and then running a Kriging prediction on all points on a target, the variance of prediction of all points can be calculated.

The motivation of the path finders introduced are to minimize the average uncertainty of a target field, by to sampling the points representing the highest prediction variance. A set of points where the highest uncertainties lay are found on the field using a simple search.

Let S_k be a singleton set containing the point of highest variance on the k^{th} iteration of the target field prediction variances, represented as the set $\text{var}\{\hat{Z}_k\}$, where $\text{var}\{\hat{Z}\} : \mathbb{R}^2 \rightarrow \mathbb{R}^+$.

$$S_k = \arg \max_{\mathbf{s}} \text{var}\{\hat{Z}_k(\mathbf{s})\} \quad (7.3)$$

The cardinality of the set S_k can be greater than one if there exist multiple instances of the same value of variance in the target field prediction variances. For the sake of simplicity, only the singleton case will be considered.

Let $\text{var}\{\hat{Z}_{k+1}\}$ be the set of points, not including the point of highest variance found in the k^{th} iteration of the set configuration (Equation 7.3), on a target field prediction.

$$\text{var}\{\hat{Z}_{k+1}(\mathbf{s})\} = \text{var}\{\hat{Z}_k(\mathbf{s})\} - S_k \quad (7.4)$$

Let S_v be the set of the N points of highest uncertainty on the target field prediction variances, $\text{var}\{\hat{Z}\}$.

$$S_v = \bigcup_{k=1}^N S_k = \bigcup_{k=1}^N \text{var}\{\hat{Z}_k(\mathbf{s})\} - \text{var}\{\hat{Z}_{k+1}(\mathbf{s})\} \quad (7.5)$$

7.3 Next Highest Variance (NHV) Trajectory Finding

Sampling the location of the next highest variance (NHV) is the simplest and most naive approach to path planning using the Kriging method. By finding the highest variance in the field using Equation 7.3, and setting $N = 1$ in 7.5, the next endpoint is found. Setting the destination, \mathbf{s}_d , to the found waypoint in S_v , and knowing the current location, \mathbf{s}_c , a set of waypoints that establish the route between the start and finish points can be defined as $T = \{[x_1 \ y_1 \ \theta_1]^T, [x_2 \ y_2 \ \theta_2]^T, \dots, [x_{i_f} \ y_{i_f} \ \theta_{i_f}]^T\}$, $i_f = \left\lceil \frac{\|\mathbf{s}_c - \mathbf{s}_d\|_2}{\alpha} \right\rceil$, where, $\alpha \in \mathbb{R}^+$, is a scaling variable representing the step size between waypoints, x and y are field coordinates, and θ is the heading angle of the vehicle.

$$T_{i+1} = T_i + \begin{bmatrix} \alpha \cos \theta_i \\ \alpha \sin \theta_i \\ \text{atan2}(s_{d_y} - T_{i+1,y}, s_{d_x} - T_{i+1,x}) \end{bmatrix} \quad (7.6)$$

The initial point, T_1 , is defined explicitly.

$$T_1 = \begin{bmatrix} s_{c_x} \\ s_{c_y} \\ \text{arctan2}(s_{d_y} - s_{c_y}, s_{d_x} - s_{c_x}) \end{bmatrix} \quad (7.7)$$

7.3.1 NHV Path Planning Algorithm

The exploration algorithm initializes by directing the vehicle to sweep across the forward diagonal of the target field to collect an initial set of samples. An initial variogram is fit, and a Kriging prediction of the field is then made. The variances of all points on the target field are then calculated. The point with the highest prediction variance is found (Section 7.2). The trajectory, T , is then calculated (Section 7.3), and set as the path to take. Once the vehicle completes the last path chosen, the field values and variances are predicted and calculated again. A new path is chosen in the same fashion. The algorithm terminates

when it can no longer choose a path that can be accomplished with the amount of fuel that remains within the exploration vehicle. Another case of planner termination occurs When the average prediction variance of the field is zero, which ideally occurs when all locations have been sampled. See Figure 7.1 for a short 1% capped version of the algorithm.

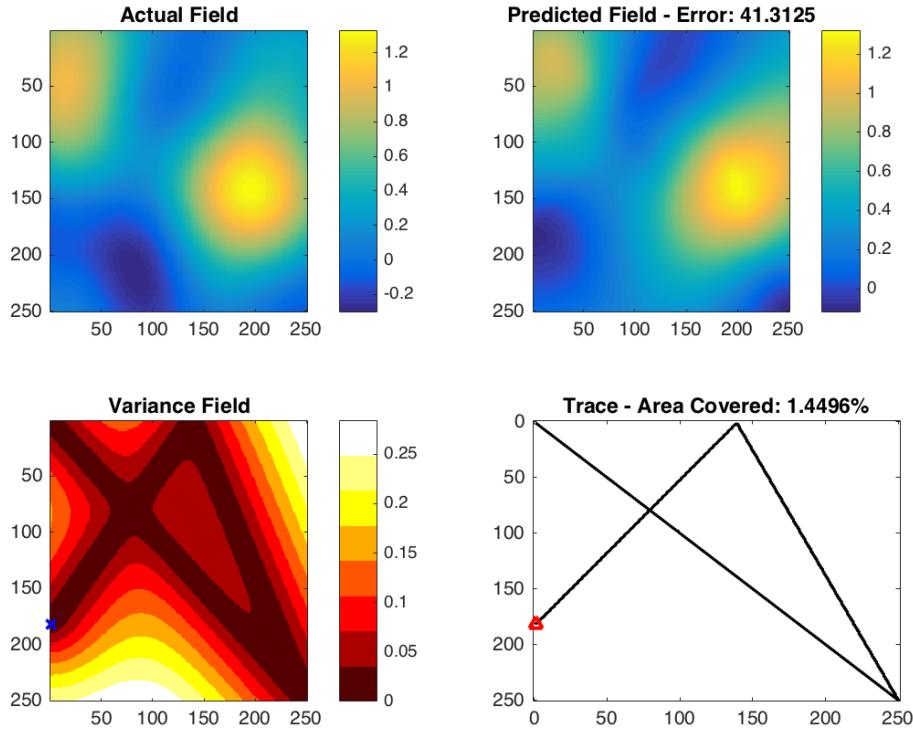


Figure 7.1: The Next Highest Variance (NHV) algorithm (Section 7.3.1 terminated after two iterations of the algorithm. The actual field that is being explored is shown in the upper left. The current prediction of the actual field is shown in the upper right. The variance of the current prediction of the field is shown in the lower left. The trace of the exploration vehicle’s path taken to the point of termination (in red) is shown in the lower right panel. All distance units are in meters.

7.3.2 Inefficiency in NHV

The NHV algorithm does not account for repeating paths, or avoiding re-sampling points. The only knowledge used is a single criteria of variance. Although the ground covered by the algorithm may be sufficient for the most exploration cases, a path planner that considers the cost of trajectories would likely yield better results.

7.4 Monte Carlo Based Trajectory Finding

A path planner designed for autonomous field exploration using a Monte Carlo path selection technique will be introduced. By first selecting a set of locations with the highest uncertainty in current prediction, the path planner will find a suitable path that will in turn minimize the average variance of the target field predictions. The problem can be stated as a sensor placement problem, where a suitable placement of sensors is found via a Monte Carlo simulation of the environment, and a minimization of the Kriging prediction variances [1]. Using Monte Carlo for trajectory finding for obstacle avoidance has been attempted [6], but not for exploration purposes.

In the case of the path planner, because the exploration vehicle is dynamic, the sensors can be thought of as being placed along the trajectory of the vehicle (assuming reasonable ergodicity of the target field). The Monte Carlo approach to path finding via the Kriging Method introduced will first find the points of highest prediction uncertainty on the target field, and then generate a finite set of random trajectories along the originally found trajectories. The trajectory that is predicted to reduce overall field uncertainty will be chosen as the path the vehicle will take.

7.4.1 Finding Monte Carlo Trajectory Sets

After a suitable set of destinations are selected from the set of points containing the N^{th} coordinates of highest uncertainty, S_v , trajectories to each point are calculated. The trajectories calculated can be both deterministic or variably non-deterministic walks from the current location of the exploration vehicle, to the final location.

For a given destination, s_d in S_v , and the current location of the vehicle, s_c , a set of waypoints that establish the route between the start and finish points can be defined as $T_k = \left\{ [x_{k_1} \ y_{k_1} \ \theta_{k_1}]^T, [x_{k_2} \ y_{k_2} \ \theta_{k_2}]^T, \dots [x_{k_{i_f}} \ y_{k_{i_f}} \ \theta_{k_{i_f}}]^T \right\}$, $i_f = \left\lceil \frac{\|s_c - s_d\|_2}{\alpha} \right\rceil$, where, $\alpha \in \mathbb{R}^+$, is a scaling variable representing the step size between waypoints, x and y are field coordinates, and θ is the heading angle of the vehicle.

$$T_{k_{i+1}} = T_{k_i} + \begin{bmatrix} \alpha \cos \theta_{k_i} \\ \alpha \sin \theta_{k_i} \\ \text{atan2}(s_{d_y} - T_{k_{i+1}_y}, s_{d_x} - T_{k_{i+1}_x}) \end{bmatrix} + \begin{bmatrix} \mathbf{w}_{k_{i+1}} \\ 0 \end{bmatrix} \quad (7.8)$$

Where $\mathbf{w}_k \in \mathbb{R}^2$ is a zero-mean Wiener Process with a tunable variance. Multiple Brownian walks from the current position to a given destination can be calculated to increase the pool of possible paths. For a zero variance Wiener process ($\text{var}\{\mathbf{w}\} = 0$), the walk defined is deterministic, and equal to finding a set of N-NHV trajectories (Section 7.4.4). The *cord* of the walk is the line that connects the initial position to the destination position. The cord acts as the expected value in the distribution of trajectories calculated. The initial point, T_{k_1} , and the final point, T_{k_f} , are defined explicitly.

$$T_{k_1} = \begin{bmatrix} s_{c_x} \\ s_{c_y} \\ \text{arctan2}(s_{d_y} - s_{c_y}, s_{d_x} - s_{c_x}) \end{bmatrix}, \quad T_{k_f} = \begin{bmatrix} s_{d_x} \\ s_{d_y} \\ \theta_{f-1} \end{bmatrix} \quad (7.9)$$

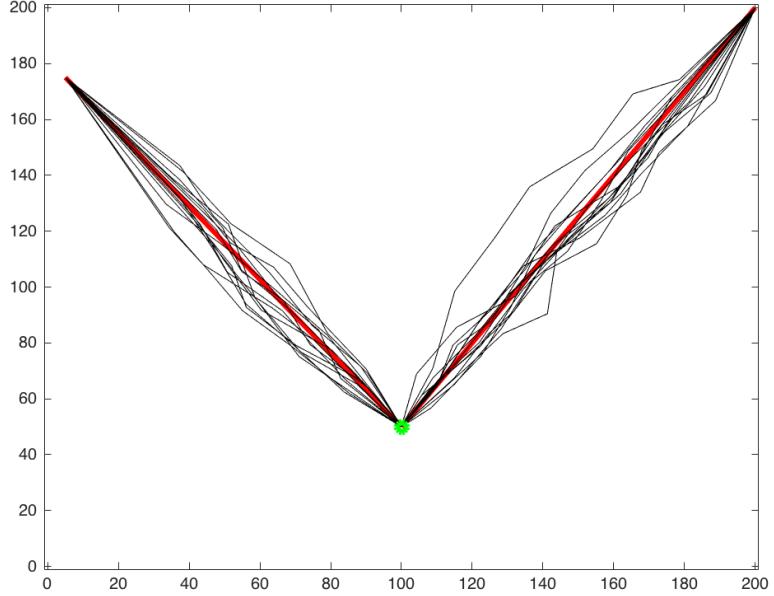


Figure 7.2: Brownian non-deterministic walks (black) surrounding deterministic cords (red).

The starting point is indicated in green. For a set of $N = 2$ possible trajectories, 15 random walks around each trajectory with a variance of 10. All distance units in meters.

7.4.2 Selecting An Optimal Trajectory (Prediction Feedback)

In Section 7.4.1, a set of possible trajectories, T was defined. The trajectory that is ultimately chosen by the introduced path finder is the path that minimizes the average uncertainty of the predicted target field, i.e. maximizes the loss function, $L(T)$. Let $\hat{Z}(S)$ be a Kriging predicted field from a set of samples at known locations. Furthermore, let $\hat{Z}(S + \hat{T}_k)$ be the a Kriging predicted field from the set S concatenated with a virtual set of samples \hat{T}_k . The values in the set \hat{T}_k have not necessarily been sampled, but they contain, as samples, the previously predicted values, $\hat{Z}(T_k)$, at the known locations in T_k .

$$samples(\hat{T}_k) = \hat{Z}(T_k) \quad (7.10)$$

The variance of the field with real and virtual samples, $\text{var}\{\hat{Z}(S + \hat{T}_k)\}$, can then be calculated. Using the definition for $\Sigma_{\text{var}}(\cdot)$ from Section 6.5, the path chosen is the trajectory that satisfies:

$$P = \arg \min_{T_k} \Sigma_{\text{var}}(\text{var}\{\hat{Z}(S + \hat{T}_k)\}) \quad (7.11)$$

7.4.3 Monte Carlo Path Planning Algorithm

The exploration algorithm initializes by directing the vehicle to sweep across the forward diagonal of the target field to collect an initial set of samples. An initial variogram is fit, and a Kriging prediction of the field is then made. The variances of all points on the target field are then calculated. The set, S_v , of points with the N highest prediction variances are found (Section 7.2). A set of trajectories, T , is then calculated (Section 7.4.1). The trajectory that is predicted to minimize the total uncertainty of the target field is chosen as the next path of the exploration vehicle (Section 7.4.2). Once the vehicle completes the last path chosen, the field values and variances are predicted and calculated again. A new path is chosen in the same fashion. The algorithm terminates when it can no longer choose a path that can be accomplished with the amount of fuel that remains within the exploration vehicle. Another case of planner termination occurs when the average prediction variance of the field is zero, which ideally occurs when all locations have been sampled.

An algorithm for Kriging Monte Carlo Path Planning (Kriging MCPP) for a target field, Z , of size $(h \times w)$ is introduced in Algorithm 2. The initial location of the vehicle is set to $s_c = [0 \ 0]^T$. The vehicle starts the path planner with F units of fuel.

In Algorithm 2, the function `SetWaypoint()` is an abstracted function which steers the vehicle in the direction of the waypoint specified, and blocks the code instruction until the waypoint has been met. The function `length(T)` finds the arc length of the path by connecting all points in the trajectory, T . For a deterministically calculated trajectory, T , the arc length is $\alpha|T|$.

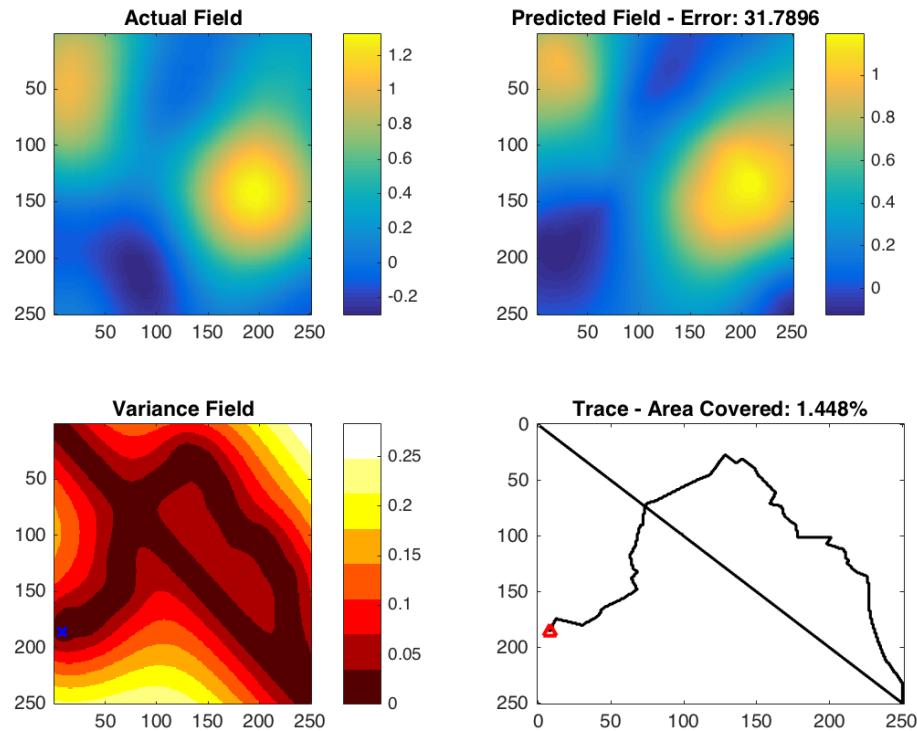


Figure 7.3: The Next Highest Variance (NHV) algorithm (Section 7.3.1 terminated after two iterations of the algorithm. The actual field that is being explored is shown in the upper left. The current prediction of the actual field is shown in the upper right. The variance of the current prediction of the field is shown in the lower left. The trace of the exploration vehicle's path taken to the point of termination (in red) is shown in the lower right panel. All distance units are in meters. Here, N , the number of endpoints selected is 3, and the number of random paths per cord calculated is 15.

7.4.4 N Next Highest Variances Algorithm (Zero Variance MCPP)

A small modification to the NHV algorithm can be made to consider more trajectories. If the set of highest variances on the field, S_v has a cardinality greater than 1 ($N > 1$), N trajectories can be made, and weighed against each other in terms of their return on investment (as discussed in Section 7.4.1). This is done by selecting a zero variance gain on the noise of each Monte Carlo Path, and only calculating a single cord per endpoint. By weighing N Next Highest Variances (N-NHV) against one another, more methodical routes can be taken. It may not always be the case that simply exploring the next highest variance will yield a smaller field variance. If a smaller prediction variance location is selected as the endpoint location (i.e. an endpoint that is not necessarily the first coordinate in the set S_v), the vehicle might sample more unknown points along the way, versus potentially rescanning already sampled area with the simple NHV algorithm.

7.4.5 Benefits in MCPP over NHV

The Monte Carlo based path planner takes into account a set of trajectories, and compares their estimated return on investment. The NHV simply takes the path to the high prediction uncertainty location. Though the MCPP algorithm, with a non-zero noise variance, does not deterministically calculate its trajectories, given enough trajectories, the algorithm could find a path that will reduce overall field uncertainty in a more methodical way over the NHV algorithm.

7.4.6 Disadvantages of The Monte Carlo Path Planner

The MCPP algorithm does not calculate the cost of each next move taken, but rather a set of waypoints taken. In other words, the algorithm only takes into account entire trajectories at a time. A more optimal approach to this planner would be to take into consideration the cost of each waypoint selected. A remedy to this problem using the MCPP would be to increase the number of random trajectories calculated for each cord.

Algorithm 2: Monte Carlo Path Planning (MCPP) with The Kriging Method

```

1: procedure KRIGING_MCPP( $Z$ )
2: Conduct Initial Sweep:
3:   SetWaypoint( $[h \ w]^T$ )
4: Krig The Field:
5:    $\hat{Z}, \text{var}\{\hat{Z}\} = \text{KrigingPredictField}(Z, S)$ 
6: while  $F > 0$  and  $\Sigma_{\text{var}} > 0$ :
7:    $P = []$ 
8:    $\Sigma_{\min} = \infty$ 
9:
10:  Find the highest field variances:
11:  for  $k = 1:N$ 
12:     $S_v(k) = \arg \max_s \text{var}\{\hat{Z}_k(s)\}$ 
13:     $\text{var}\{\hat{Z}_{k+1}(s)\} = \text{var}\{\hat{Z}_k(s)\} - S_v(k)$ 
14:
15:  Calculate trajectories to all points found:
16:   $\forall s_k \in S_v(k)$ :
17:     $s_d = s_k$ 
18:     $f = \left\lceil \frac{\|\mathbf{s}_c - \mathbf{s}_d\|_2}{\alpha} \right\rceil$ 
19:     $T(1) = \begin{bmatrix} s_{c_x} \\ s_{c_y} \\ \arctan2(s_{d_y} - s_{c_y}, s_{d_x} - s_{c_x}) \end{bmatrix}$ 
20:     $\forall i \in (1, f - 1)$ :
21:       $T(i + 1) = T(i) + \begin{bmatrix} \alpha \cos \theta(i) \\ \alpha \sin \theta(i) \\ \text{atan2}(s_{d_y} - T(i)_y, s_{d_x} - T(i)_x) \end{bmatrix} + \begin{bmatrix} \mathbf{w}(i) \\ 0 \end{bmatrix}$ 
22:       $T(f) = \begin{bmatrix} s_{d_x} \\ s_{d_y} \\ \theta_{f-1} \end{bmatrix}$ 
23:
24:  Calculate estimated field confidence for trajectory computed:
25:   $\forall i \in [1, f]$ :
26:    samples( $\hat{S}_T$ ) +=  $\hat{Z}(T(i))$ 
27:    locations( $\hat{S}_T$ ) +=  $T(i)$ 
28:     $\hat{Z}_T, \text{var}\{\hat{Z}\}_T = \text{KrigingPredictField}(Z, \hat{S}_T)$ 
29:     $\Sigma_{\text{var}}(T) = \text{avg}(\text{var}\{\hat{Z}\}_T)$ 
30:
31:    if  $\Sigma_{\text{var}}(T) < \Sigma_{\min}$  and length( $T$ )  $< F$ :
32:       $\Sigma_{\min} = \Sigma_{\text{var}}(T)$ 
33:       $P = T$ 
34:
35:  Navigate through the chosen path:
36:   $\forall p \in P$ :
37:    SetWaypoint( $p$ )

```

Part IV

Simulation & Results

Chapter 8

Simulation Framework

Using *MATLAB*, the methods described in Chapter 7 on Path Planning were implemented to provide a simulation environment to show the effectiveness of the introduced methods. The target fields in the simulations are of a variable size with a single state of interest that can be sampled. The autocorrelation factor of the field in the simulation is an adjustable variable that determines the likeliness of each pair of neighboring points as a function of distance.

The simulation includes a software in the loop vehicle with variable dynamics and incorporates the ability to follow a preplanned route. The simulation will be used to demonstrate the abilities of the algorithms introduced, and comparisons to an algorithm with a preplanned trajectory.

8.1 Simulated Exploration Vehicle Model Dynamics

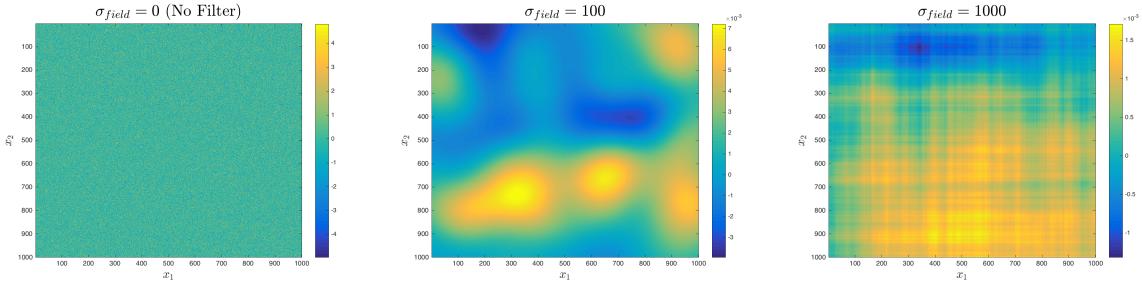
The dynamics of the simulated vehicle are modeled after the vehicle model described in Equation 6.3. In the simulation, ΔT and V are constants predefined in the simulation environment. For a given waypoint destination, the vehicle will pass over every vesicle on the line connecting its original position and its final position. A control angular velocity for each time step, ω_k , is calculated for the trajectory. A trajectory (a set of waypoints), T , calculated in Chapter 7, is loaded into waypoint queue for the vehicle, where each waypoint is me one after another. After meeting the final waypoint in the trajectory, the next set of waypoints is calculated by a path planner. The process continues until the termination condition (maximum area scanned) is satisfied. A sample is taken at every possible vesicle that the vehicle passes over. The location and value of each sample is stored in the vehicle object's memory for later use in the prediction procedures.

8.2 Generating a Target Field

The simulation yields a target field that is of variable height h , and width w . Each vesicle in the field is exactly the area of the sensor footprint of the simulated vehicle's sensor. This is to make the sensor measurements as ideal as possible, so no samples are missed when a vesicle is flown over.

The field is composed of a single feature which is geospatially autocorrelated. Initially, the points on the field are generated from a normal distribution with a standard deviation of 1, and expected value of 0. The field is then convolved with a two dimensional Gaussian filter, of size $h \times w$, with a variable standard deviation, σ_{field} . The final filter “smooths”

the field in order to simulate autocorrelation. The result is a randomly-generated, variably-sized, and autocorrelated field with a unit-less feature of interest. One such field can be observed in Figure 4.1a. As the value of the standard deviation of the Gaussian filter kernel, σ_{field} , increases, the field exhibits higher spatial autocorrelation. Inversely, when σ_{field} is close to zero, the field becomes has no signs of spatial autocorrelation.



(a) $\sigma_{field} = 0$. The field exhibits no spatial autocorrelation.
(b) $\sigma_{field} = \frac{w}{10} = 100$. The field appears to exhibit some degree of spatial autocorrelation.
(c) $\sigma_{field} = w = 1000$. The field exhibits a high degree of spatial autocorrelation.

Figure 8.1: A field is generated using a zero mean normal distribution with variance 1.

Varying degrees of spatial autocorrelation are shown for different values of σ_{field} .

8.3 Simulation Environment

The simulation, once started, runs a single exploration method at a time. A generated field is unknown to a vehicle object, but samples are collected as it passes along the field. The variances of Kriging predictions, the currently predicted field, and the path traversed are plotted along with the actual field being explored.

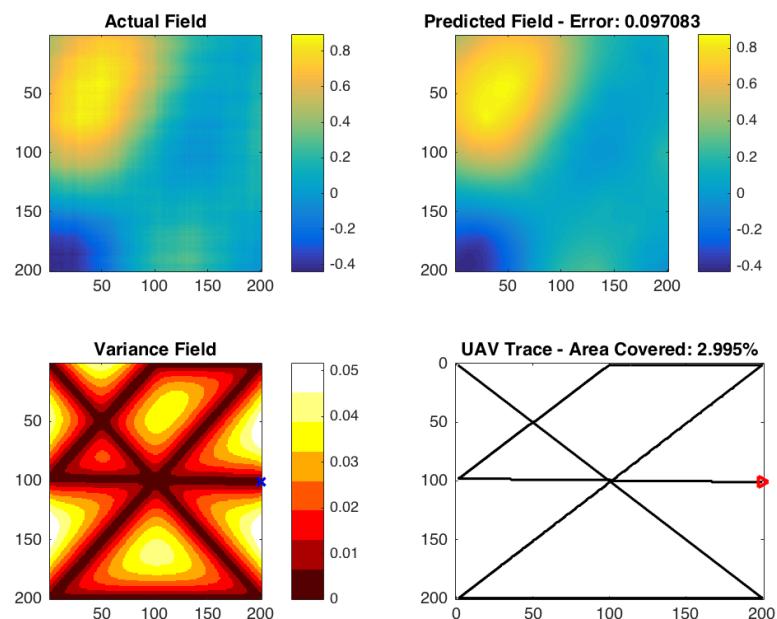


Figure 8.2: Next Highest Variance Path Planning. The actual field (top left), predicted field (with error) (top right), variance field (bottom left), and traversed path (bottom right). The path planner is conducted on the same field with an approximate 3% scan area limit.

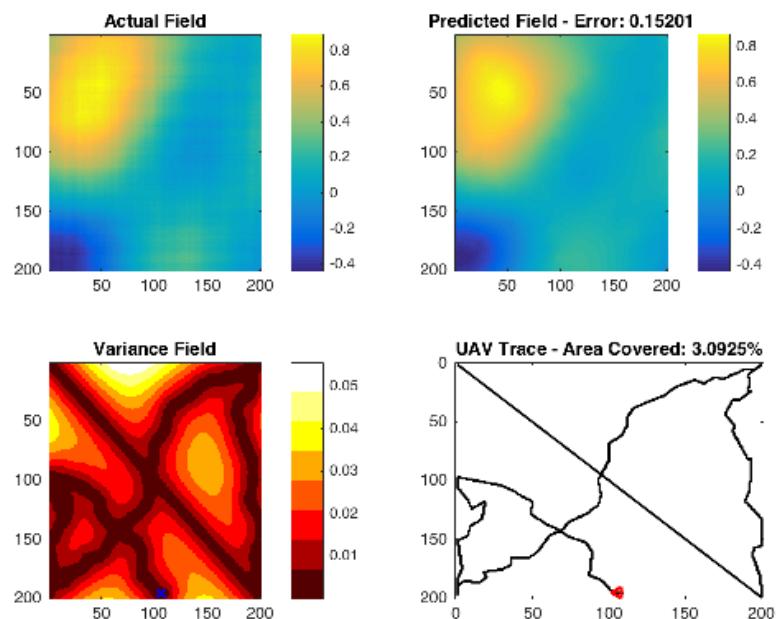


Figure 8.3: Monte Carlo path planner run ($N = 5$, 20 trajectories per path). The actual field (top left), predicted field (with error) (top right), variance field (bottom left), and traversed path (bottom right). The path planner is conducted on the same field with an approximate 3% scan area limit.

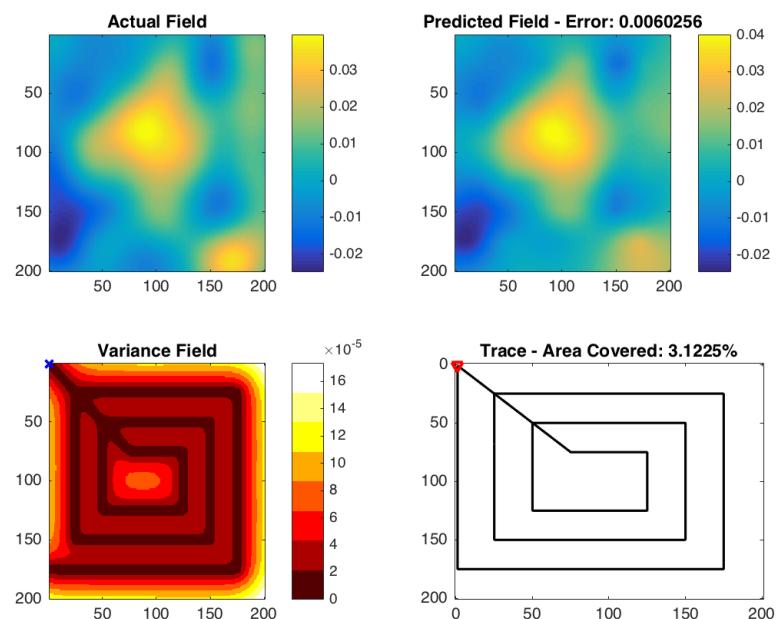


Figure 8.4: Zig-Zag method. The actual field (top left), predicted field (with error) (top right), variance field (bottom left), and traversed path (bottom right). The path planner is conducted on the same field with an approximate 3% scan area limit.

Chapter 9

Results

The three path planners (NHV, N-NHV, and MCPP) introduced in Chapter 7 all aim to reduce the overall prediction uncertainty of a field given a limited amount of flight time. They accomplish the task by calculating variances of a target field's predictions and attempting to choose a trajectory that reduces overall uncertainty.

9.1 Result Comparison Criteria

A common approach to exploration and patrolling problems is the use of a spiral, zig-zag, or lawn mower pattern. The methods introduced will be compared to equally time limited version of a zig-zagging approaches seen in Nikhil Nigam, et al. Control and Design of Multiple Unmanned Air Vehicles for a Persistent Surveillance Task (Part II.C.3, Figure 6, [9]). The method, for the sake of fairer comparison, will run a Kriging prediction and variance calculation on the samples taken using the zig-zag explorer. This is to generate measurable and comparable metrics against the path planners introduced. The path plan-

ners introduced will be compared against the zig-zag exploration method shown in Figure 9.1.

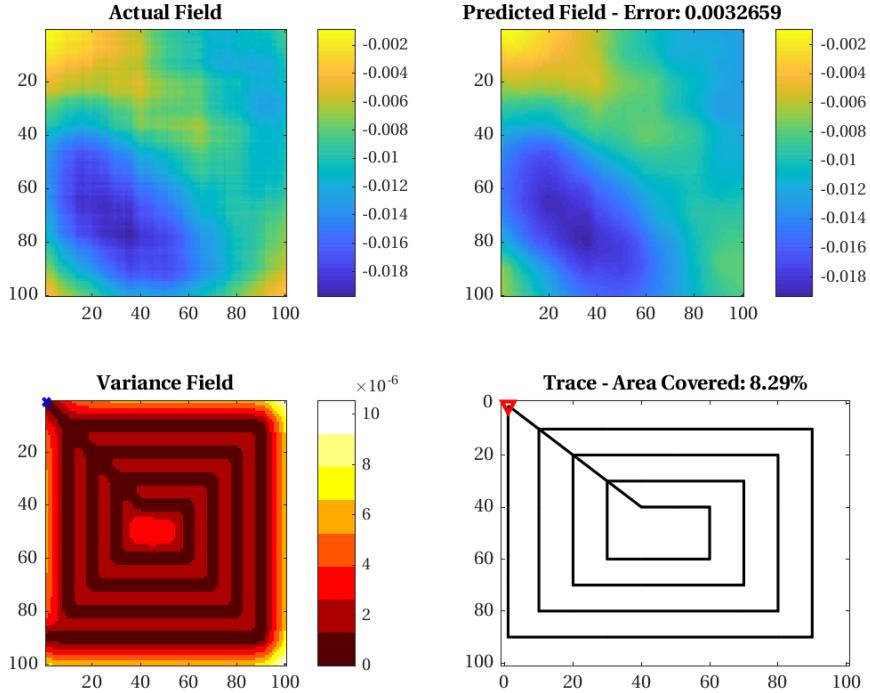


Figure 9.1: Zig-Zag exploration method set to scan a fixed percentage of a target field. A Kriging prediction and variance calculation is computed after completing the maneuver. The actual field that is being explored is shown in the upper left. The current prediction of the actual field is shown in the upper right. The variance of the current prediction of the field is shown in the lower left. The trace of the exploration vehicle's path taken to the point of termination (in red) is shown in the lower right panel. All distance units are in meters.

9.1.1 Variance Drop Calculation

The variance of every target field over each iteration will be the average prediction variance of the field after every prediction recalculation. This criteria is introduced in Section 6.2 on field uncertainty. This metric relates prediction quality of the path planner

to its predicted prediction quality. A drop in variance over iterations should signal a better prediction of the target field over that iteration, and less overall uncertainty of the target field.

9.1.2 Simulation Results

The methods introduced will be compared to one another and the zig-zag method for the same termination condition. Each method will stop the field exploration process when the exploration vehicle traverses a fixed path length expressed in terms of the area percentage scanned, A_{scan} . For example, if the maximum scan percentage of a size $w \times h$ size field is $p\%$, then the method will stop exploring when $A_{scan} = \frac{p}{100}wh$ number of vesicles have been sampled. In an effort to allow the zig-zag method to cover as much of the field as possible, the spacing between each spiral bound, r , will be pre-calculated.

$$r = \frac{100}{p} \quad (9.1)$$

9.1.3 Simulation Result Parameters

The number of trajectories compared in both the NNHV and MCPP methods is $N = 5$. For the MCPP method, an additional 20 Monte Carlo trajectories are calculated for each of the N trajectories. The target field size of the fields compared in the simulation have unit-less vesicle dimensions of 100×100 . Two random number generator seeds (2, 3) are used to generate two sets of runs in an effort to show the methods for a variety of random fields. The autocorrelation factors of the field will be varied in an effort to show the effectiveness of the methods for different field statistics. When the prediction variances of the methods are compared, the values are normalized to an a priori mean variance, which is equal to the

mean variance of the field generated from running a Kriging prediction on the equivalent field from a set of samples taken from the first five forward diagonal vesicles on the target field.

9.1.4 Prediction Error Calculation

The quality of each path planner will be judged by its ability to explore a field in a fixed amount of time. The prediction error of each method will be used as a metric of path planning quality. The actual values of the fields scanned are known in simulation, and for each rerouting iteration, the predictions and prediction errors will be recalculated.

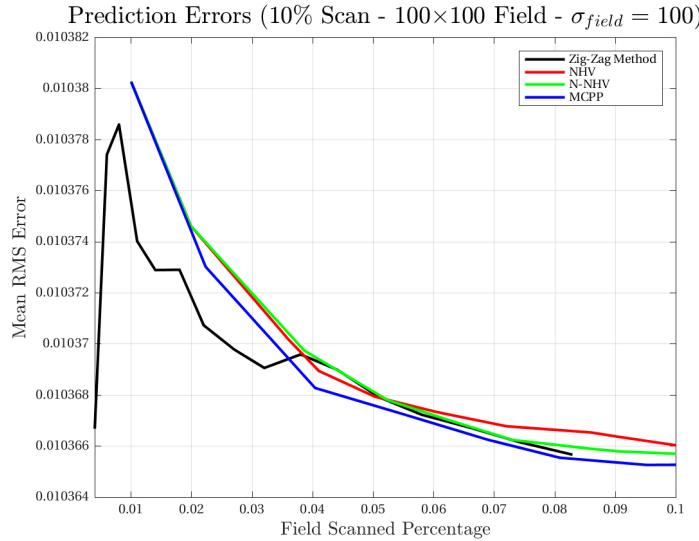
The prediction error function, $\text{erf}(Z, \hat{Z})$, will be the average root mean square (RMS) value for all N field predictions made, point by point, on the actual field, Z , and the predicted field, \hat{Z} .

$$\text{erf}(Z, \hat{Z}) = \frac{1}{N} \sum_{\forall i \in Z} (Z(s_i) - \hat{Z}(s_i))^2 \quad (9.2)$$

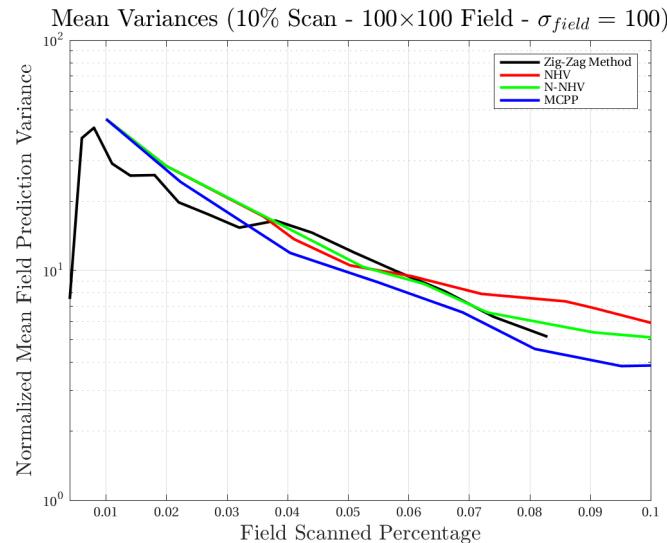
9.2 High Spatial Autocorrelation ($\sigma_{field} = w$) Results

The methods will be compared on target fields generated with the high degree of geospatial autocorrelation.

9.2.1 10% Maximum Field Scan



(a) Prediction errors ($\text{erf}(Z, \hat{Z})$).



(b) Semi-logarithmic prediction variances normalized to an a priori mean variance for the field.

Figure 9.2: A 10% maximum area scan on a field of size 100×100 , $\sigma_{field} = 100$, random seed: 2.

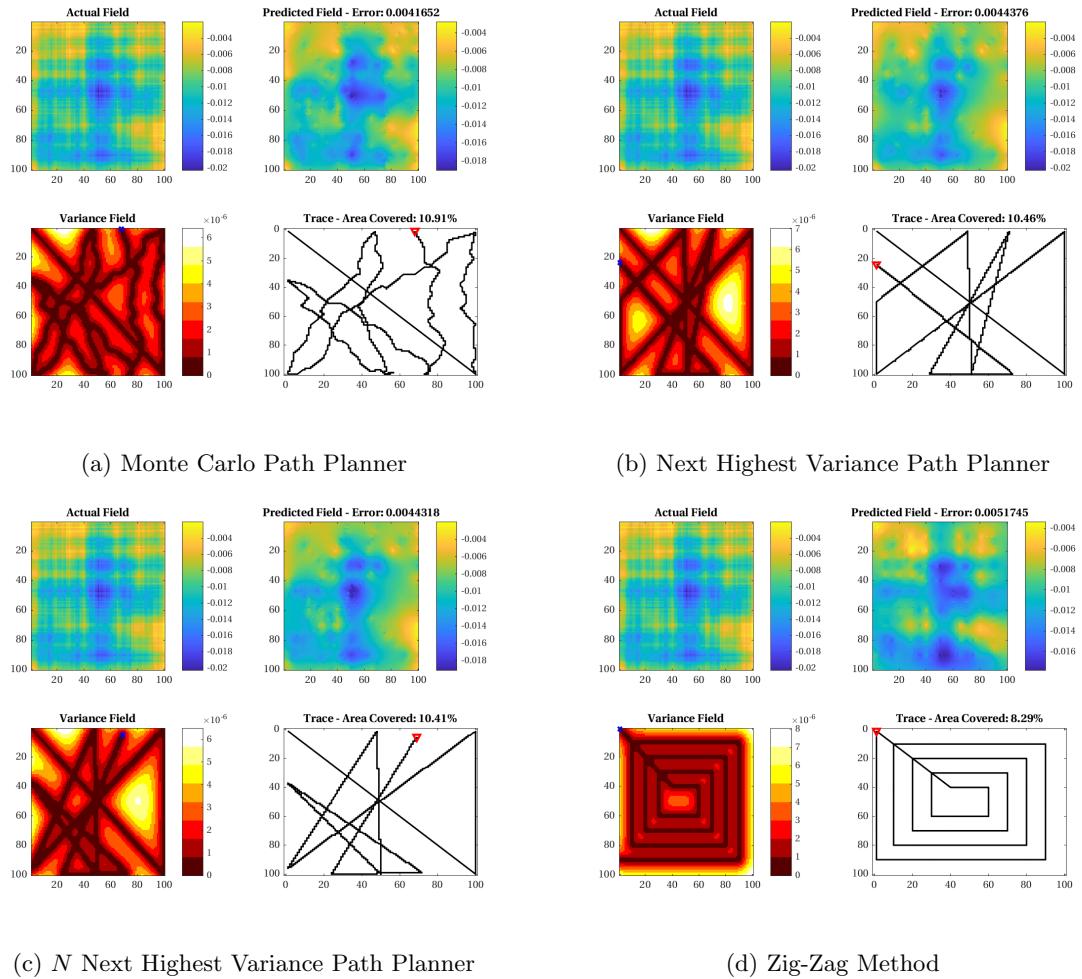
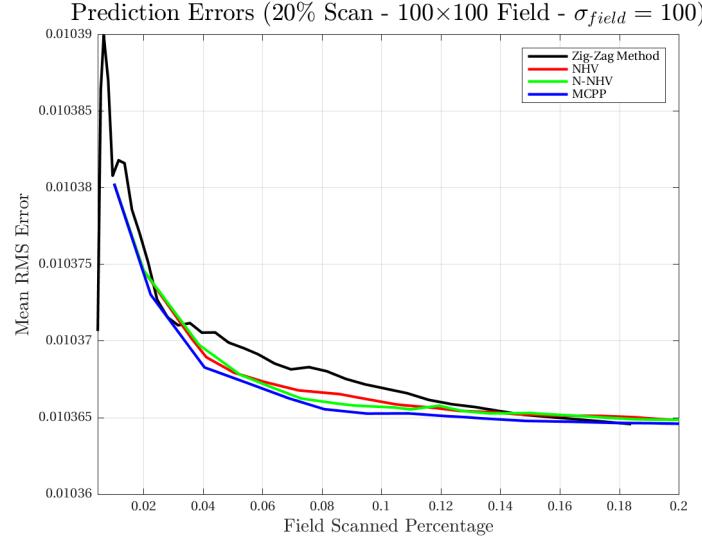


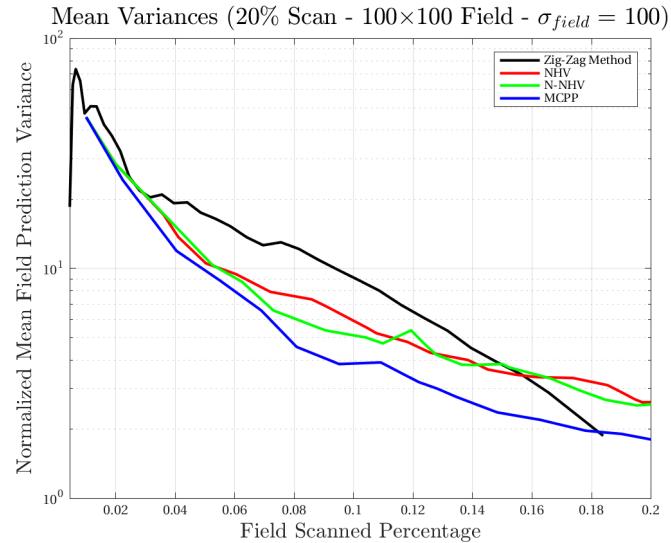
Figure 9.3: Simulation output for a 10% maximum area scan on a field of size 100×100 ,

$\sigma_{field} = 100$, random seed: 2.

9.2.2 20% Maximum Field Scan



(a) Prediction errors ($\text{erf}(Z, \hat{Z})$).



(b) Semi-logarithmic prediction variances normalized to an a priori mean variance for the field.

Figure 9.4: A 20% maximum area scan on a field of size 100×100 , $\sigma_{\text{field}} = 100$, random seed: 2.

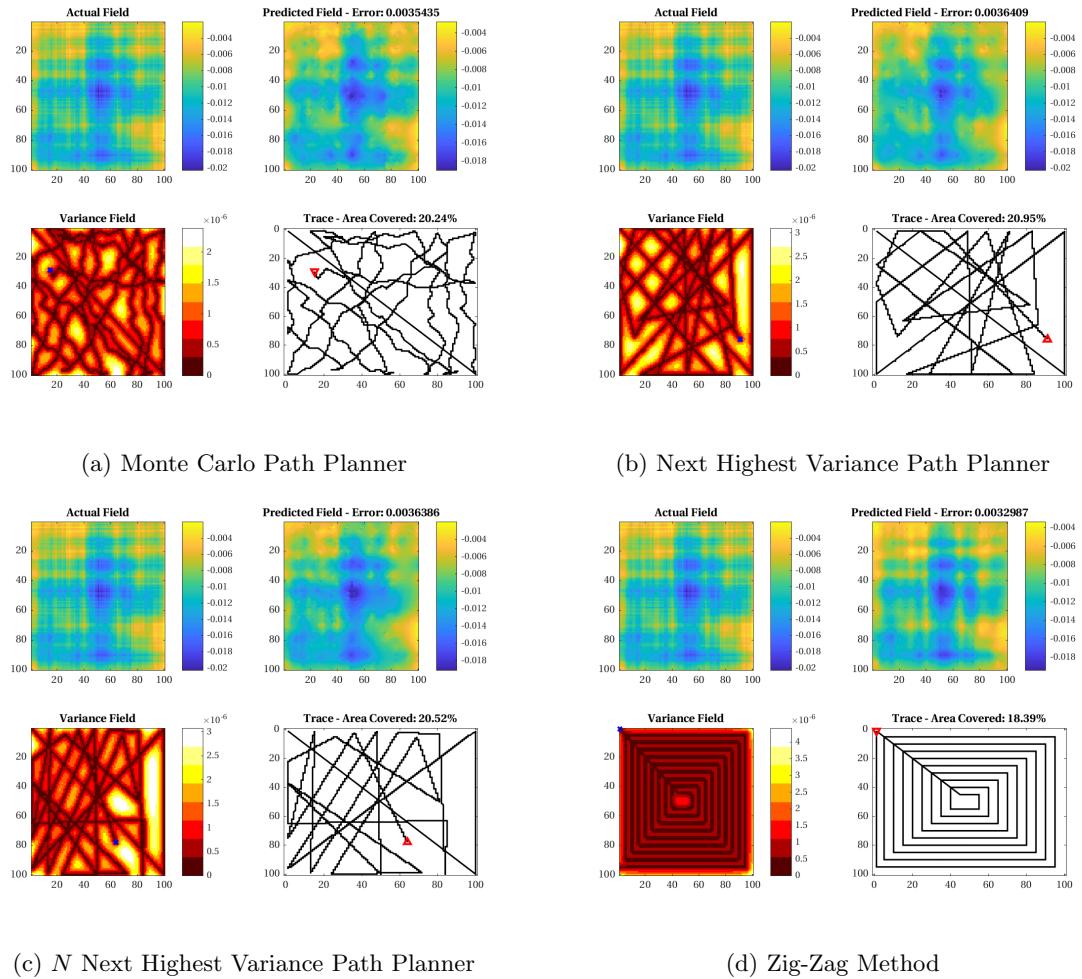
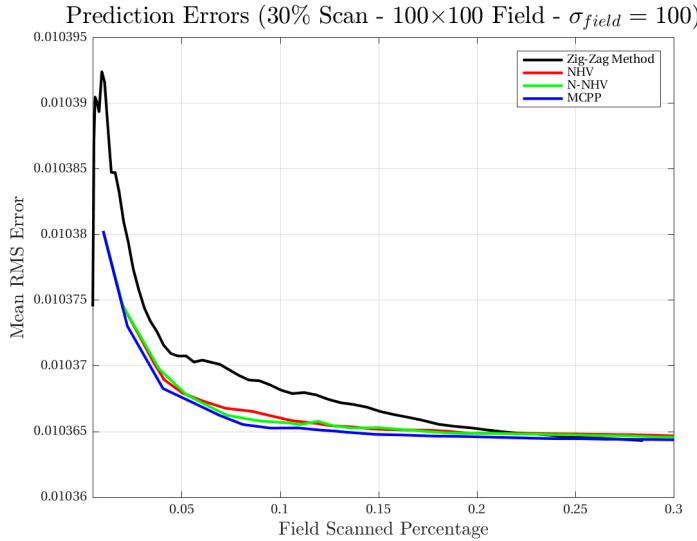


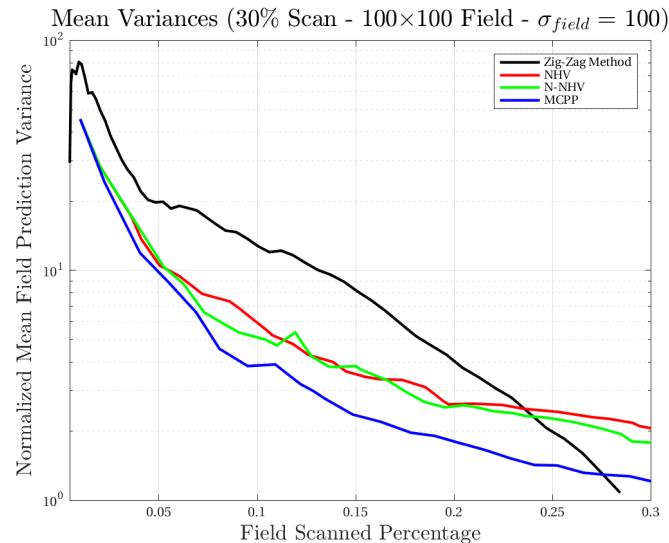
Figure 9.5: Simulation output for a 20% maximum area scan on a field of size 100×100 ,

$\sigma_{field} = 100$, random seed: 2.

9.2.3 30% Maximum Field Scan



(a) Prediction errors ($\text{erf}(Z, \hat{Z})$).



(b) Semi-logarithmic prediction variances normalized to an a priori mean variance for the field.

Figure 9.6: A 30% maximum area scan on a field of size 100×100 , $\sigma_{field} = 100$, random seed: 2.

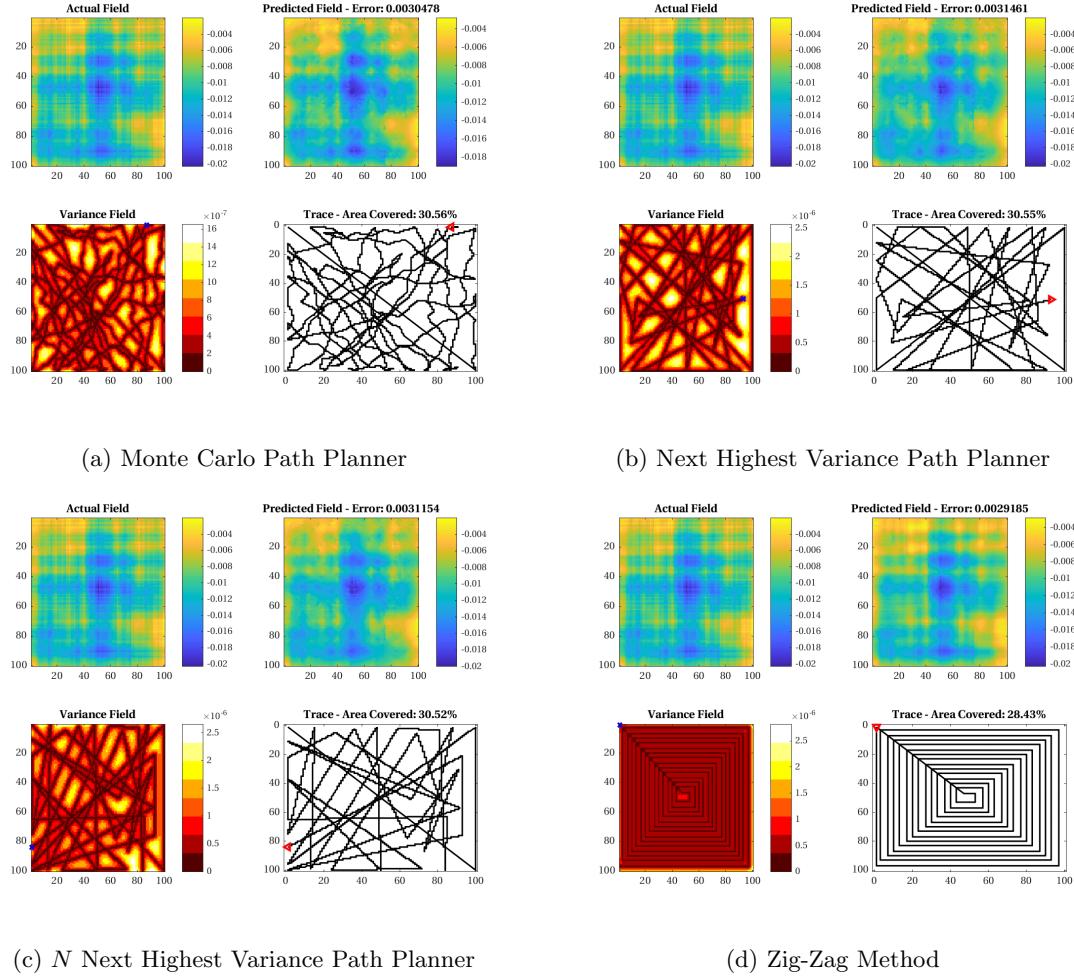


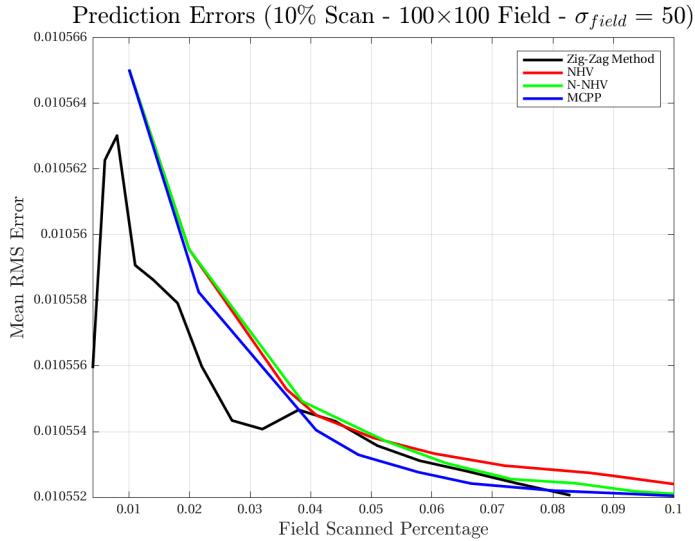
Figure 9.7: Simulation output for a 30% maximum area scan on a field of size 100×100 ,

$\sigma_{field} = 100$, random seed: 2.

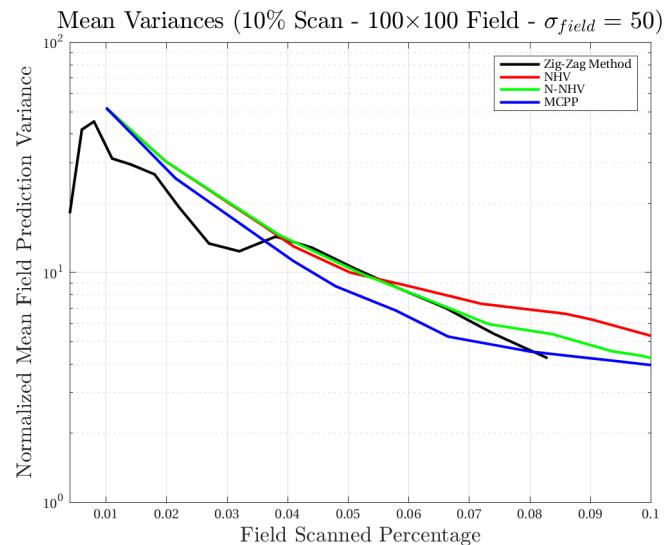
9.3 Half Width Spatial Autocorrelation ($\sigma_{field} = 0.5w$) Results

The methods will be compared on target fields generated with an autocorrelation factor that is half of the field width.

9.3.1 10% Maximum Field Scan



(a) Prediction errors ($\text{erf}(Z, \hat{Z})$).



(b) Semi-logarithmic prediction variances normalized to an a priori mean variance for the field.

Figure 9.8: A 10% maximum area scan on a field of size 100×100 , $\sigma_{field} = 50$, random seed: 2.

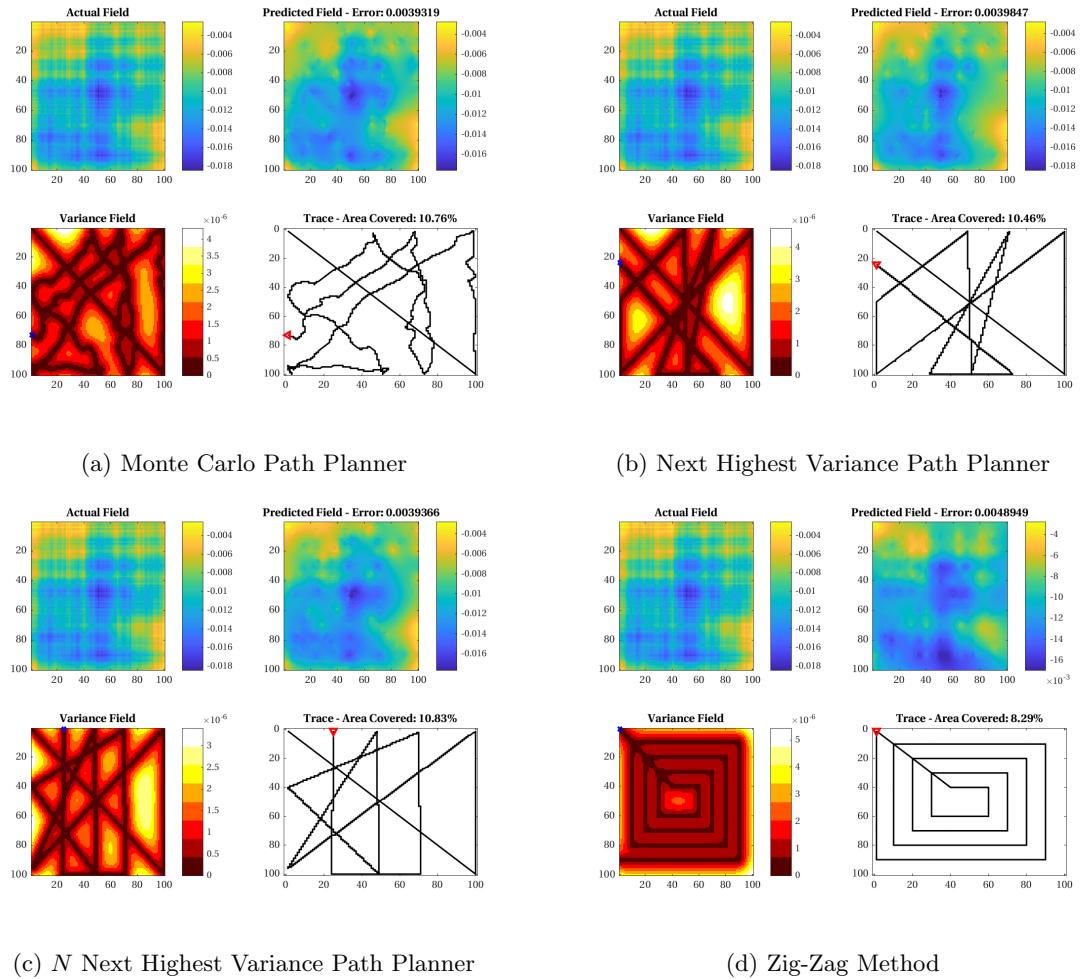
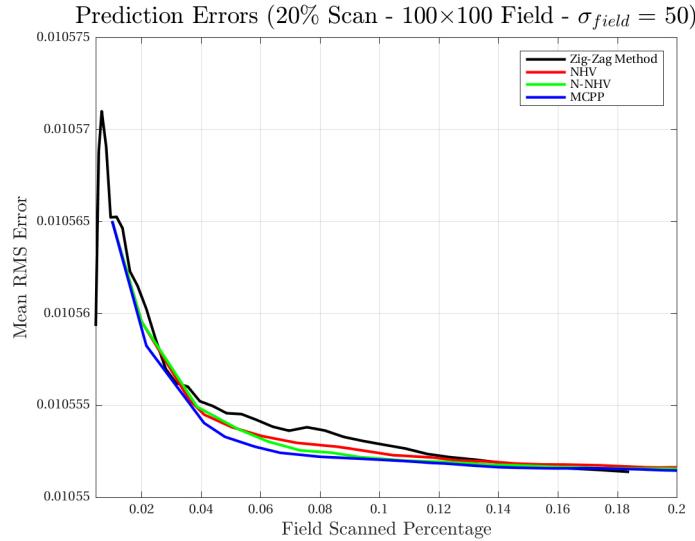


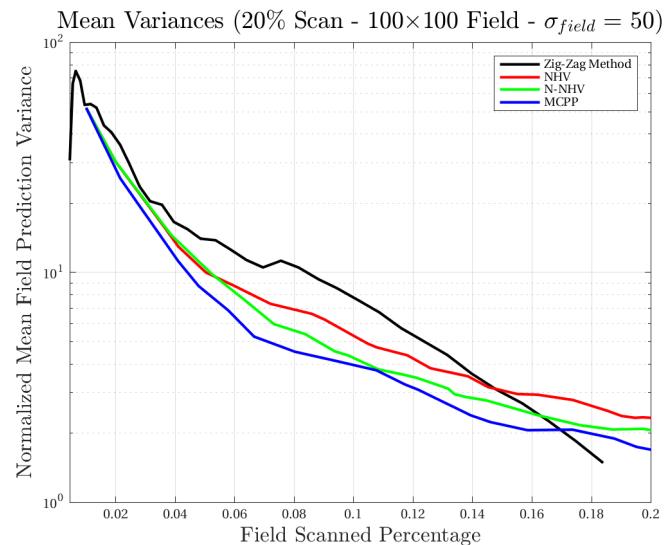
Figure 9.9: Simulation output for a 10% maximum area scan on a field of size 100×100 ,

$\sigma_{field} = 50$, random seed: 2.

9.3.2 20% Maximum Field Scan



(a) Prediction errors ($\text{erf}(Z, \hat{Z})$).



(b) Semi-logarithmic prediction variances normalized to an a priori mean variance for the field.

Figure 9.10: A 20% maximum area scan on a field of size 100×100 , $\sigma_{field} = 50$, random seed: 2.

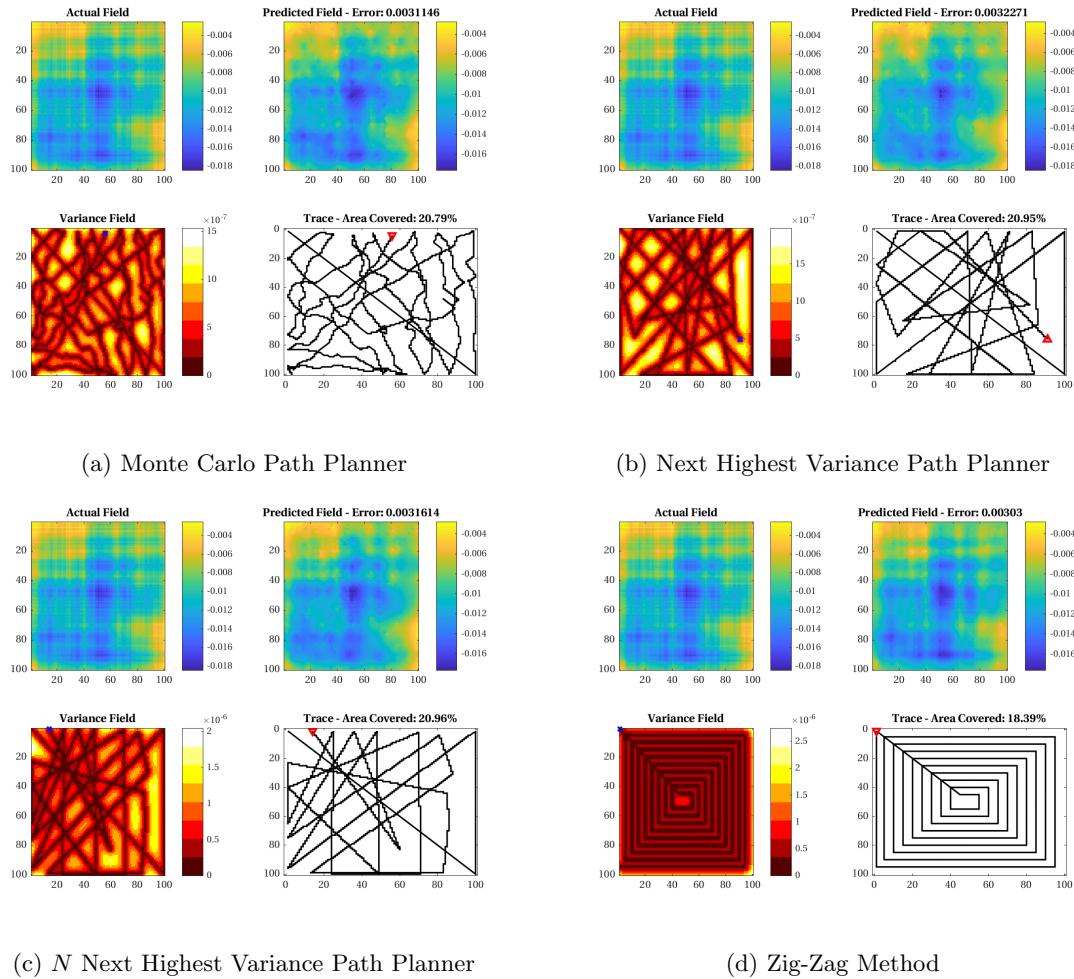
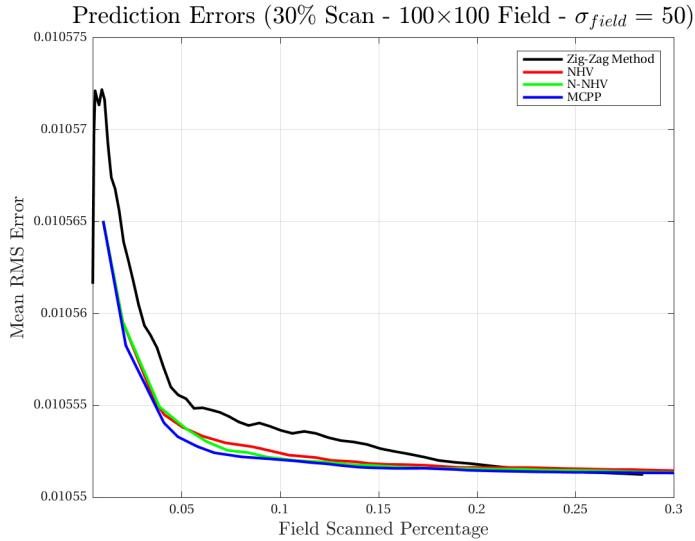


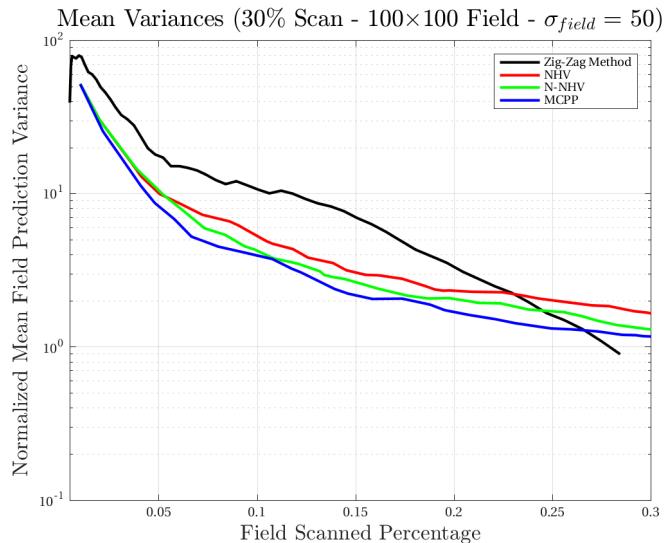
Figure 9.11: Simulation output for a 20% maximum area scan on a field of size 100×100 ,

$\sigma_{field} = 50$, random seed: 2.

9.3.3 30% Maximum Field Scan



(a) Prediction errors ($\text{erf}(Z, \hat{Z})$).



(b) Semi-logarithmic prediction variances normalized to an a priori mean variance for the field.

Figure 9.12: A 30% maximum area scan on a field of size 100×100 , $\sigma_{field} = 50$, random seed: 2.

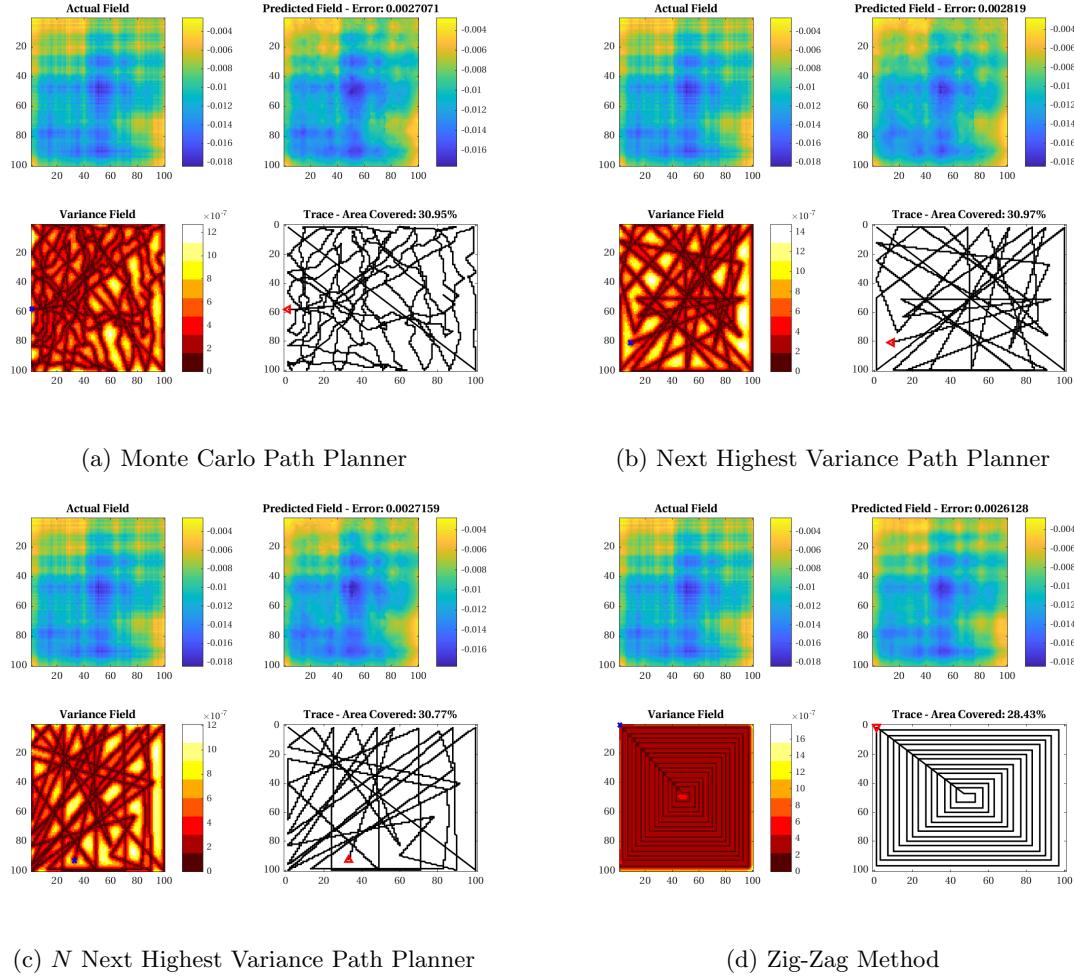


Figure 9.13: Simulation output for a 30% maximum area scan on a field of size 100×100 ,

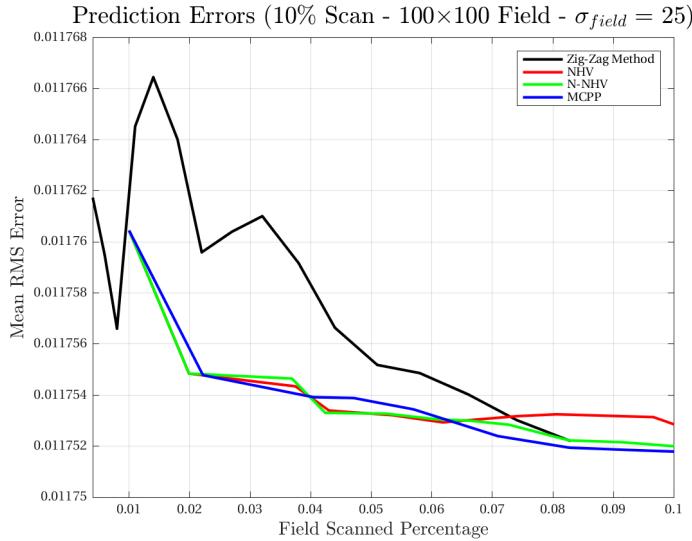
$\sigma_{field} = 50$, random seed: 2.

9.4 Quarter Width Spatial Autocorrelation ($\sigma_{field} = 0.25w$)

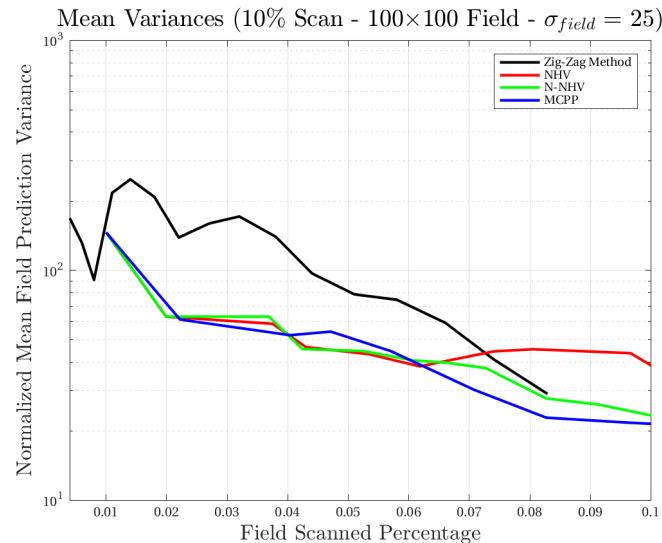
Results

The methods will be compared on target fields generated with an autocorrelation factor that is one quarter of the field width.

9.4.1 10% Maximum Field Scan



(a) Prediction errors ($\text{erf}(Z, \hat{Z})$).



(b) Semi-logarithmic prediction variances normalized to an a priori mean variance for the field.

Figure 9.14: A 10% maximum area scan on a field of size 100×100 , $\sigma_{field} = 25$, random seed: 2.

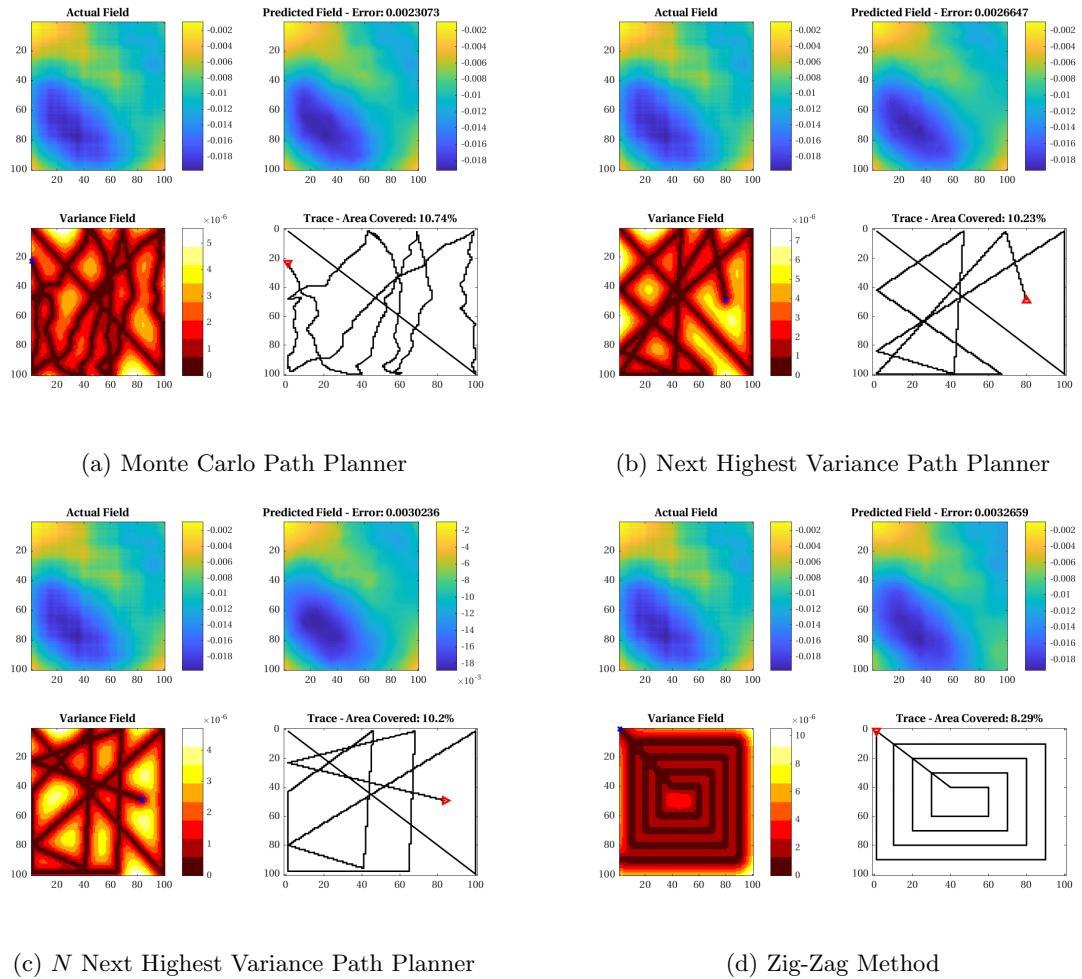
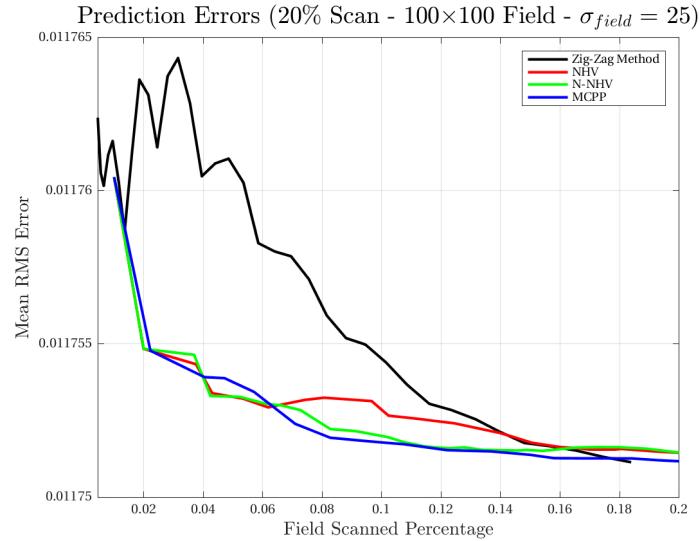


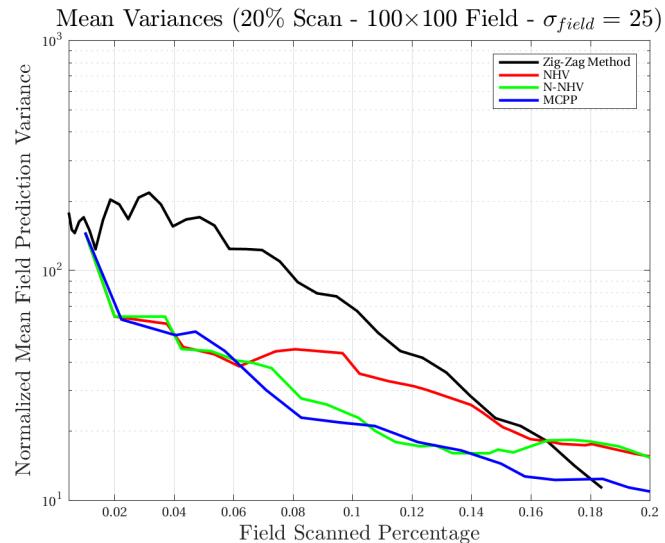
Figure 9.15: Simulation output for a 10% maximum area scan on a field of size 100×100 ,

$\sigma_{field} = 25$, random seed: 2.

9.4.2 20% Maximum Field Scan



(a) Prediction errors ($\text{erf}(Z, \hat{Z})$).



(b) Semi-logarithmic prediction variances normalized to an a priori mean variance for the field.

Figure 9.16: A 20% maximum area scan on a field of size 100×100 , $\sigma_{field} = 25$, random seed: 2.

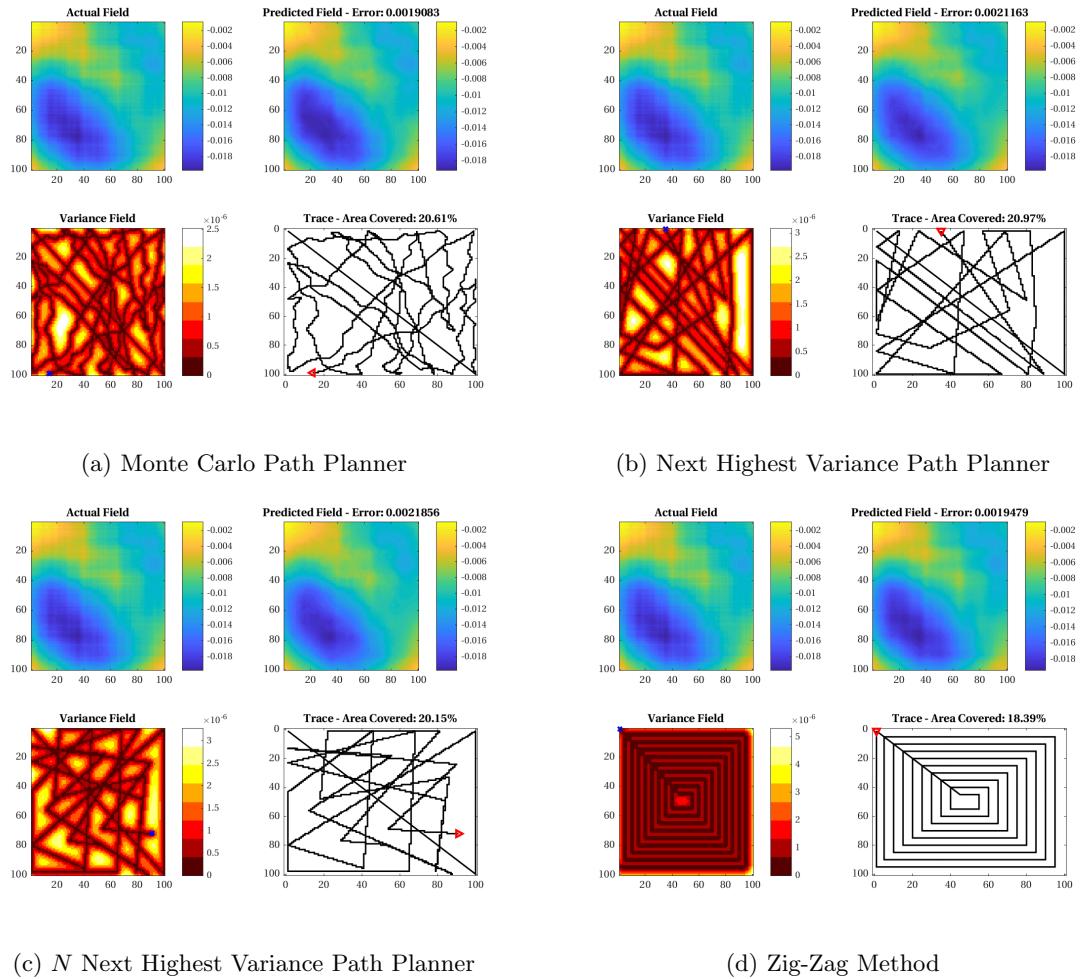
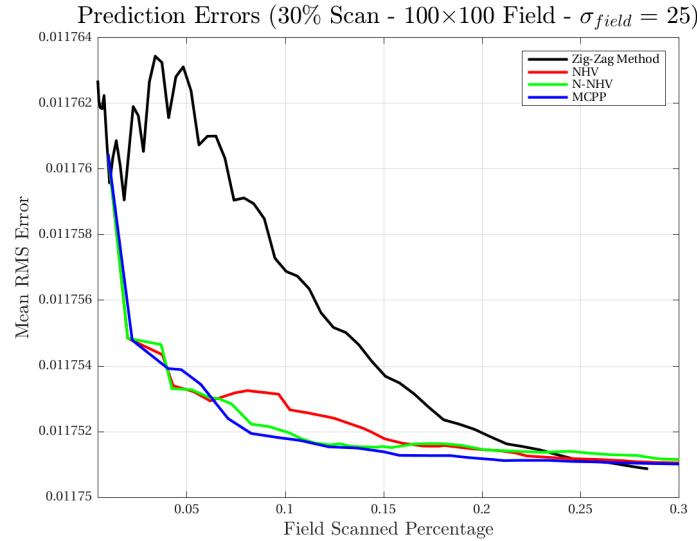


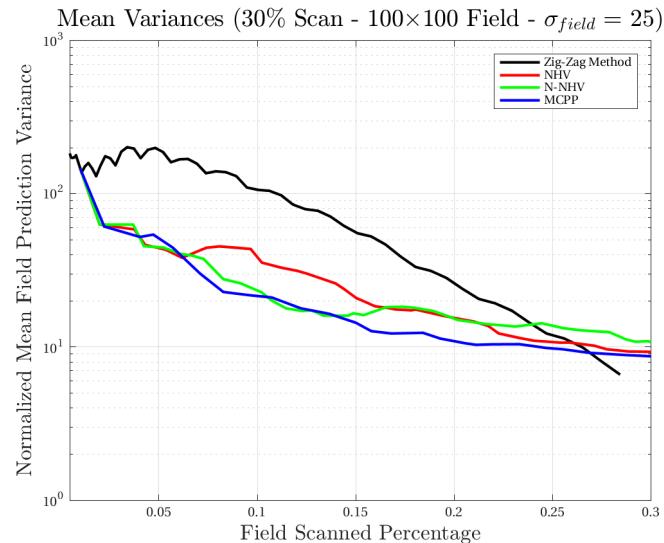
Figure 9.17: Simulation output for a 20% maximum area scan on a field of size 100×100 ,

$\sigma_{field} = 25$, random seed: 2.

9.4.3 30% Maximum Field Scan



(a) Prediction errors ($\text{erf}(Z, \hat{Z})$).



(b) Semi-logarithmic prediction variances normalized to an a priori mean variance for the field.

Figure 9.18: A 30% maximum area scan on a field of size 100×100 , $\sigma_{field} = 25$, random seed: 2.

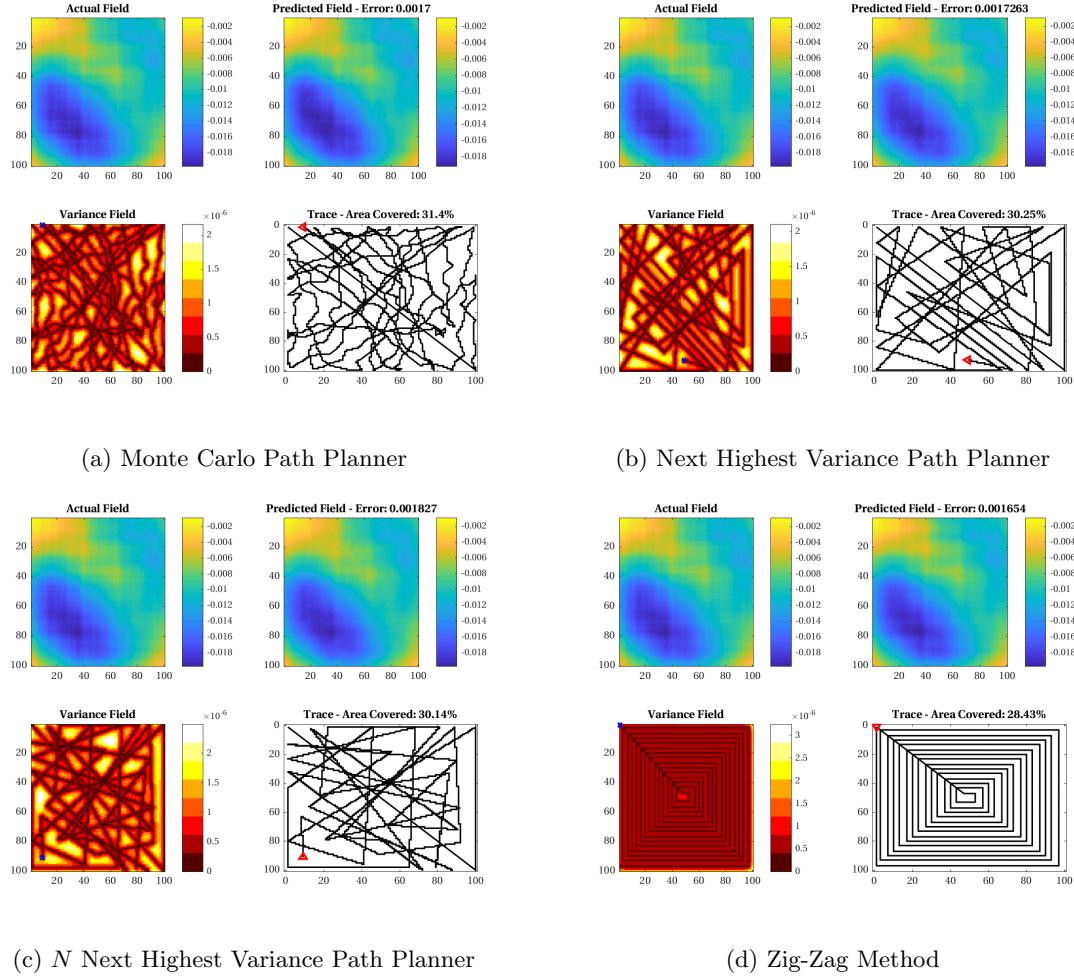


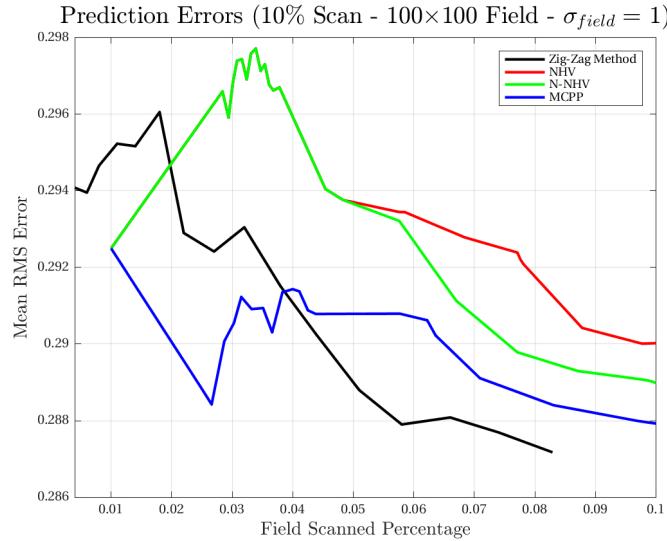
Figure 9.19: Simulation output for a 30% maximum area scan on a field of size 100×100 ,

$\sigma_{field} = 25$, random seed: 2.

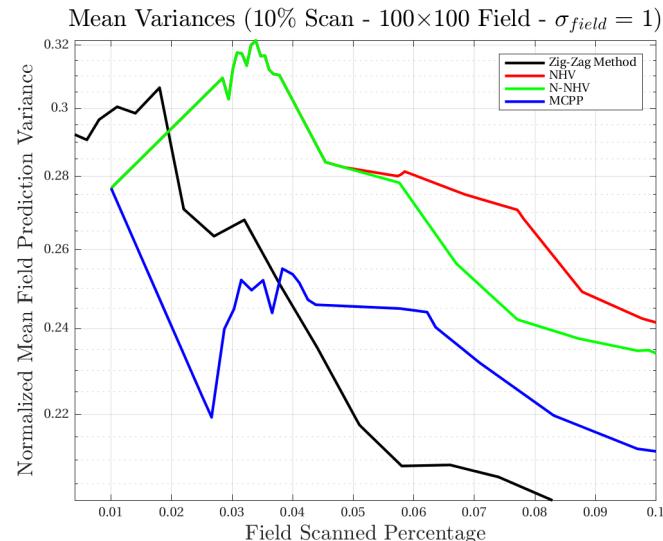
9.5 Low Spatial Autocorrelation ($\sigma_{field} = 1$) Results

The methods will be compared on target fields generated with a low autocorrelation factor.

9.5.1 10% Maximum Field Scan



(a) Prediction errors ($\text{erf}(Z, \hat{Z})$).



(b) Semi-logarithmic prediction variances normalized to an a priori mean variance for the field.

Figure 9.20: A 10% maximum area scan on a field of size 100×100 , $\sigma_{\text{field}} = 1$, random seed: 2.

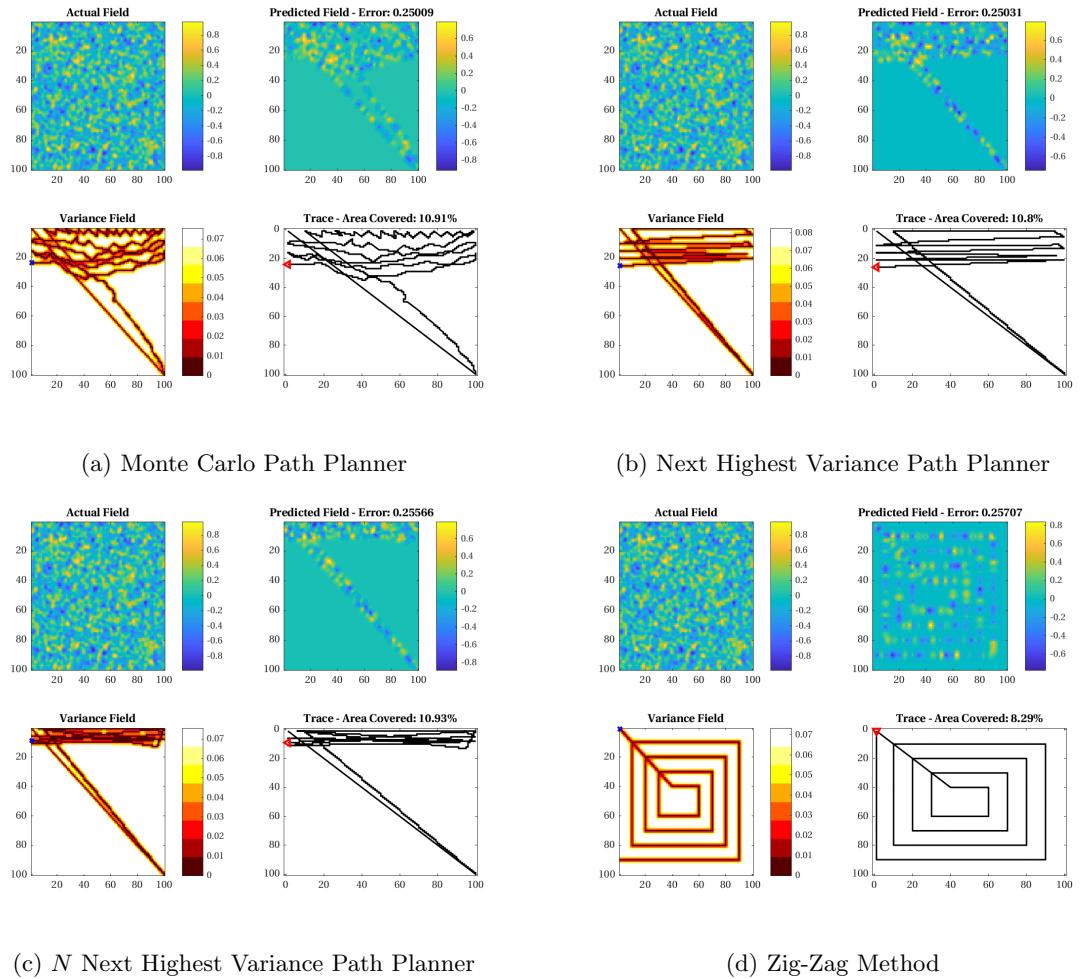
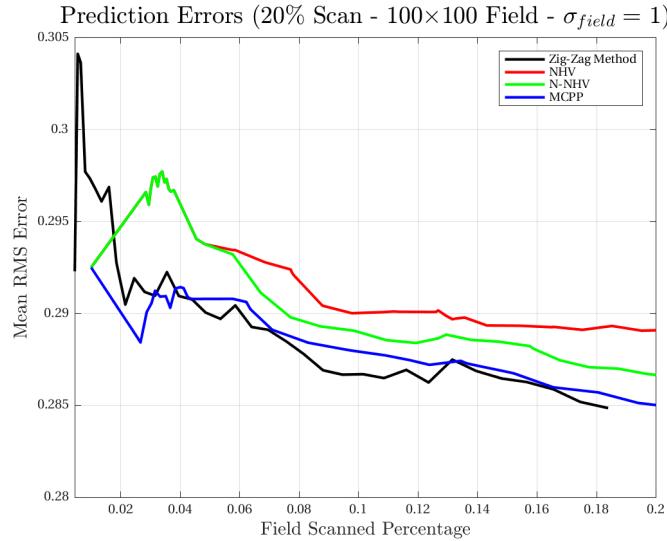


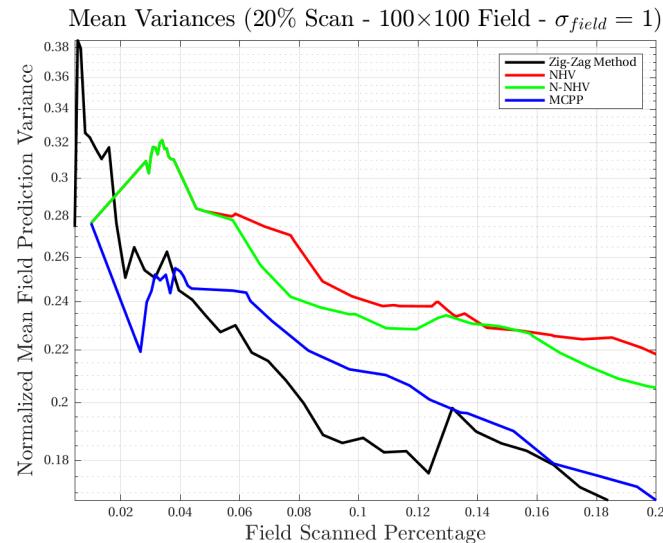
Figure 9.21: Simulation output for a 10% maximum area scan on a field of size 100×100 ,

$\sigma_{field} = 1$, random seed: 2.

9.5.2 20% Maximum Field Scan



(a) Prediction errors ($\text{erf}(Z, \hat{Z})$).



(b) Semi-logarithmic prediction variances normalized to an a priori mean variance for the field.

Figure 9.22: A 20% maximum area scan on a field of size 100×100 , $\sigma_{\text{field}} = 1$, random seed: 2.

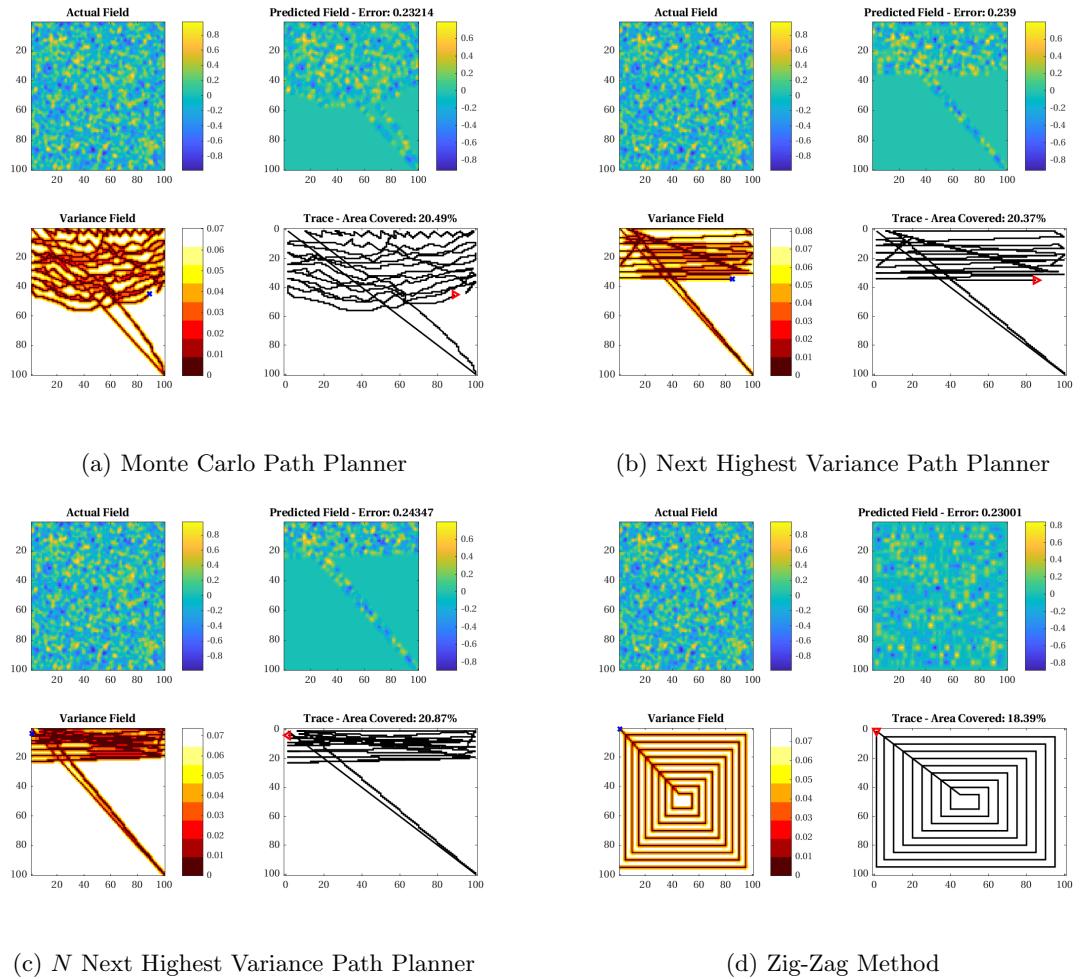
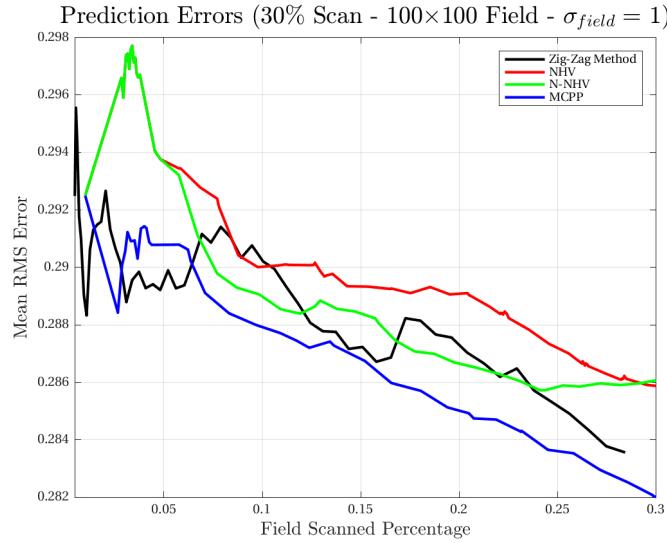


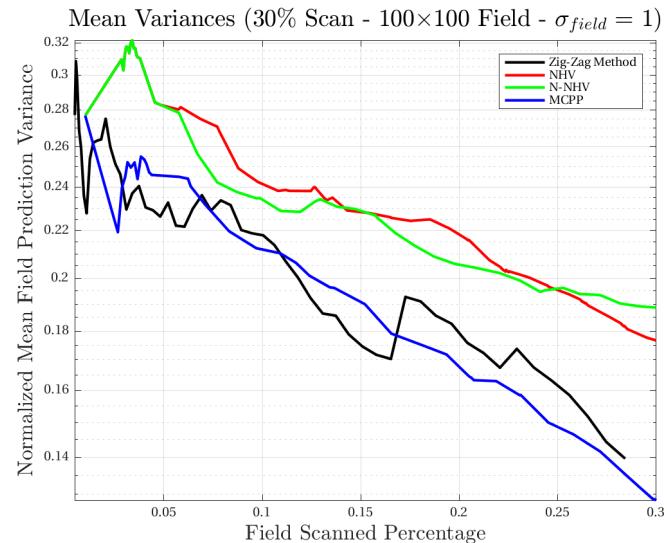
Figure 9.23: Simulation output for a 20% maximum area scan on a field of size 100×100 ,

$\sigma_{field} = 1$, random seed: 2.

9.5.3 30% Maximum Field Scan



(a) Prediction errors ($\text{erf}(Z, \hat{Z})$).



(b) Semi-logarithmic prediction variances normalized to an a priori mean variance for the field.

Figure 9.24: A 30% maximum area scan on a field of size 100×100 , $\sigma_{\text{field}} = 1$, random seed: 2.

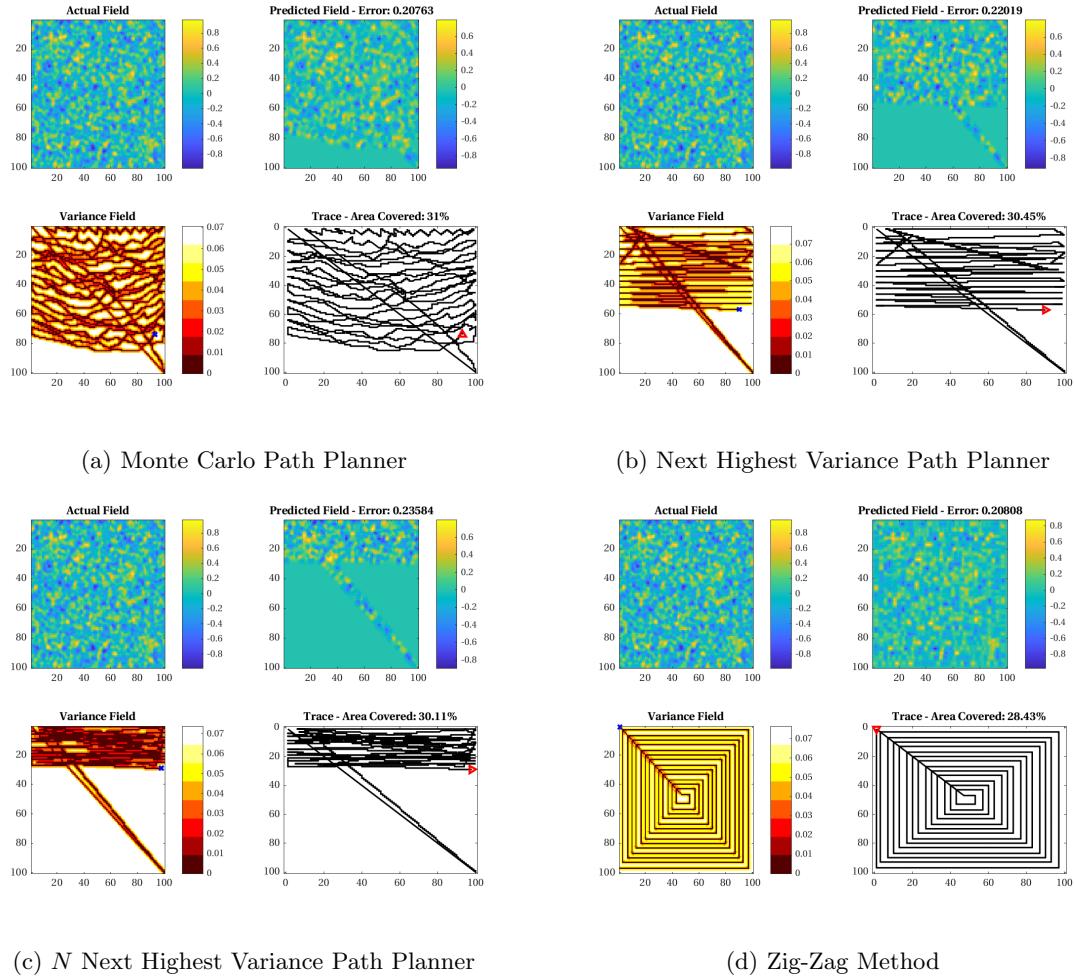


Figure 9.25: Simulation output for a 30% maximum area scan on a field of size 100×100 ,

$\sigma_{field} = 1$, random seed: 2.

9.6 Comparing The Methods

100 × 100 Size Final Field Prediction Errors ($\sigma_{field} = 1$)				
Coverage Limit (A_{scan})		10%	20%	30%
Zig-Zag	—	—	—	—
NHV	—	—	—	—
N-NHV	—	—	—	—
MCPP	—	—	—	—

Table 9.1: Comparing field prediction errors for varying coverage limitations on a 100×100 size field ($\sigma_{field} = 1$).

100 × 100 Size Final Field Prediction Variances ($\sigma_{field} = 1$)				
Coverage Limit (A_{scan})		10%	20%	30%
Zig-Zag	—	—	—	—
NHV	—	—	—	—
N-NHV	—	—	—	—
M CPP	—	—	—	—

Table 9.2: Comparing field prediction variances for varying coverage limitations on a 100×100 size field ($\sigma_{field} = 1$).

100 × 100 Size Final Field Prediction Errors ($\sigma_{field} = 100$)				
Coverage Limit (A_{scan})		10%	20%	30%
Zig-Zag	—	—	—	—
NHV	—	—	—	—
N-NHV	—	—	—	—
M CPP	—	—	—	—

Table 9.3: Comparing field prediction errors for varying coverage limitations on a 100×100 size field ($\sigma_{field} = 100$).

100 × 100 Size Final Field Prediction Variances ($\sigma_{field} = 100$)				
Coverage Limit (A_{scan})	10%	20%	30%	
Zig-Zag	—	—	—	
NHV	—	—	—	
N-NHV	—	—	—	
MCPP	—	—	—	

Table 9.4: Comparing field prediction variances for varying coverage limitations on a 100×100 size field ($\sigma_{field} = 100$).

Conclusion

The potential in a procedure using the Kriging Method as the core of a field exploration technique with an autonomous vehicle was demonstrated. By characterizing the confidence of the Kriging predictions made from observations in a field, along with uncertainty suppressing motivated path planners, the overall confidence in prediction of a target field as a whole can be maximized without having to scan every point. Furthermore, variance motivated path planning for prediction uncertainty suppression was shown to yield better results for equal path lengths compared to predetermined paths.

An exploration vehicle could be maneuvered through a field to collect samples in areas of low Kriging prediction confidence. This in turn can increase the quality of prediction of the target field's state of interest to a higher degree of certainty. Using the Kriging variance generated path planning techniques, a more accurate prediction of a target field, though explored with the same path lengths as a fixed path spiral formation, was generated.

By simulating the three introduced path planners as well as the predetermined zig-zag method, it was shown that the prediction errors of a variance motivated path planner outperformed that of a predetermined maneuver. For highly autocorrelated target fields, it was found that ... For target fields with low autocorrelation, it was found that ...

Future Work

Future works can be done in an effort to further develop Kriging prediction variance motivated path planning techniques. A method which filters noisy measurement on a target field, with potentially dynamic properties, using a Kalman Filtering technique may be in the interest of a variance motivated path planner. The field states of a potentially dynamic field can be estimated using a Kalman Filter. The variances from both the Kalman filter state estimates, along with the Kriging prediction variances, could then potentially be fused in an effort to better track the states of a dynamic field by constructing paths along the field that suppress overall Kalman filter variances and Kriging prediction variances. This would ideally, in turn, produce predictions with higher certainty of state estimate for the overall field. The method could then also be used to help determine if a vehicle is suitable for tracking a certain state about the field. For example, if the dynamics of the field are found to be too dynamic for a simulated vehicle with a fixed velocity to track, while minimizing Kriging prediction variances and Kalman state variances, a more suitable vehicle might be chosen instead for that field.

Further work could also be the development of a comparison of the introduced methods, for varying vehicle dynamics, e.g. a Dubins Vehicle.

Bibliography

- [1] C. C. Castello, J. Fan, A. Davari, and R. Chen. Optimal sensor placement strategy for environmental monitoring using wireless sensor networks. In *2010 42nd Southeastern Symposium on System Theory (SSST)*, pages 275–279, March 2010.
- [2] Howie Choset. Coverage for robotics – a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31(1):113–126, Oct 2001.
- [3] Noel A. C. Cressie. *Geostatistics*, pages 27–104. John Wiley & Sons, Inc., 2015.
- [4] Y.A. Felus, Alan Saalfeld, and B Schaffrin. Delaunay triangulation structured kriging for surface interpolation. 65:27–36, 03 2005.
- [5] Steven Fortune. *Handbook of Discrete and Computational Geometry*. CRC Press, Inc., Boca Raton, FL, USA, 1997.
- [6] Lucas Janson, Edward Schmerling, and Marco Pavone. *Monte Carlo Motion Planning for Robot Trajectory Optimization Under Uncertainty*, pages 343–361. Springer International Publishing, Cham, 2018.
- [7] Georges Matheron. Principles of geostatistics. *Economic Geology*, 58(8):1246–1266, 1963.
- [8] Harvey J. Miller. Tobler’s first law and spatial analysis. *Annals of the Association of American Geographers*, 94(2):284–289, 2004.
- [9] N. Nigam. Control and design of multiple unmanned air vehicles for persistent surveillance. 2009.
- [10] Nikhil Nigam. The multiple unmanned air vehicle persistent surveillance problem: A review. *Machines*, 2(1):13–72, 2014.
- [11] S. Rabinovich. Multi-uav path coordination based on uncertainty estimation. In *UC Santa Cruz: Division of Graduate Studies*, 2017.
- [12] S. Rabinovich, R. E. Curry, and G. H. Elkaim. A methodology for estimation of ground phenomena propagation. In *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pages 1239–1244, April 2018.

- [13] Eduard Semsch, Michal Jakob, Dušan Pavlíček, and Michal Pěchouček. Occlusion-aware multi-uav surveillance. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*, AAMAS '10, pages 1407–1408, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems.
- [14] Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM National Conference*, ACM '68, pages 517–524, New York, NY, USA, 1968. ACM.
- [15] W. R. Tobler. A computer movie simulating urban growth in the detroit region. *Economic Geography*, 46:234–240, 1970.
- [16] S.C. Van der Graaf. Natural neighbour kriging and its potential for quality mapping and grid design. *TU Delft Repository*, page 8, 2016.
- [17] Cheng Zhang and Hailong Pei. Oil spills boundary tracking using universal kriging and model predictive control by uav. In *Proceeding of the 11th World Congress on Intelligent Control and Automation*, pages 633–638, June 2014.

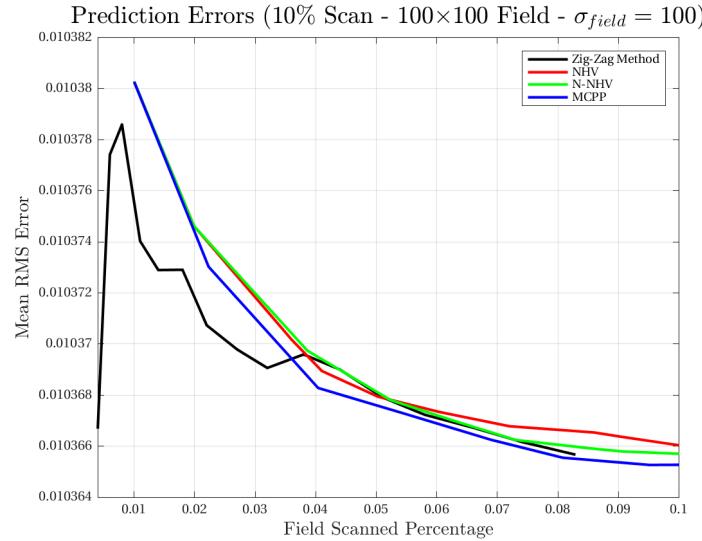
Appendix A

Results (Random Seed - 3)

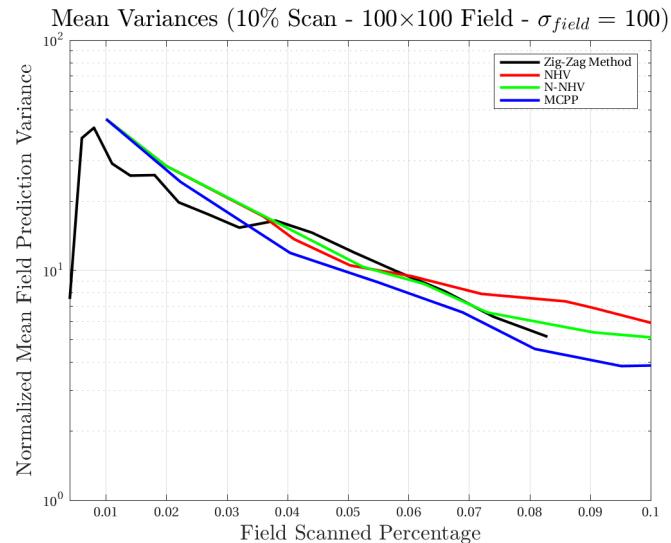
A.1 High Spatial Autocorrelation ($\sigma_{field} = w$) Results

The methods will be compared on target fields generated with the high degree of geospatial autocorrelation.

A.1.1 10% Maximum Field Scan



(a) Prediction errors ($\text{erf}(Z, \hat{Z})$).



(b) Semi-logarithmic prediction variances normalized to an a priori mean variance for the field.

Figure A.1: A 10% maximum area scan on a field of size 100×100 , $\sigma_{field} = 100$, random seed: 2.

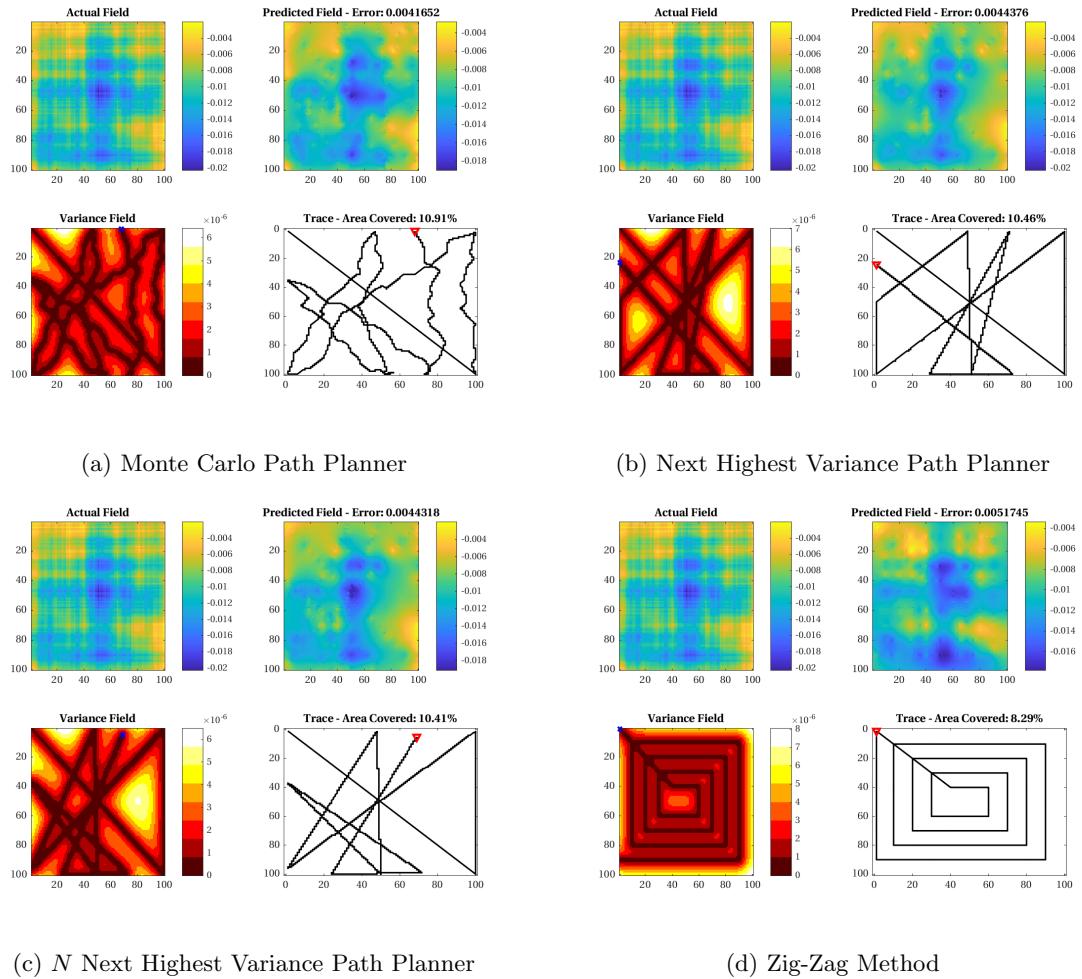
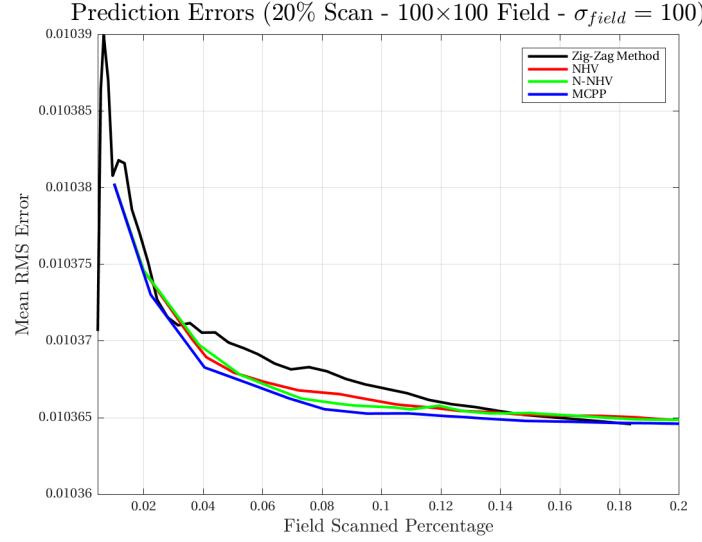


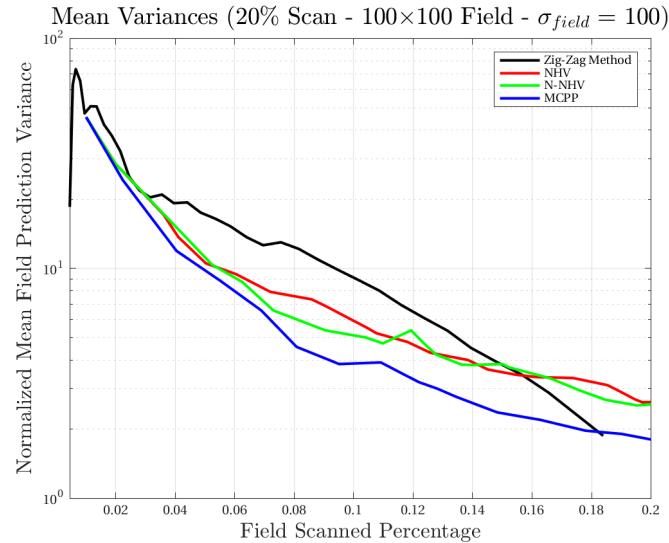
Figure A.2: Simulation output for a 10% maximum area scan on a field of size 100×100 ,

$\sigma_{field} = 100$, random seed: 2.

A.1.2 20% Maximum Field Scan



(a) Prediction errors ($\text{erf}(Z, \hat{Z})$).



(b) Semi-logarithmic prediction variances normalized to an a priori mean variance for the field.

Figure A.3: A 20% maximum area scan on a field of size 100×100 , $\sigma_{field} = 100$, random seed: 2.

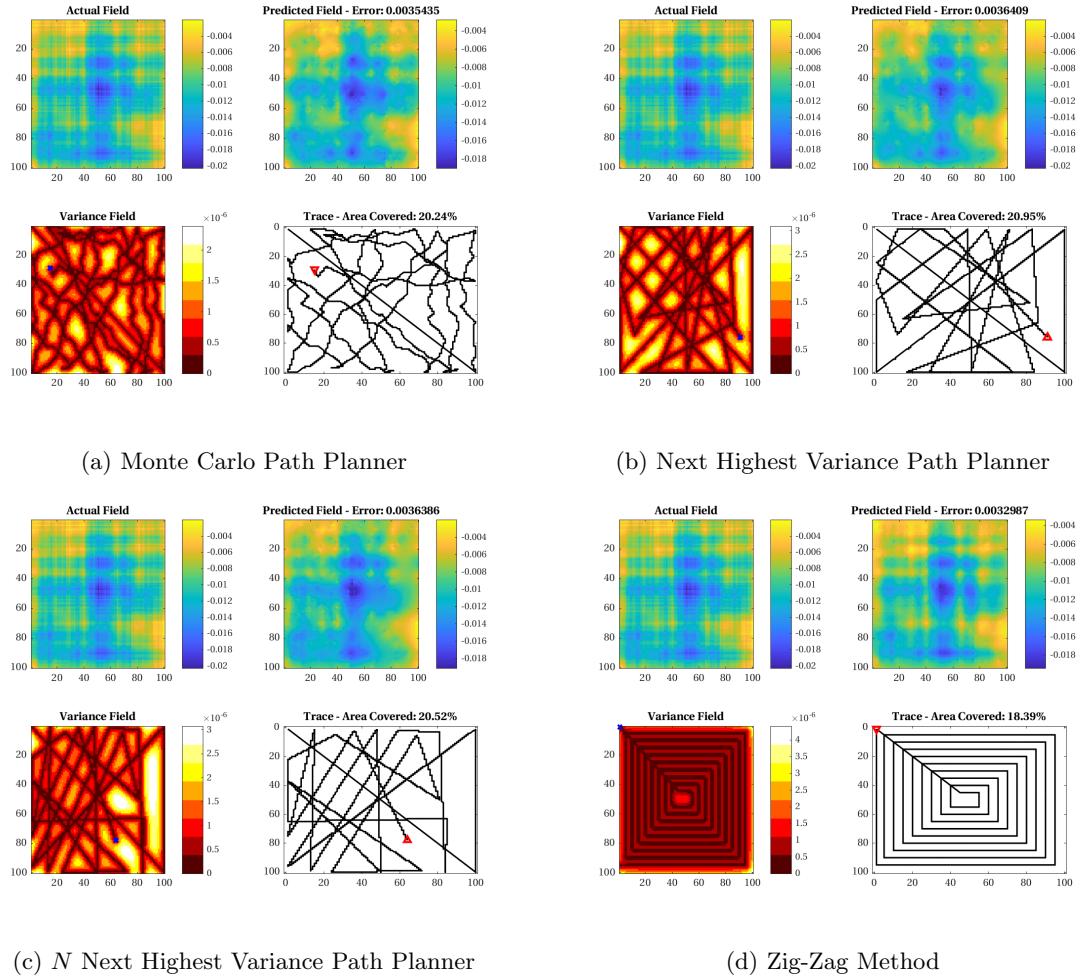
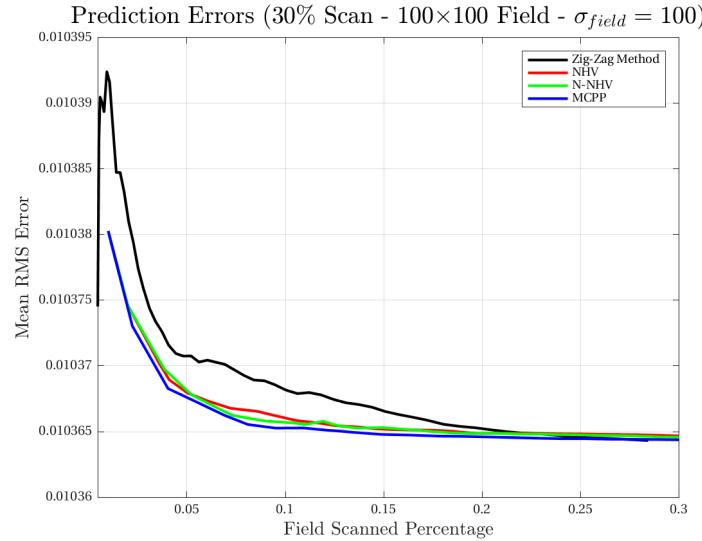


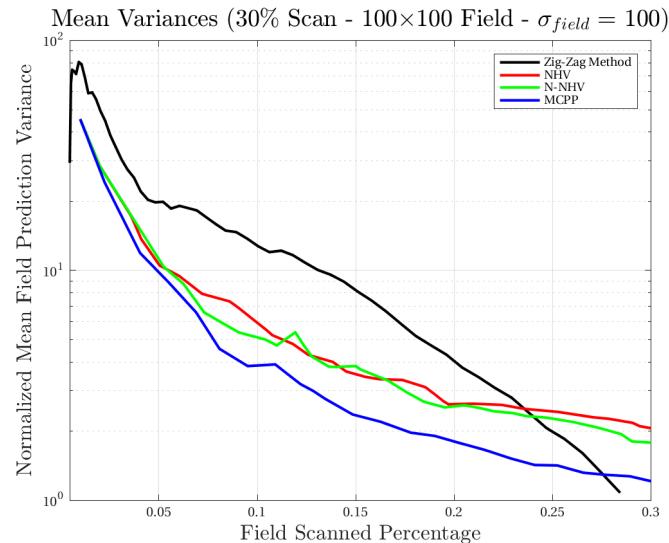
Figure A.4: Simulation output for a 20% maximum area scan on a field of size 100×100 ,

$\sigma_{field} = 100$, random seed: 2.

A.1.3 30% Maximum Field Scan



(a) Prediction errors ($\text{erf}(Z, \hat{Z})$).



(b) Semi-logarithmic prediction variances normalized to an a priori mean variance for the field.

Figure A.5: A 30% maximum area scan on a field of size 100×100 , $\sigma_{field} = 100$, random seed: 2.

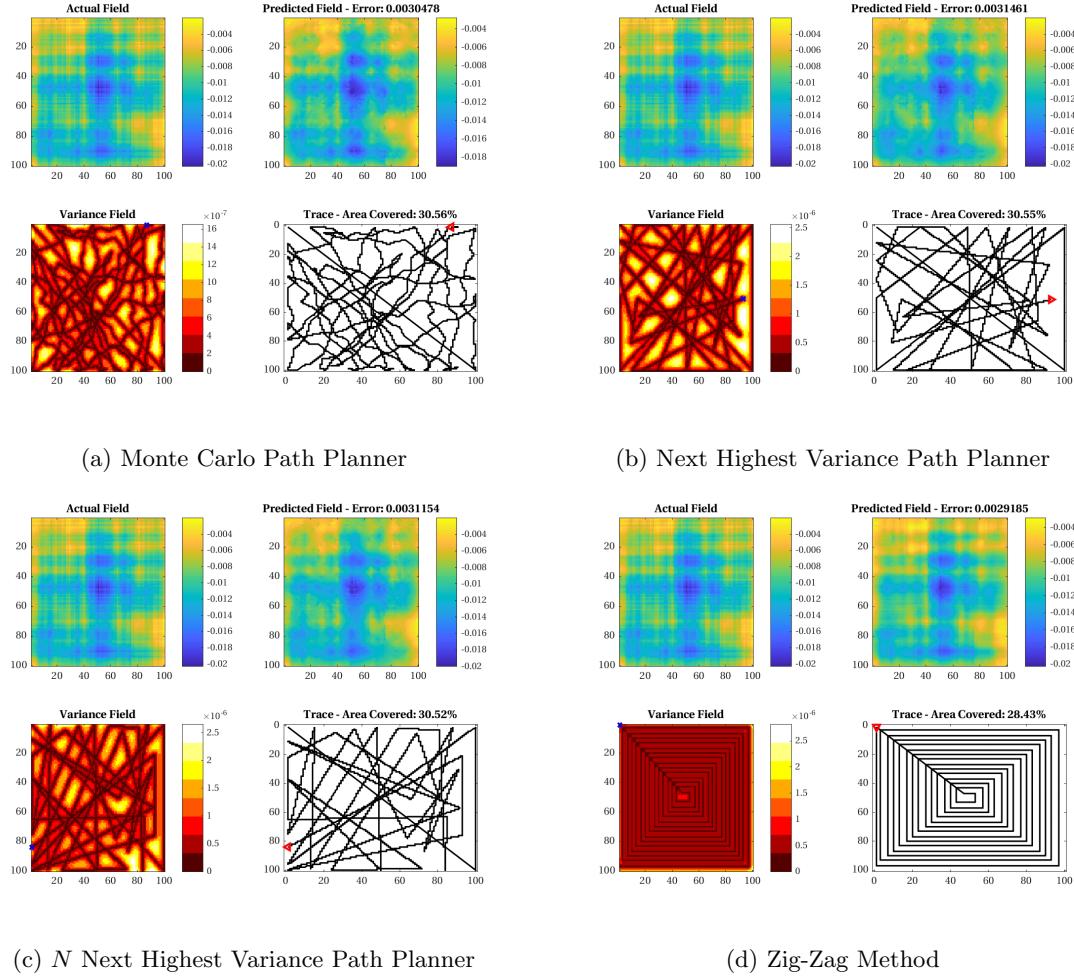
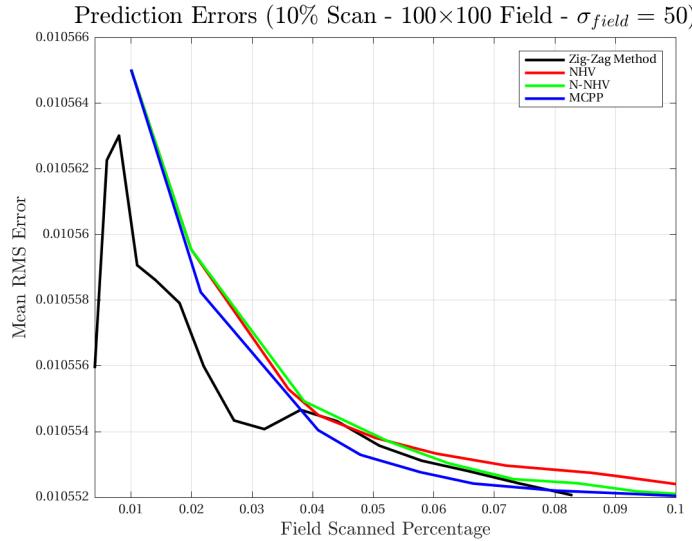


Figure A.6: Simulation output for a 30% maximum area scan on a field of size 100×100 , $\sigma_{field} = 100$, random seed: 2.

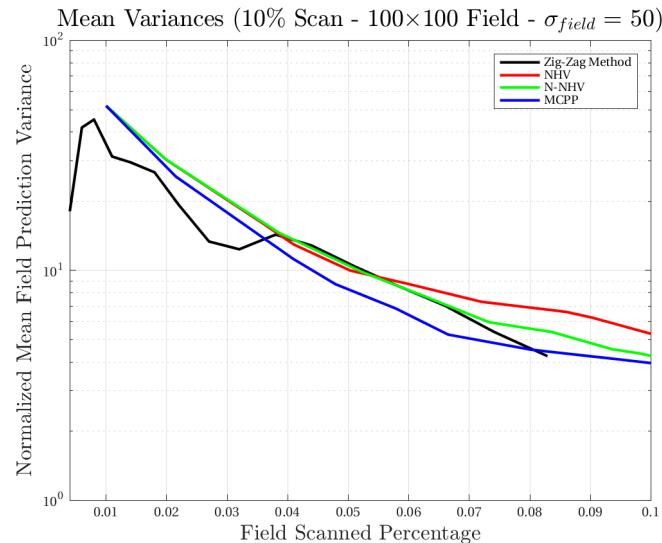
A.2 Half Width Spatial Autocorrelation ($\sigma_{field} = 0.5w$) Results

The methods will be compared on target fields generated with an autocorrelation factor that is half of the field width.

A.2.1 10% Maximum Field Scan



(a) Prediction errors ($\text{erf}(Z, \hat{Z})$).



(b) Semi-logarithmic prediction variances normalized to an a priori mean variance for the field.

Figure A.7: A 10% maximum area scan on a field of size 100×100 , $\sigma_{field} = 50$, random seed: 2.

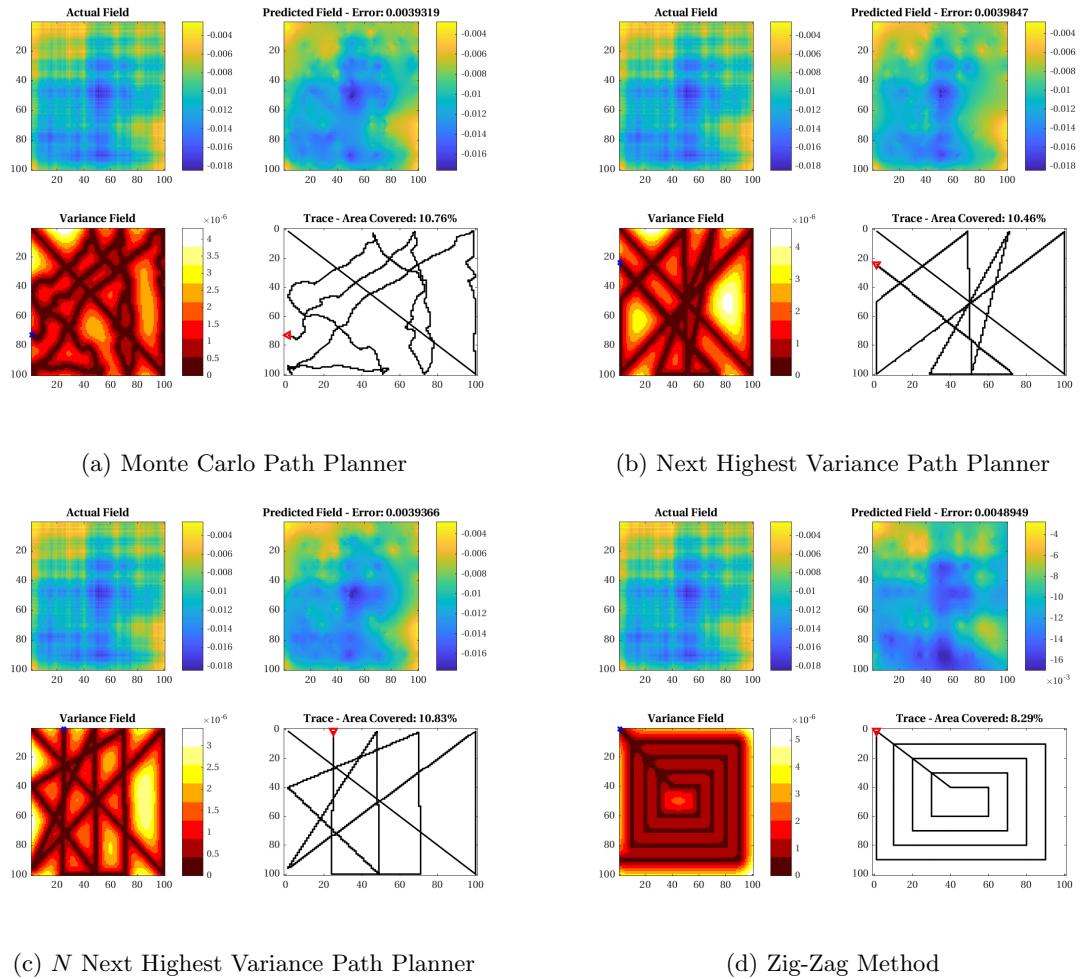
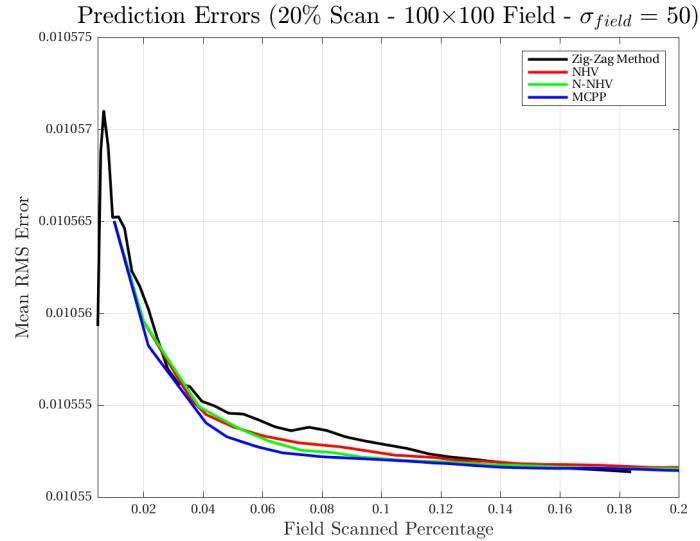


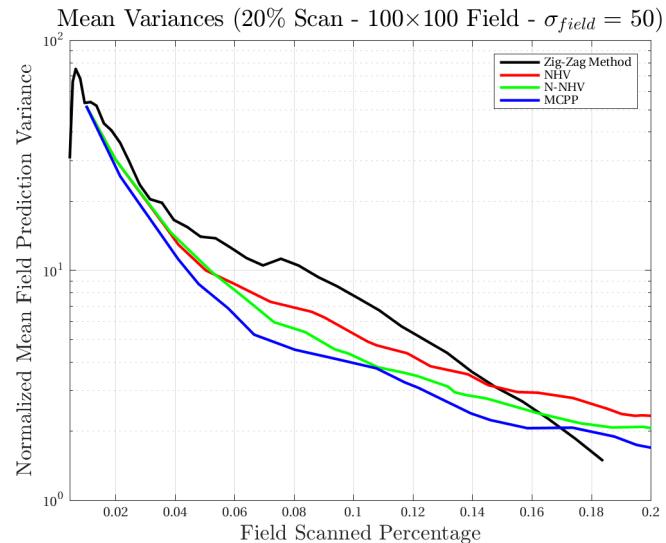
Figure A.8: Simulation output for a 10% maximum area scan on a field of size 100×100 ,

$\sigma_{field} = 50$, random seed: 2.

A.2.2 20% Maximum Field Scan



(a) Prediction errors ($\text{erf}(Z, \hat{Z})$).



(b) Semi-logarithmic prediction variances normalized to an a priori mean variance for the field.

Figure A.9: A 20% maximum area scan on a field of size 100×100 , $\sigma_{field} = 50$, random seed: 2.

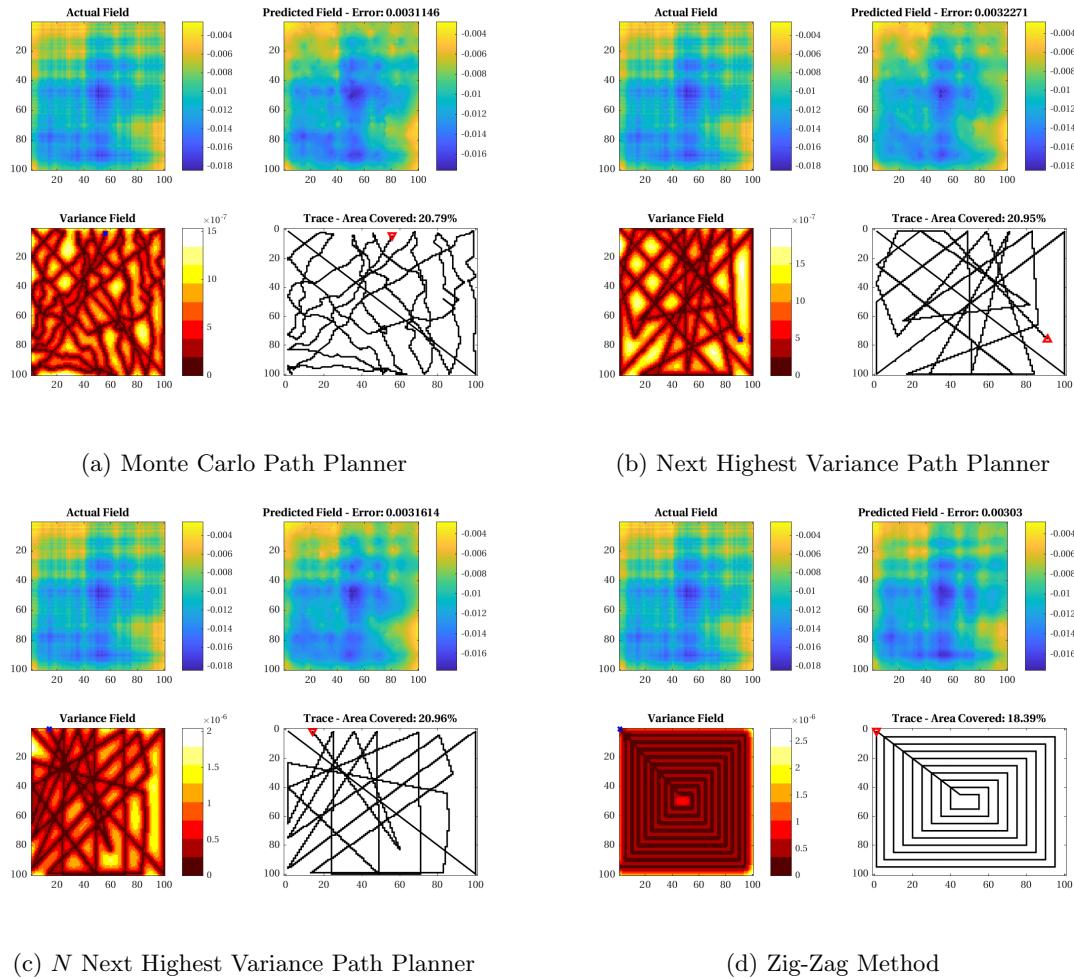
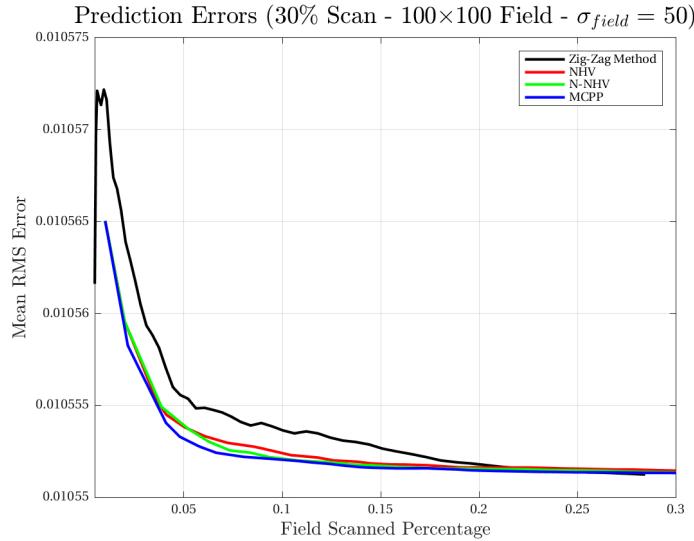


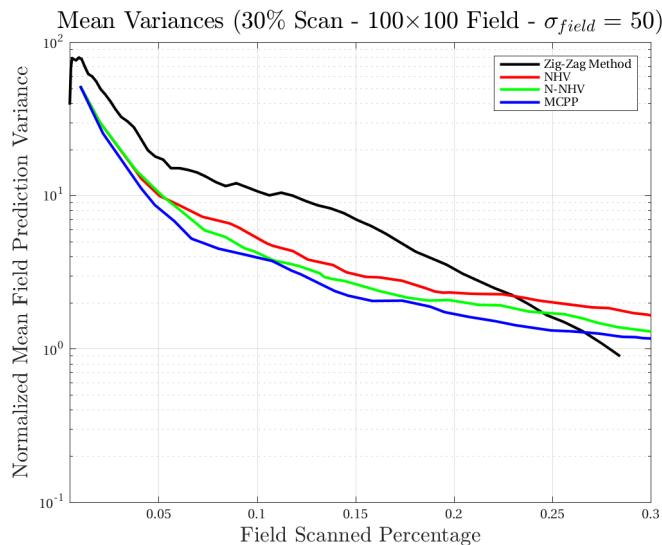
Figure A.10: Simulation output for a 20% maximum area scan on a field of size 100×100 ,

$\sigma_{field} = 50$, random seed: 2.

A.2.3 30% Maximum Field Scan



(a) Prediction errors ($\text{erf}(Z, \hat{Z})$).



(b) Semi-logarithmic prediction variances normalized to an a priori mean variance for the field.

Figure A.11: A 30% maximum area scan on a field of size 100×100 , $\sigma_{field} = 50$, random seed: 2.

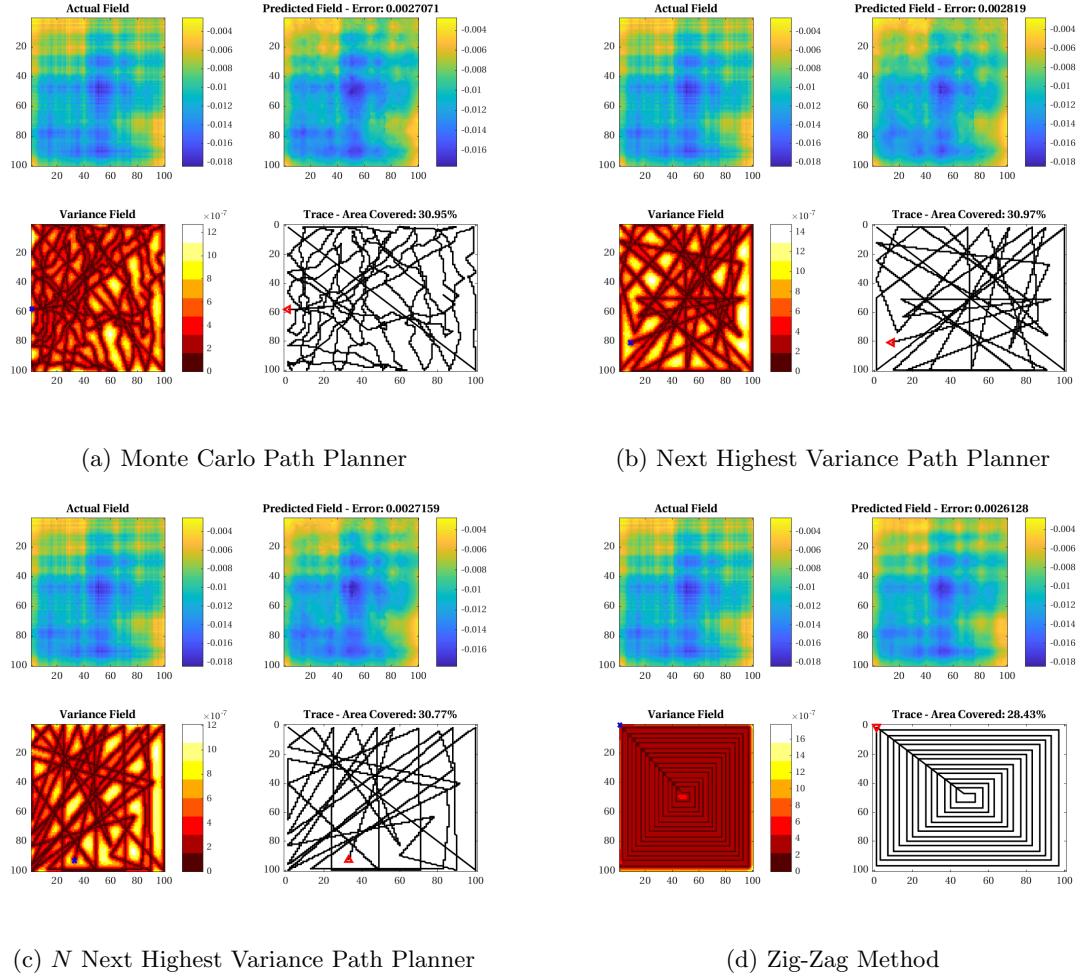


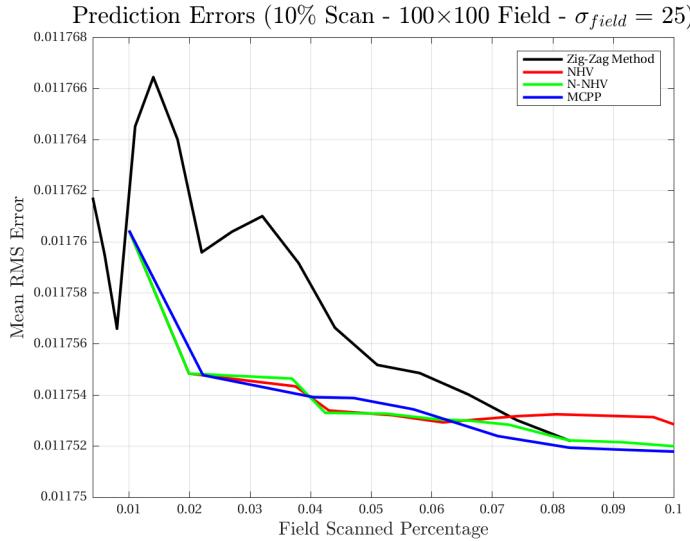
Figure A.12: Simulation output for a 30% maximum area scan on a field of size 100×100 , $\sigma_{field} = 50$, random seed: 2.

A.3 Quarter Width Spatial Autocorrelation ($\sigma_{field} = 0.25w$)

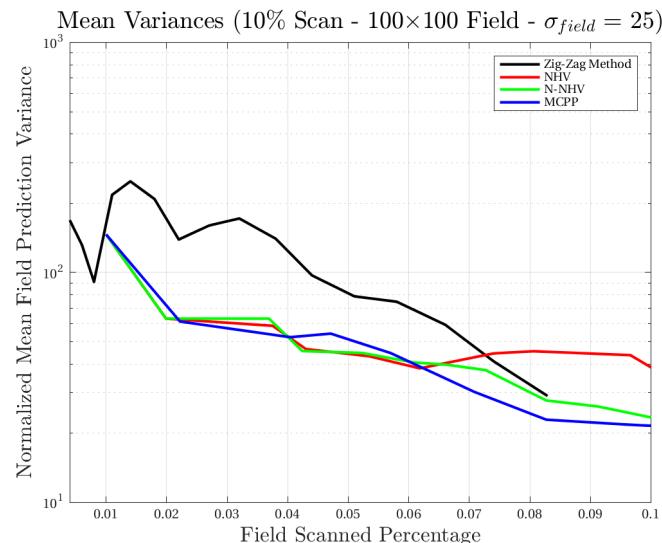
Results

The methods will be compared on target fields generated with an autocorrelation factor that is one quarter of the field width.

A.3.1 10% Maximum Field Scan



(a) Prediction errors ($\text{erf}(Z, \hat{Z})$).



(b) Semi-logarithmic prediction variances normalized to an a priori mean variance for the field.

Figure A.13: A 10% maximum area scan on a field of size 100×100 , $\sigma_{field} = 25$, random seed: 2.

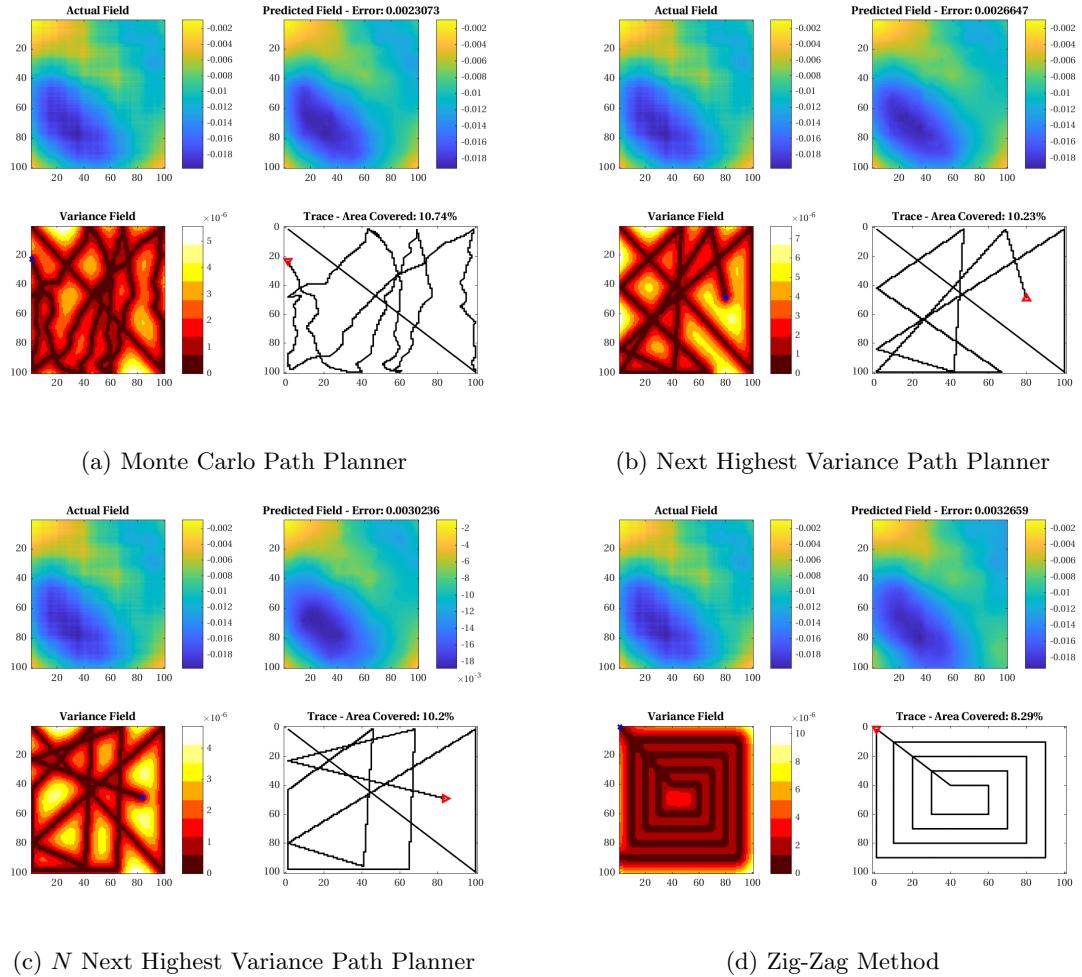
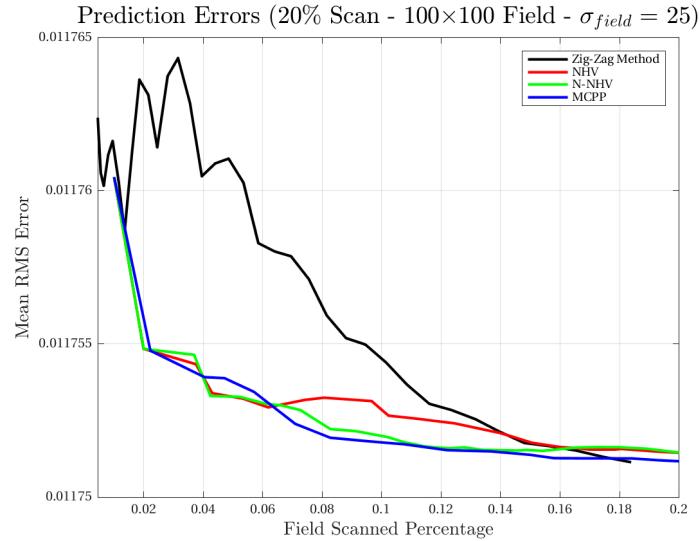


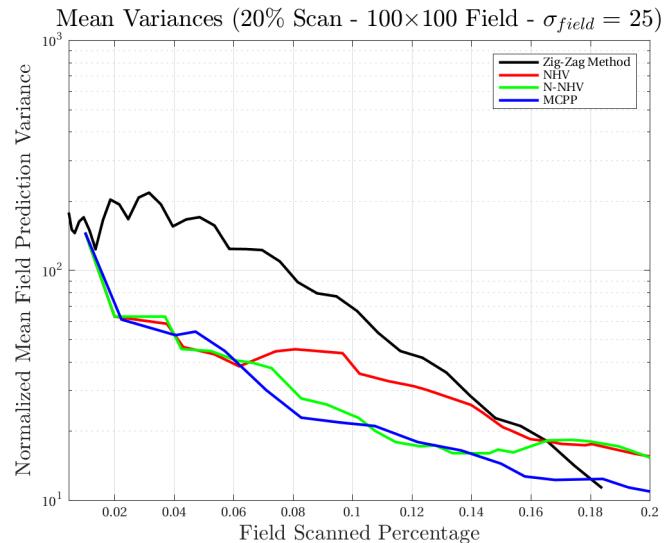
Figure A.14: Simulation output for a 10% maximum area scan on a field of size 100×100 ,

$\sigma_{field} = 25$, random seed: 2.

A.3.2 20% Maximum Field Scan



(a) Prediction errors ($\text{erf}(Z, \hat{Z})$).



(b) Semi-logarithmic prediction variances normalized to an a priori mean variance for the field.

Figure A.15: A 20% maximum area scan on a field of size 100×100 , $\sigma_{field} = 25$, random seed: 2.

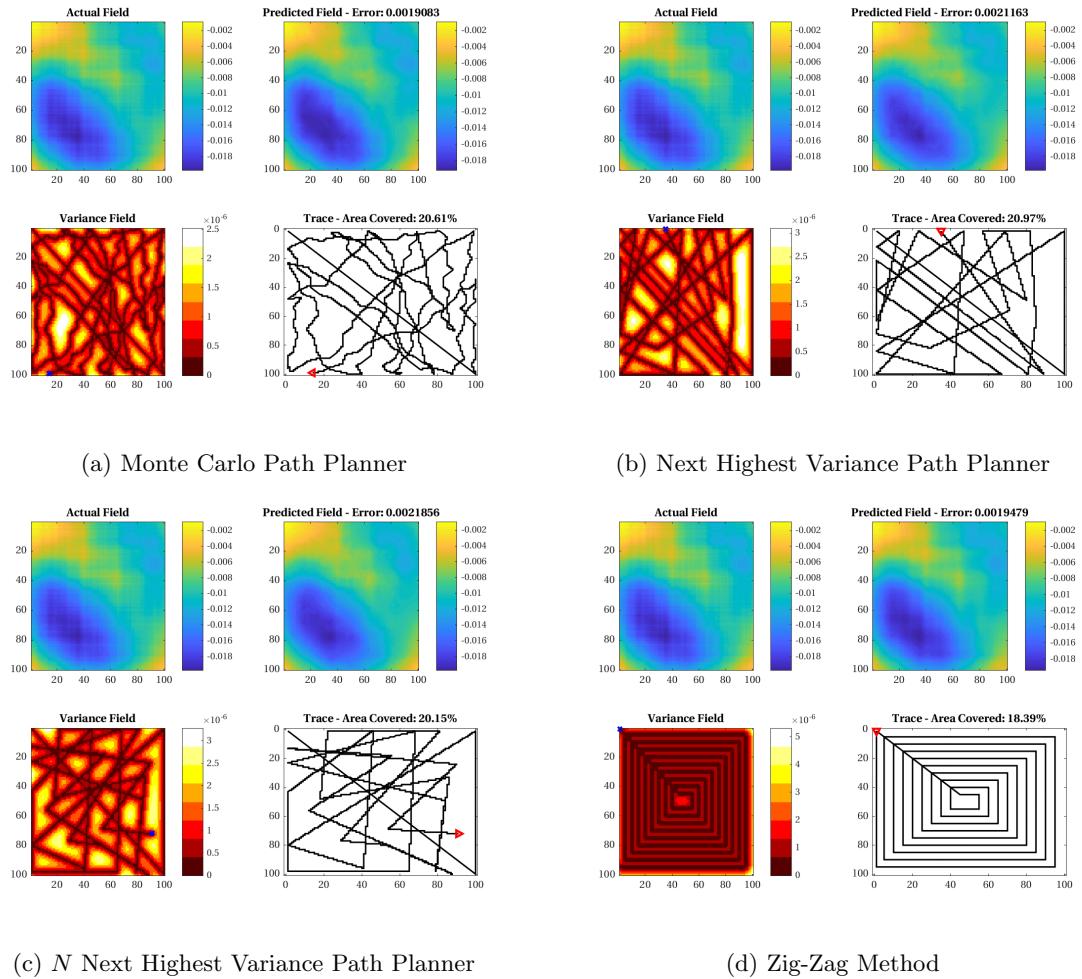
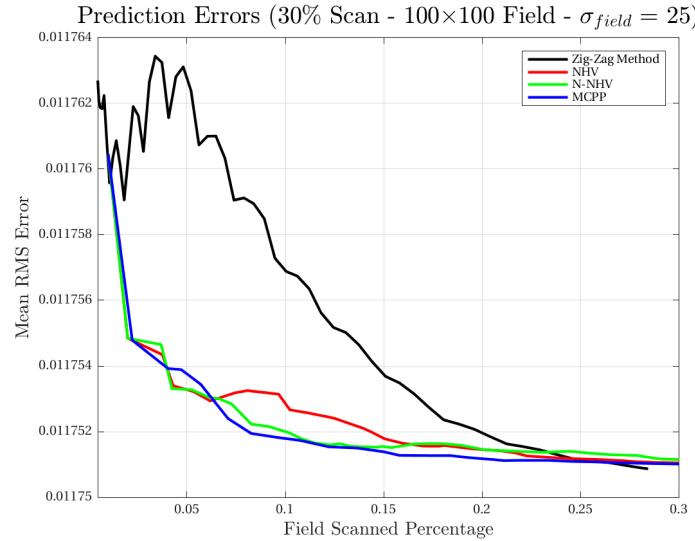


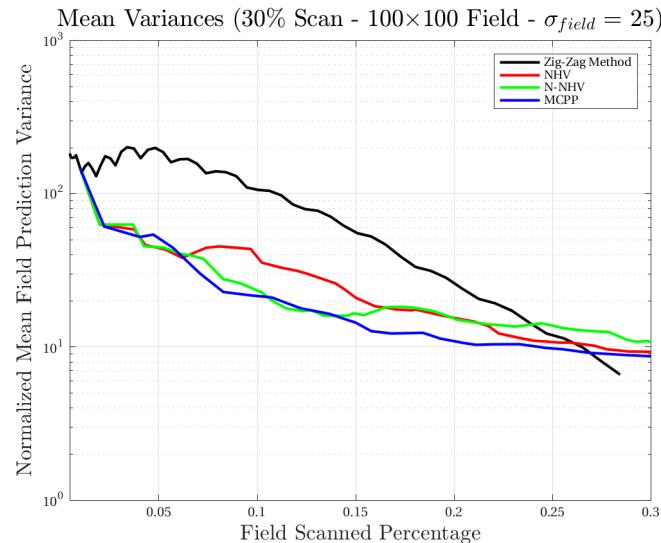
Figure A.16: Simulation output for a 20% maximum area scan on a field of size 100×100 ,

$\sigma_{field} = 25$, random seed: 2.

A.3.3 30% Maximum Field Scan



(a) Prediction errors ($\text{erf}(Z, \hat{Z})$).



(b) Semi-logarithmic prediction variances normalized to an a priori mean variance for the field.

Figure A.17: A 30% maximum area scan on a field of size 100×100 , $\sigma_{field} = 25$, random seed: 2.

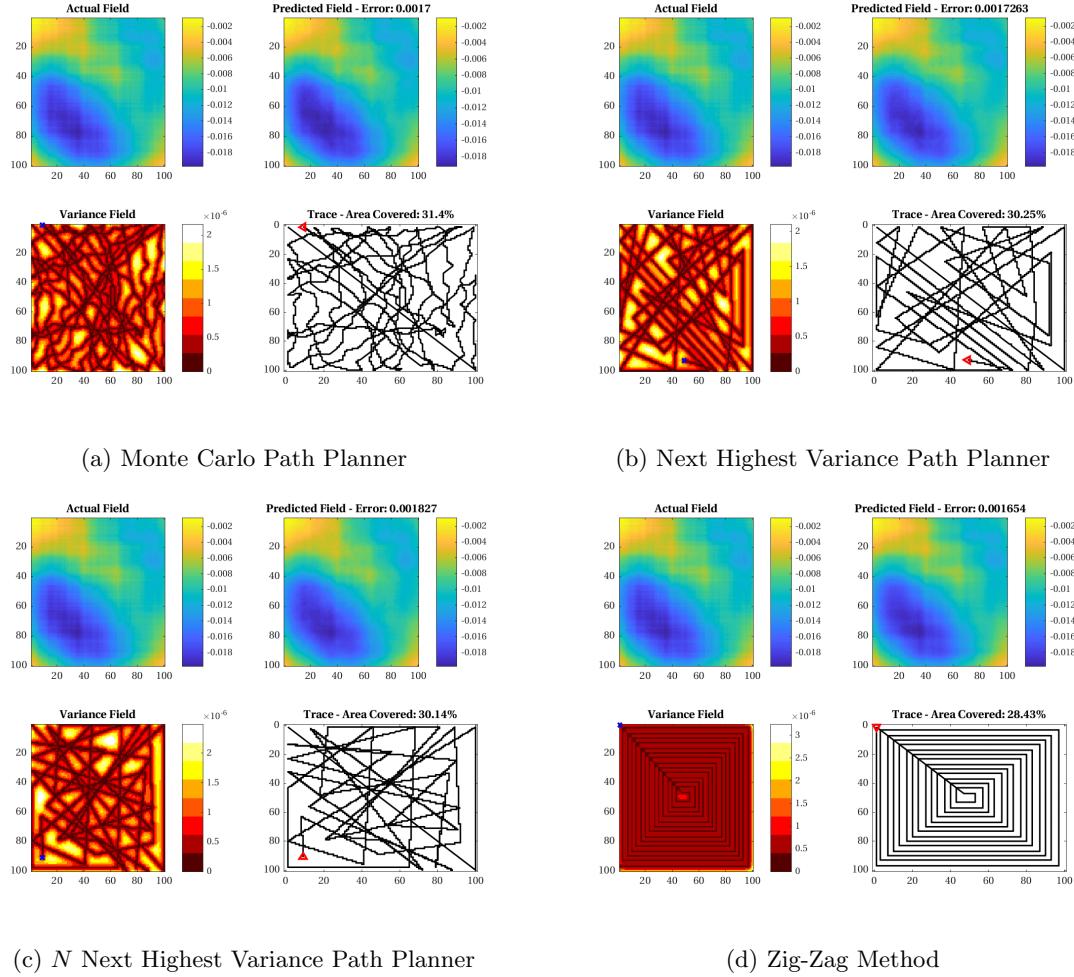
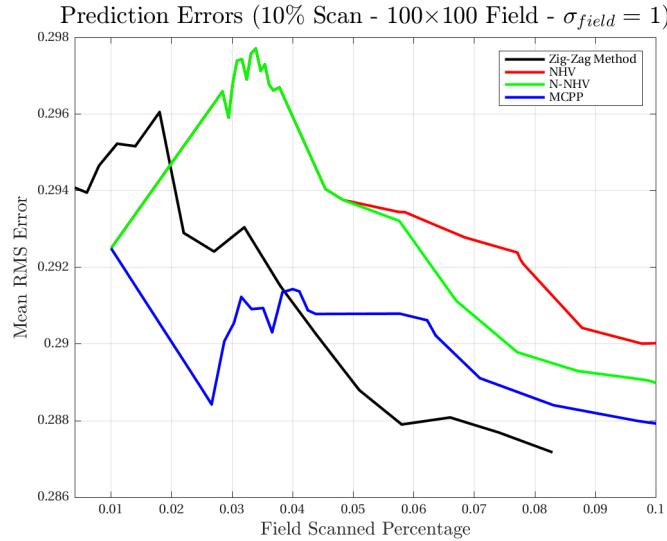


Figure A.18: Simulation output for a 30% maximum area scan on a field of size 100×100 , $\sigma_{field} = 25$, random seed: 2.

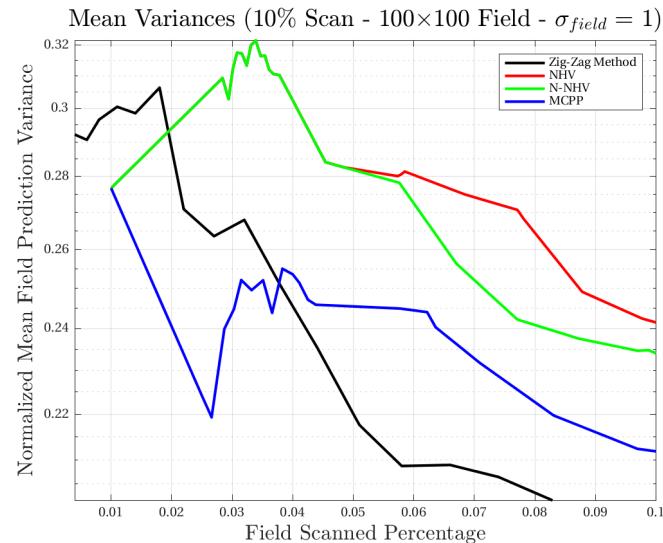
A.4 Low Spatial Autocorrelation ($\sigma_{field} = 1$) Results

The methods will be compared on target fields generated with a low autocorrelation factor.

A.4.1 10% Maximum Field Scan

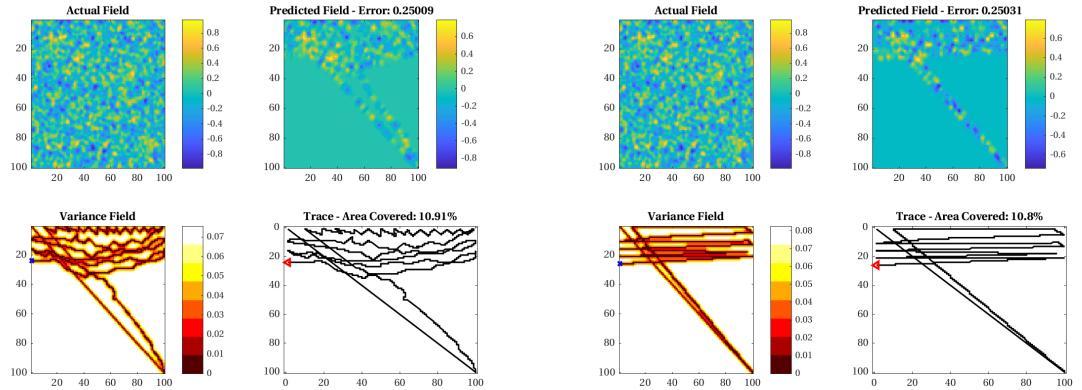


(a) Prediction errors ($\text{erf}(Z, \hat{Z})$).

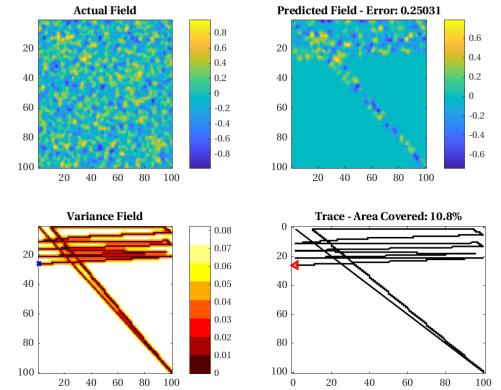


(b) Semi-logarithmic prediction variances normalized to an a priori mean variance for the field.

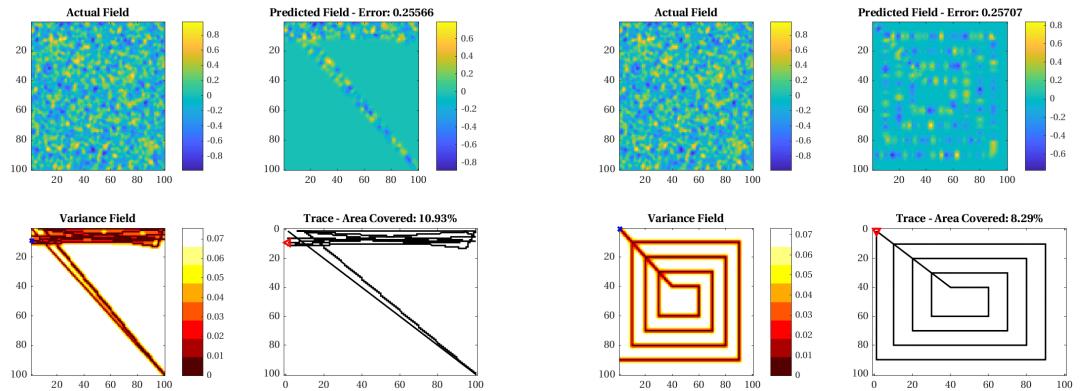
Figure A.19: A 10% maximum area scan on a field of size 100×100 , $\sigma_{field} = 1$, random seed: 2.



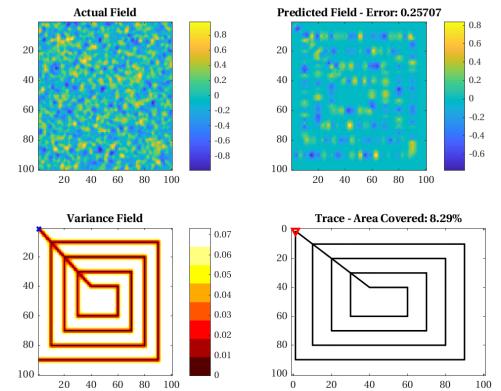
(a) Monte Carlo Path Planner



(b) Next Highest Variance Path Planner



(c) N Next Highest Variance Path Planner

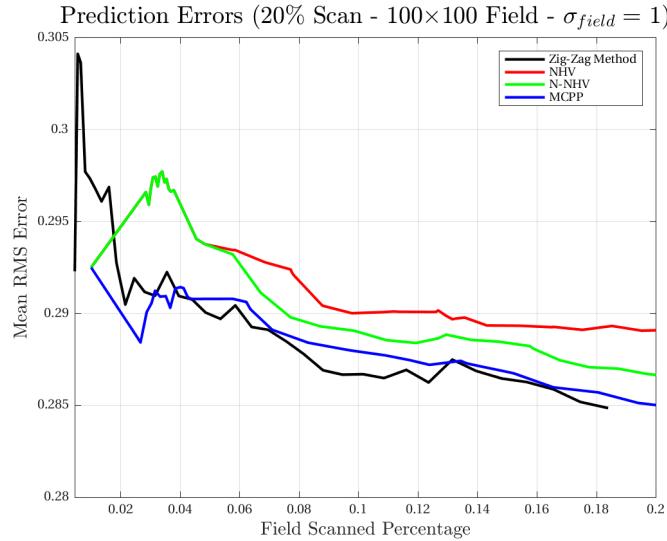


(d) Zig-Zag Method

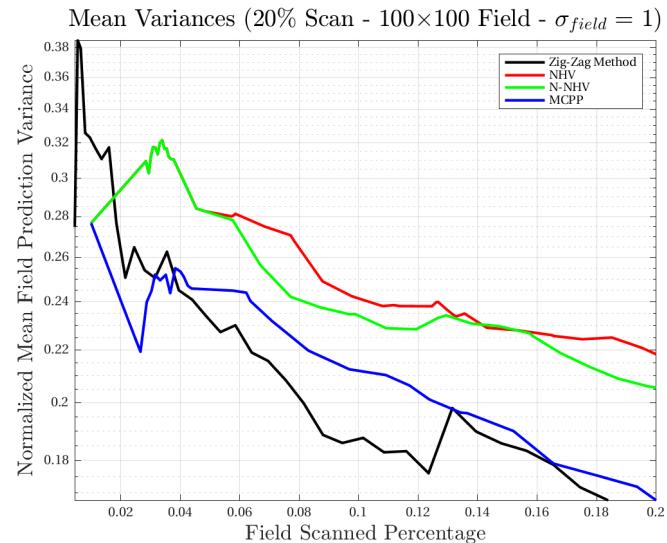
Figure A.20: Simulation output for a 10% maximum area scan on a field of size 100×100 ,

$\sigma_{field} = 1$, random seed: 2.

A.4.2 20% Maximum Field Scan

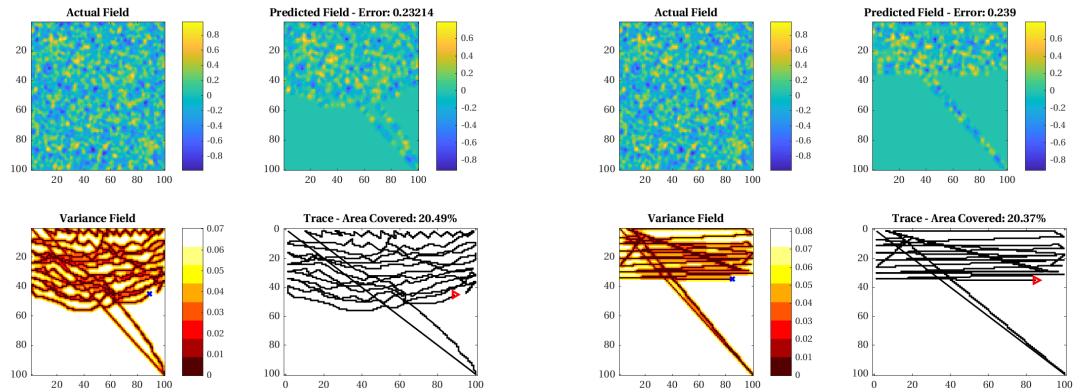


(a) Prediction errors ($\text{erf}(Z, \hat{Z})$).

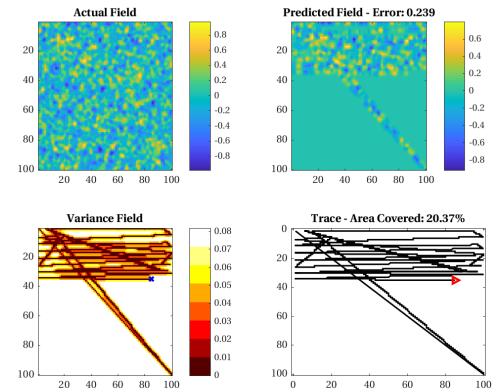


(b) Semi-logarithmic prediction variances normalized to an a priori mean variance for the field.

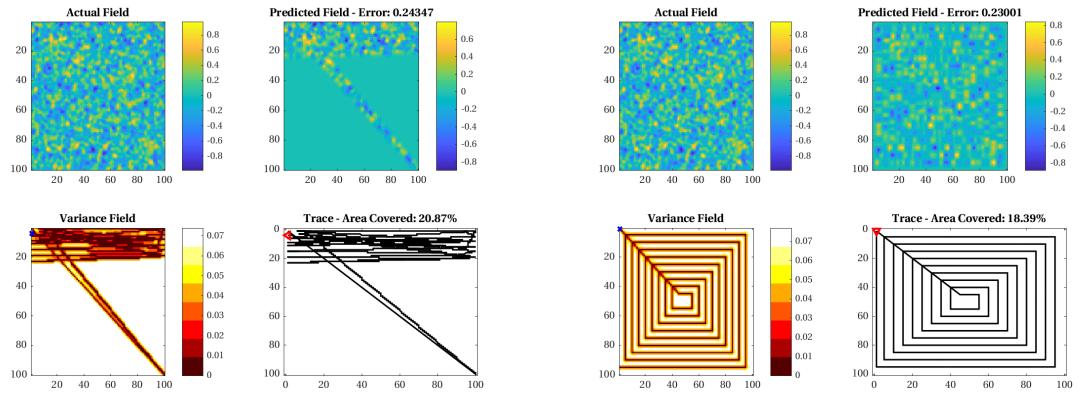
Figure A.21: A 20% maximum area scan on a field of size 100×100 , $\sigma_{\text{field}} = 1$, random seed: 2.



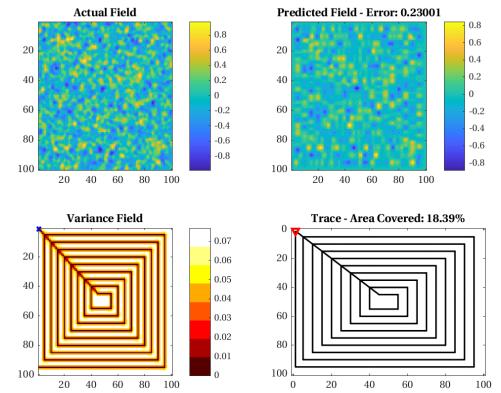
(a) Monte Carlo Path Planner



(b) Next Highest Variance Path Planner



(c) N Next Highest Variance Path Planner

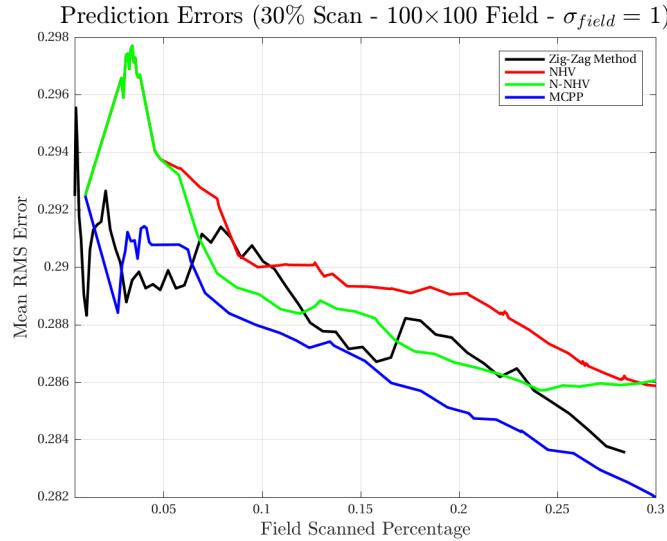


(d) Zig-Zag Method

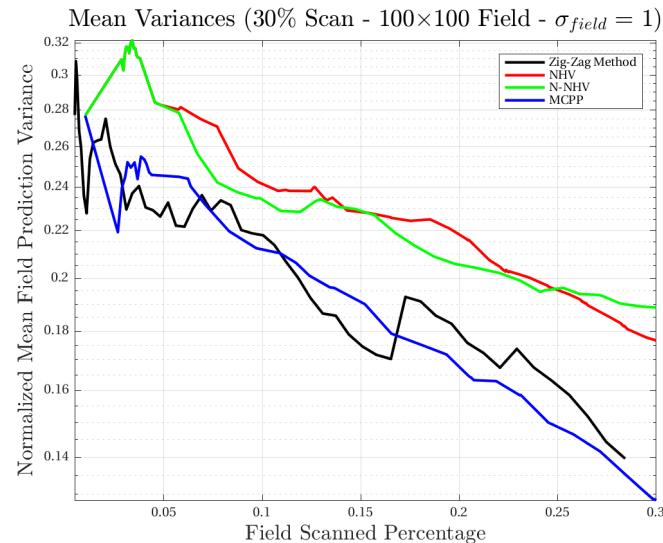
Figure A.22: Simulation output for a 20% maximum area scan on a field of size 100×100 ,

$\sigma_{field} = 1$, random seed: 2.

A.4.3 30% Maximum Field Scan



(a) Prediction errors ($\text{erf}(Z, \hat{Z})$).



(b) Semi-logarithmic prediction variances normalized to an a priori mean variance for the field.

Figure A.23: A 30% maximum area scan on a field of size 100×100 , $\sigma_{\text{field}} = 1$, random seed: 2.

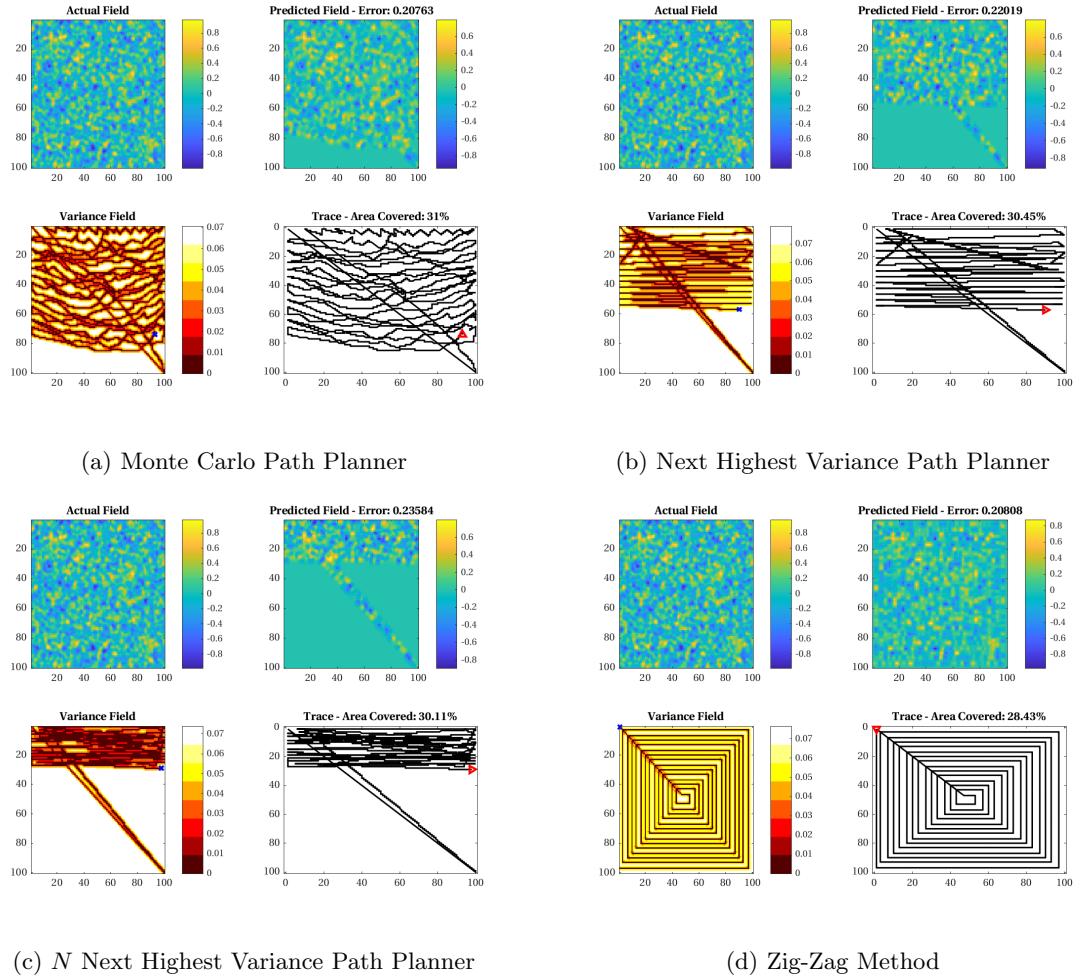


Figure A.24: Simulation output for a 30% maximum area scan on a field of size 100×100 ,

$\sigma_{field} = 1$, random seed: 2.