PREDICTIVE MODELS FOR ROBOTICS AND BIOMEDICAL APPLICATIONS

By

Huan N Do

A Dissertation

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Mechanical Engineering – Doctor of Philosophy

2017

ABSTRACT

PREDICTIVE MODELS FOR ROBOTICS AND BIOMEDICAL APPLICATIONS

By

Huan N Do

Data science has been transforming an enormous number of research areas. It has opened the door to new measures to analyze and extract useful information from raw data. However, while it has been applied extensively in computer science community, there has been a modest number of such applications in the field of robotics and biomedical engineering. In this dissertation, we consider the applications of data analysis and machine learning tools in two research topics: mobile robot localization and cardiovascular predictive models.

In the first part of the dissertation, we tackle a problem of feature selection for appearance-based localization. Raw image is a high-dimensional source of data, and as the resolution of visual sensor has been improved rapidly, we are equipped with even higher dimensional and richer visual information. To deal with the high dimensionality problem, a common and straightforward strategy is to select the most effective visual features for the localization task, i.e., feature selection. In this dissertation, we propose two methods of feature selection. First of all, we model each dimension of the feature vector as a Gaussian process random field with the independent variables as the coordinates of the robot. Thus, the locations of the robot can be inferred by applying a maximum likelihood estimator. The optimal set of features are chosen by backward elimination scheme. Secondly, to minimize the localization error in spatial space and to select the optimal subset of features, we formulate a multivariate version of the Least Absolute Selection and Shrinkage Operator (LASSO) regression model. Under this formulation, we develop a combined localization scheme that consists of the regression and a filtering estimator.

In the later part of this dissertation, we explore the use of predictive models to predict the growth of an abdominal artery under the progression of a disease, Abdominal Aortic Aneurysm (AAA). As a patient who is diagnosed with AAA, his/her artery may locally

be enlarged in pathological conditions and finally ruptures, we develop two prediction approaches using two common types of AAA geometrical data: 3D shapes from computer tomography (CT) scans and 2D profile of maximal diameters over centerline. First of all, we develop our Dynamical Gaussian Process Implicit Surface (DGPIS) for 3D shape prediction. In this method, we consider a 3D surface as a manifold embedded in a scalar field over the 3D dimensional space, the changes of which propagate the changes in the surface. Thus, by utilizing a dynamic model to represent the evolution of the field over time, we can make an inference about the AAA surface in a future time. Secondly, maximal AAA diameter is a crucial criterion for making a surgery intervention decision in clinical practice. Thus, we investigate a Deep Belief Network (DBN) model that is trained on artificial data created from Probabilistic Collocation Method (PCM) and real patients based reconstructed data. Since the merit of DBN and deep structure in general depends on a massive size of training data, which is commonly rare in this application, we overcome the shortage by pre-training the DBN on simulated data generated from PCM, then fine-tuning the neural net on reconstructed data from the real patients. The experimental results illustrate the effectiveness of our proposed methods.

Acknowledgments

First of all, I would like to express my sincere gratitude toward two of my advisers, Dr. Jongeun Choi and Dr. Seungik Baek, who have been not only my professional mentors but also my respected companions along my PhD career. Without their constant encouragement and support, I certainly would not be able to accomplish this dissertation. Additionally, I would like to thank my PhD committee members, Dr. Guoming Zhu and Dr. Pang-Ning Tan, for their suggestions and advices.

Second of all, I would like to thank my collages, Mr. David York, Mr. Alexander Robinson, and Ms. Tam Le, who have been kindly helping me to collect experimental data.

In addition, I would like to appreciate the Vietnam Education Foundation, which has granted me this opportunity to pursue higher education in the US.

Finally, I am grateful to my family: my parents, my wife, and my little daughter. My love is with them forever.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Part 1: appearance-based localization

In recent years, building a complete self-driving car has been a brutal race among a wide range of competitors[1], from information-technology companies such as Google to prolong automobile makers such as General Motors, or start-up-based companies like Tesla. Self-driving is a futuristic technology that with no doubt will be a game changer in terms of not only profitability but also humanity as more human lives can be saved from traffic accidents. However, the advance of self-driving technology, which critically depends on the localization capability, now is hindered by the limitation of GPS acccuracy[2] or being interrupted at GPS-denied regions. Hence, there are strong motivations for developing local navigation technologies that do not solely depend on GPS. Therefore, we develop a localization scheme that is built based on visual sensor measurements, i.e., appearance-based localization. By including the whole dimensions of visual data and providing a complete treatment of associating the feature selection outcomes and the dynamics of the system, our approaches yield excellent tracking performance, which are validated in a number of experimental results.

[1]The list of companies that apply for testing license from California Department of Motor Vehicles: `https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/testing`

[2]The state-of-the-art GPS now has accuracy of $1 \sim 2$ meters, which is obviously inadequate to prevent high speed collision.

Thus, they may be combined with localization with GPS as the information is collected by self-driving vehicles. Furthermore, as GPS is disabled in a certain GPS-denied regions, our algorithms are able to serve as a back-up unit to main the reliable tracking.

1.1.1 Background

In order to make a fast and precise estimation, most of the existing localization algorithms extract a small set of important features from the robotic sensor measurements. The features used in different approaches for robotic localization range from *C.1*: artificial markers such as color tags [56] and barcodes (that need to be installed) [97], *C.2*: geometric features such as straight wall segments and corners [57], and to *C.3*: natural features such as light and color histograms [5]. Most of the landmark-based localization algorithms are classified in *C.1* and *C.2*. It is shown in [94] that autonomous navigation is possible for outdoor environments with the use of a single camera and natural landmarks. In a similar attempt, [23] addressed the challenging problem of indoor place recognition from wearable video recording devices.

The localization methods which rely on artificial markers (or static landmarks) have disadvantages such as lack of flexibility and lack of autonomy. A method is described in [109] that enables robots to learn landmarks for localization. Artificial neural networks are used for extracting landmarks. However, the localization methods which rely on dynamic landmarks [109] have disadvantages such as lack of stability. Furthermore, there are reasons to avoid the geometric model as well, even when a geometric model does exist. Such cases may include: 1) the difficulty of reliably extracting sparse, stable features using geometrical models, 2) the ability to use all sensory data directly rather than a relatively small amount of abstracted discrete information obtained from feature extraction algorithms, and 3) high computational and storage costs of dealing with dense geometric features.

In contrast to the localization problem with artificial markers or popular geometrical models, there is a growing number of practical scenarios in which global statistical information is used instead. Some works illustrate localization using various spatially distributed

(continuous) signals such as distributed wireless Ethernet signal strength [30], or multi-dimensional magnetic fields [113]. In [117], a neural network is used to learn the implicit relationship between the pose displacements of a 6-DOF robot and the observed variations in global descriptors of the image such as geometric moments and Fourier descriptors. In similar studies, gradient orientation histograms [67] and low dimensional representation of the vision data [110] are used to localize mobile robots. In [83], an algorithm is developed for navigating a mobile robot using a visual potential. The visual potential is computed from the image appearance sequence captured by a camera mounted on the robot. A method for recognizing scene categories by comparing the histograms of local features is presented in [70]. Without explicit object models, by using global cues as indirect evidence about the presence of an object, they consistently achieve an improvement over an orderless image representation [70].

1.2 Part 2: predictive models of Abdominal Aortic Aneurysm

Computer-aided diagnosis (CAD) has been studied and applied[3] since the 1960s to help physicians to make decisions faster and more accurate, especially with time pressure. The technology, when being mature, will certainly play pivotal role in disease prognosis and clinical management. CAD offers two main analysis functions that outperform those of humans. Firstly, it provides synthetic information from various sources of data, e.g., CT, MRI or PET scans, which is not obvious to see with information from a single source. Secondly, given measurements for one patient, it can process through an available database to find similar cases and their pathology. These features aim a physician with more information that he/she could obtain by viewing at one image or medical record alone, thus yield better

[3]The market of medical image analysis software is forecasted to reach 4.5 billion USA in 2024: `http://www.grandviewresearch.com/industry-analysis`

informed treatment decisions. One of the aspect of CAD that has been receiving attention from researchers is its predictive capability of biomedical variables that evolve in time. In this study, we particularly consider a disease named Abdominal Aortic Aneurysms (AAAs), which is a local enlargement of the aorta coming from the heart to carry blood to the abdominal body. An undetected aneurysm will keep growing until the point of rupture. Because risks from open surgery or endovascular repair outweigh the risk of AAA rupture, surgical treatments are not recommended with AAAs less than 5.5cm in diameter [85]. Since monitoring the growth of AAA to decide when a surgery is necessary is critical for the treatment of the disease, a predictive model to predict the growth has been heavily investigated.

1.2.1 Background

To predict the temporal growth of the 3D shape of an AAA, there have been research efforts to adopt statistical regression techniques in predicting AAA shape growth. Ijaz et al. [54] has applied a similar approach of spatio-temporal Gaussian process regression to infer the geometrical growth of a parameterized AAA's surface. Additionally, there is a growth and remodeling (G&R) model that uses a Finite Element Method (FEM)-based stress-mediated disease progression [27]. Finally, surface parameterization based methods have been commonly deployed in applications dealing with AAAs [54, 88]. The surface parameterization with cross-sectional contours and a longitudinal line are naturally favored due to the tubular form of an aorta [88].

On the other hand, beside the 3D shape, there are also various biomechanical analyses using other geometrical factors (e.g., different diameter measurements [34,65], tortuosity [84], morphological parameters [91], presence of thrombus [6, 76, 125], influence of the spine [27]) have been proposed to reliably predict the risk of rupture. Among those geometrical factors, maximum diameter is a critical criterion for screening, surveillance and intervention decision making [73]. In particular, a group of authors [34] proposes a method to model AAA's

geometrical shape, given by a series of maximal spheres along the centerline within the lesion. The geometrical model is plotted as a 2D profile curve (diameters versus axial direction) and is used to assess the risk of local rupture of an aorta. Sweeting et al. [107] utilize a multilevel models to make predictions about the future size of aneurysm of individual with longitudinal AAA scanning data. The hierarchical linear growth model utilizes a zero-mean Gaussian distributed random-effects term to simulate the growing effects of aneurysms. Additionally, the linear and quadratic hierarchical growth models have been also heavily used to make predictions about the future size of aneurysms [11, 26].

1.3 Contribution

In this section, we discuss an overview of the distribution of this dissertation in the order of chapters.

In chapter 2, we propose a position estimation method using an omnidirectional camera. We present an approach to build a map from optimally selected visual features using GP regression. First, we describe how we extract some robust properties from vision data captured by an omnidirectional camera. In particular, we describe how different transformations are applied to the panoramic image to calculate a set of image properties. We then transform the high dimensional vision data to a set of uncorrelated feature candidates. A multivariate GP regression with unknown hyperparameters is formulated to connect the set of selected features to their corresponding sampling positions. An empirical Bayes method using a point estimate is used to predict the feature map. Next, a feature reduction approach is developed using the backward sequential elimination method such that an optimal subset of the features is selected to minimize the Root Mean Square Error (RMSE) and compress the feature size. The effectiveness of the proposed algorithms is illustrated by experimental results under indoor and outdoor conditions. Additionally, we compare our results with another appearance-based localization method utilizing the bag of words (BOW) algorithm [9].

In chapter 3, we focus on building a direct mapping from features space to the coordinates instead of solving for locations from the feature map via MLE. Therefore, the test phase is executed in a short period of time since our method does not require any image matching or optimization steps. Secondly, our method is robust to visual noise and does not need high resolution images, which allows it to be applicable in low cost embedded systems. For instance, we use relatively low quality and noise-prone images captured from a regular webcam (Logitech C270, Logitech, Newark, CA, U.S.A.), yet the system yields remarkably accurate results. Furthermore, in contrast to approach in [99], our proposed method does not require depth measurements of the SURF points. Finally, as the dimensionality decreases in the data, the required storage in its database also reduces. Unlike the work in [8] when the quantity of data (i.e., number of images to be stored) is reduced based on removing images that have similar visual features, our method eliminates redundancy in the quality of data. In addition, its application is not limited to localization, but is versatile to other computer vision problems such as object recognition, movement tracking, etc.

In chapter 4, we propose a method to model a dynamically changing 3D shape of an AAA given its longitudinal surface data for its prediction in future time. We briefly discuss our Dynamical Gaussian Process Implicit Surface (DGPIS) model to tackle our problem as follows. Our approach builds on the concept of implicit surfaces [24]. An implicit surface describes the shape of an AAA by a function that maps coordinates of each point in the space to a scalar value (z), which may indicate the point's relative position with respect to the surface of the AAA, i.e., inside ($z < 0$), outside ($z > 0$), or on the surface ($z = 0$). Note that the AAA can have an arbitrarily complex shape. When we consider dynamically changing shape (e.g., AAA), the outcome of the implicit surface function is a time-varying random field over the 3D spatial space. The surface is visible while the field is hidden except the noisy measurements at $z = 0$. In other words, the surface is embedded in the random field. Thus, the changes in the surface are driven by the corresponding changes in the hidden field. We first estimate the hidden field from the point cloud data by utilizing the spatio-temporal

Gaussian process regression as an observation process from the point clouds data. We then further refine the spatio-temporal field by assuming that each point in the field follows linear dynamics in time, key model parameters of which will be estimated via the Expectation-Maximization (E-M) algorithm [101]. Finally, the refined spatio-temporal field allows us to estimate the embedded surface. The refined spatio-temporal random field using linear dynamics provides a way to predict a future shape of the AAA and its growing uncertainty as the prediction time horizon increases otherwise not possible with the Gaussian process regression technique. The model in this study can be viewed as a step towards a Bayesian approach that will be capable of incorporating various uncertainties, patient-specific data, and computational models for aneurysm growth.

In chapter 5, we introduce a deep learning network to the solve the problem of maximum diameter curve prediction of an AAA based on a limited data size collected from real patients. In particular, we provide a complete framework to connect the computational model with a deep belief network. Computational models have been serving as an investigation tool to simulate effects of external structure to the progression of the AAA [28]. Moreover, due to the simplified geometry, the computational models have not been used as (or a part of) a predictive tool. Up to date the computational model and deep structure are developed in two separated and independent paradigms. In this study, the computational model provides deterministic growing trend while while the deep structure can learn the variation in the data.

1.4 Publication

In this section, I would like to present the list of published (as well as will be published) journals articles and conference proceedings that are highly related to this dissertation.

1.4.1 Journal articles

(J1) **Huan N. Do**, J. Choi and S. Baek,"Prediction of maximum diameters of Abdominal Aortic Aneurysms based on Deep Belief Network and Probabilistic Collocation Method", (submission pending).

(J2) **Huan N. Do** and J. Choi, "Appearance-based Localization using Group LASSO regression", *IEEE Transactions on Vehicular Technology*, (submitted).

(J3) **Huan N. Do**, A. Ijaz, H. Gharahi, B. Zambrano, J. Choi, W. Lee and S. Baek, "Prediction of Abdominal Aortic Aneurysms using Gaussian Process Implicit Surfaces", *IEEE Transactions on Biomedical Engineering*, (submitted).

(J4) **Huan N. Do**, M. Jadaliha, J. Choi and C. Y. Lim, "Feature selection for position estimation using an omnidirectional camera", *Image and Vision Computing*, vol. 39, pp. 1-9, 2015.

(J5) **Huan N. Do**, M. Jadaliha, M. Temel and J. Choi, "Fully Bayesian Field SLAM using Gaussian Markov Random Fields", *Asian Journal of Control*, vol. 18, Issue 4, 2015.

1.4.2 Conference proceedings

(C1) **Huan N. Do**, Jongeun Choi, and Seungik Baek. "Prediction of Abdominal Aortic Aneurysm shape evolution using Gaussian process implicit surfaces." Summer Biomechanics, Bioengineering and Biotransport Conference (SB^3C), 2016. July 2016, National Harbor, Maryland, USA.

(C2) **Huan N. Do**, Jongeun Choi, and Chae Young Lim. "Visual feature selection for GP-based localization using an omnidirectional camera." American Control Conference (ACC), IEEE, 2015. (PDF)

(C3) **Huan N. Do**, J. Choi, C. Y. Lim and T. Maiti, "Appearance-based localization using Group LASSO regression with an indoor experiment", IEEE International Conference on Advanced Intelligent Mechatronics (AIM), 2015.

(C4) **Huan N. Do**, J. Choi, C. Y. Lim and T. Maiti, "Appearance-based outdoor localization using group LASSO regression", Dynamics System and Control (DSCC), 2015.

Chapter 2

Feature selection in feature space

Minimizing levels of location uncertainties in sensor networks or robotic sensors is important for regression problems, e.g., prediction of environmental fields [14, 55]. Localization of a robot relative to its environment using vision information (i.e., appearance-based localization) has received extensive attention over past few decades from the robotic and computer vision communities [7, 19]. Vision-based robot positioning may involve two steps. The first step involves learning some properties of vision data (features) with respect to the spatial position where observation is made, so-called mapping. The second step is to find the best match for the new spatial position corresponding to the newly observed features, so-called matching. The mapping from these visual features to the domain of the associated spatial position is highly nonlinear and sensitive to the type of selected features. In most cases, it is very difficult to derive the map analytically. The features shall vary as much as possible over the spatial domain while varying as small as possible for a given position over the disturbance. For example, they should be insensitive to changes in illumination and partial obstruction.

Motivated by the aforementioned situations, we consider the problem of selecting features from the original feature set in order to improve the localization performance of a robot. The central assumption when using a feature selection technique is that the original feature

set contains many redundant or irrelevant features. To facilitate further discussion, let us consider a configuration where the input vector X is the robot position and the output feature vector Y is the collection of extracted features from the vision data. We first build a feature map F at a robot location X such that $F(X) = Y$. In order to reduce position estimation error, the ideal subset is defined as follows.

$$Y_{opt} = \arg\min_{\hat{Y}} \|X - F^{-1}(\hat{Y})\|^2,$$

where $\hat{Y}$ is a vector that consists of the selected entries of the original vector Y. However with a high cardinality of the original feature set, the optimal solution relies on the combinatorial optimization which is not feasible.

One example of the function $F(X)$ can be the mutual information criterion, in which F and F^{-1} could be chosen as follows:

$$F(X) = \arg\max_{Y} I(X,Y), \quad F^{-1}(Y) = \arg\max_{X} I(X,Y),$$

where $I(X,Y) = \int\int \mathbb{P}(X,Y) \log\left(\frac{\mathbb{P}(X,Y)}{\mathbb{P}(X)\mathbb{P}(Y)}\right)$ is the mutual information of X and Y. Note that, in the case where $\mathbb{P}(X)$ and $\mathbb{P}(Y)$ are constant then $F(X)$ obtained by maximizing the log-likelihood function. Guo et al. [39] show by using mutual information, one can achieve a recognition rate higher than 90% while just using 0.61% of feature space for a classification problem. However, the approach based on mutual information could suffer from its computational complexity [86].

The recent research efforts that are closely related to our problem are summarized as follows. The location for a set of image features from new observations is inferred by comparing new features with the calculated map [12, 78, 79]. In [116], a neural network is used to learn the mapping between image features and robot movements. Similarly, there exists effort on automatically finding the transformation that maximizes the mutual information between two random variables [115]. Using Gaussian process (GP) regression, the authors

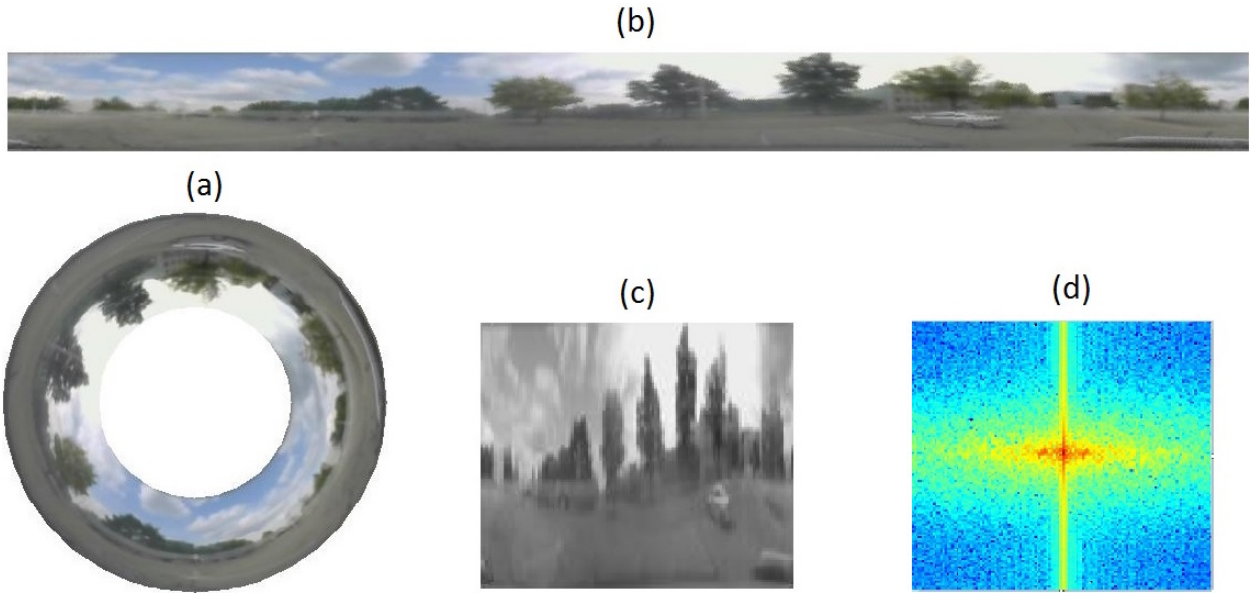


Figure 2.1 (a) and (b) show the wrapped omnidirectional image and the unwrapped panoramic image, respectively. (c) and (d) show the reduced size gray scale image and the two-dimensional FFT magnitude plot, respectively.

of [12,96] present effective approaches to build a map from a sparse set of noisy observations taken from known locations using an omnidirectional camera. While the selection of visual features for such applications determines the ultimate performance of the algorithms, such a topic has not been investigated to date. Therefore, building on Brook's approach [12] our work expands it more on the feature extraction and selection in order to improve the quality of localization. A Bayesian point of view is taken to make the map using a GP framework.

2.1 Image features

Conventional video cameras with projective lens have restricted fields of view. With different mirrors, 360° panoramic views can be achieved in a single image [29]. In this study, to make localization insensitive to the heading angle, an omni-directional camera is used to capture a 360° view from the environment of a robot.

Before an omnidirectional image is processed, it is first unwrapped. When it comes from the camera, the image is a nonlinear mapping of a 360° panoramic view onto a donut shape.

Recovering the panoramic view from the wrapped view requires the reverse mapping of pixels from the wrapped view onto a panoramic view [17, 78]. Figs. 2.1-(a) and 2.1-(b) show the wrapped omnidirectional image and the unwrapped panoramic image, respectively.

We will use the notation $y^{[i]}$ generally for all types of image properties that will be extracted from image i. In particular, we will use the FFT coefficients, the histogram, and the Steerable Pyramid (SP) decomposition [103] as image properties [70]. These feature types and their properties (indicated by $y^{[i]}$) are briefly explained as follows.

FFT (128): The fast Fourier transform (FFT) is applied to the panoramic image to calculate a set of image properties y. For a square image of size $N \times N$, the two-dimensional FFT is given by

$$F^{[i]}(\rho, l) = \sum_{a=0}^{N-1} \sum_{b=0}^{N-1} f^{[i]}(a, b) e^{-j2\pi(\rho \frac{a}{N} + l \frac{b}{N})},$$

where $f^{[i]}$ is the i-th two-dimensional realized image, and j is the imaginary unit. To use FFT, we convert panoramic color images to gray scale 128×128 pixel images, i.e., $[f(a, b)]$. Fig. 2.1-(c) and (d) show the reduced size gray scale image and its two-dimensional FFT magnitude plot, respectively. Often in image processing, only the magnitude of the Fourier transformed image is utilized, as it contains most of the information of the geometric structure of the spatial domain image [81]. Additionally, the magnitude of the Fourier transformed panoramic image is not affected by the rotation in yaw angle. In [78], it was shown that the first 15 components of FFT carry enough information to correctly match a pair of images. We specify the first 64 FFT components of each axis, e.g., $y^{[i]} = \{F^{[i]}(1, 0), \cdots, F^{[i]}(64, 0), F^{[i]}(0, 1), \cdots, F^{[i]}(0, 64)\}$ to be our 128-dimensional image properties of the FFT features.

Histogram (156): The image histogram [40] is a type of a histogram that acts as a graphical representation of the tonal distribution in a digital image. The number of pixels in each tonal bin of the histogram for the image is used as a image property from the

histogram. Thus, the number of different tonal bins, (which is 156) corresponds to the number of image properties from the histogram of the image.

SP (72): The Steerable Pyramid (SP) [103] is a multi-scale wavelet decomposition in which the image is linearly decomposed into scale and orientation subbands, and then the band-pass filters are applied to each subband individually. Using the method from [12], an image is decomposed by 4 scale and 6 orientations, which yields 24 subbands. Each subband is represented by three values, viz., the average filters responses from top, middle, and bottom of the image such that we have 72 image properties for the SP decomposition. The multi-scale wavelet decomposition is also used widely by appearance-based place recognition methods [12, 110].

SURF (64): The Speeded-Up Robust Features (SURF) [45] is a powerful scale- and rotation-invariant that utilizes Haar wavelet responses to produce a 64 dimensional descriptor vector for points of interest in an image. Furthermore, the SURF feature of each point of interest is calculated locally based on the neighborhood region around it.

In general, specific image processing to generate original features will affect the overall performance of the localization. These features are robust to changes in the yaw angle of the vehicle, which results in horizontal shifts of the pixels of the panoramic images. Additionally, images are converted into gray-scale for all types of features since the gray-scale images are less likely to be affected by illumination [59]. The presence of moving objects and occlusions are treated by modeling image features as Gaussian processes via vertical variability and measurement noise, respectively.

2.2 Gaussian process (GP) model

We propose a multivariate GP as a model for the collection of image features. A GP defines a distribution over a space of functions and it is completely specified by its mean function

and covariance function. We denote that $y_\rho^{[i]} := y_\rho\left(s^{[i]}\right) \in \mathbb{R}$ is the i-th realization of the ρ-th image property and $s^{[i]} \in \mathcal{S}$ is the associated position where the realization occurs. Here $\mathcal{S}$ denotes the surveillance region, which is a compact set. Then, the accumulative image properties y is a random vector defined by $y = \left(y_1^T, \cdots, y_m^T\right)^T \in \mathbb{R}^{nm}$, and $y_\rho = \left(y_\rho^{[1]}, \cdots, y_\rho^{[n]}\right) \in \mathbb{R}^n$ contains n realizations of the ρ-th image property.

We assume that the accumulative image properties can be modeled by a multivariate GP, i.e. $y \sim \mathcal{GP}(\Gamma, \Lambda)$, where $\Gamma : \mathcal{S}^n \to \mathbb{R}^{mn}$ and $\Lambda : \mathcal{S}^n \to \mathbb{R}^{mn \times mn}$ are the mean function and the covariance function, respectively. However, the size and multivariate nature of the data lead to computational challenges in implementing the framework.

For models with multivariate output, a common practice is to specify a separable covariance structure for the GP for efficient computation. For example, Higdon [46] calibrated a GP simulator with the high dimensional multivariate output, using principal components to reduce the dimensionality. Following such model reduction techniques, we transform the vector y to a vector z such that its elements $\{z_\rho | \rho \in \Omega_m\}$, where $\Omega_m = \{1, \cdots, m\}$ are i.i.d.

The statistics of y can be computed from the learning data set.

$$\mu_y = \frac{1}{n}\sum_{i=1}^{n} y^{[i]}, \quad \Sigma_y = \frac{1}{n-1}\sum_{i=1}^{n} ||y^{[i]} - \mu_y||^2.$$

The singular value decomposition (SVD) of Σ_y is a factorization of the form $\Sigma_y = USU^T$, where U is a real unitary matrix and S is a rectangular diagonal matrix with nonnegative real numbers on the diagonal. In summary, the transformation will be performed by the following formula.

$$z^{[i]} = S^{-1/2}U^T(y^{[i]} - \mu_y). \tag{2.1}$$

From now on, we assume that we applied the transformation given by (2.1) to the visual data. Hence, we have the zero-mean multivariate GP: $z(s) \sim \mathcal{GP}(0, \mathcal{K}(s, s'))$, which consists of multiple scalar GPs that are independent of each other.

2.2.1 The ρ-th random field

In this subsection, we only consider the ρ-th random field (visual feature). Other scalar random fields can be treated in the same way. A random vector x, which has a multivariate normal distribution of mean vector μ and covariance matrix Σ, is denoted by $x \sim \mathcal{N}(\mu, \Sigma)$. The collection of n realized values of the ρ-th random field is denoted by $z_\rho := (z_\rho^{[1]}, \cdots, z_\rho^{[n]})^T \in \mathbb{R}^n$, where $z_\rho^{[i]} := z_\rho(s^{[i]})$ is the i-th realization of the ρ-th random field and $s^{[i]} = (s_1^{[i]}, s_2^{[i]}) \in \mathcal{S} \subset \mathbb{R}^2$ is the associated position where the realization occurs. We then have $z_\rho(s) \sim \mathcal{N}(0, \Sigma_\rho)$, where $\Sigma_\rho \in \mathbb{R}^{n \times n}$ is the covariance matrix. The i, j-th element of Σ_ρ is defined as $\Sigma_\rho^{[ij]} = \mathbb{C}\text{ov}(z_\rho^{[i]}, z_\rho^{[j]})$. In this dissertation, we consider the squared exponential covariance function [90] defined as

$$\Sigma_\rho^{[ij]} = \sigma_{f,\rho}^2 \exp\left(-\frac{1}{2} \sum_{\ell=1}^{2} \frac{(s_\ell^{[i]} - s_\ell^{[j]})^2}{\sigma_{\ell,\rho}^2} \right). \tag{2.2}$$

In general, the mean and the covariance functions of a GP can be estimated *a priori* by maximizing the likelihood function [123].

The prior distribution of z_ρ is given by $\mathcal{N}(0, \Sigma_\rho)$. A noise corrupted measurement $\tilde{z}_\rho^{[i]}$ at its corresponding location $s^{[i]}$ is defined as follows.

$$\tilde{z}_\rho^{[i]} = z_\rho^{[i]} + \epsilon_\rho^{[i]}, \tag{2.3}$$

where the measurement errors $\{\epsilon_\rho^{[i]}\}$ are assumed to be an independent and identically distributed (i.i.d.) Gaussian white noise, i.e., $\epsilon_\rho^{[i]} \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_{\epsilon,\rho}^2)$. Thus, we have that

$$\tilde{z}_\rho \sim \mathcal{N}(0, R_\rho),$$

where $R_\rho = \left(\Sigma_\rho + \sigma_{\epsilon,\rho}^2 I\right)$. The log-likelihood function is defined by

$$L_{\theta,\rho} = -\frac{1}{2} \tilde{z}_\rho^T R_\rho^{-1} \tilde{z}_\rho - \frac{1}{2} \log\det R_\rho - \frac{n}{2} \log 2\pi, \tag{2.4}$$

where n is the size of $\tilde{z}_\rho$.

The hyperparameter vector of the ρ-th random field is defined as $\theta_\rho = (\sigma_{f,\rho}, \sigma_{\epsilon,\rho}, \sigma_{1,\rho}, \sigma_{2,\rho}) \in \mathbb{R}^4_{>0}$. Using the likelihood function in (2.4) the hyperparameter vector can be computed by the ML estimator

$$\bar{\theta}_\rho = \arg\max_\theta L_{\theta,\rho}, \tag{2.5}$$

which will be plugged in prediction as in an empirical Bayes way.

All parameters are learned simultaneously. If no prior information is given, then the maximum a posteriori probability (MAP) estimator is equal to the ML estimator [123].

In a GP, every finite collection of random variables has a multivariate normal distribution. Consider a realized value of the ρ-th random field $z^\star_\rho$ being taken from the associated location $s^\star$. The probability distribution $\mathbb{P}(z^\star_\rho | s^\star, s, \tilde{z}_\rho)$ is a normal distribution with the following mean and variance.

$$\mu_\rho(s^\star) = C_\rho^T R_\rho^{-1} \tilde{z}_\rho, \quad \sigma^2_\rho(s^\star) = \sigma^2_{f,\rho} - C_\rho^T R_\rho^{-1} C_\rho, \tag{2.6}$$

where the covariance $C_\rho := \mathbb{C}\text{ov}(z^\star_\rho, z_\rho) \in \mathbb{R}^{1\times n}$ is defined similar to (2.2).

In order to estimate location $s^\star$, using the MAP estimator, we need to compute $\mathbb{P}(s^\star | \tilde{z}^\star_\rho, s, \tilde{z}_\rho)$, where the noisy observation $\tilde{z}^\star_\rho$ is the summation of the realized values of the random field $z^\star_\rho$ and a noise process.

$$\mathbb{P}(s^\star | \tilde{z}^\star_\rho, s, \tilde{z}_\rho) = \frac{\mathbb{P}(\tilde{z}^\star_\rho | s^\star, s, \tilde{z}_\rho)\mathbb{P}(s^\star | s, \tilde{z}_\rho)}{\mathbb{P}(\tilde{z}^\star_\rho | s, \tilde{z}_\rho)}. \tag{2.7}$$

A MAP estimator given the collection of observations $\tilde{z}_\rho$ is a mode of the posterior distribution.

$$\bar{s}^\star_\rho = \arg\max_{s^\star \in \mathcal{S}} \mathbb{P}(s^\star | \tilde{z}^\star_\rho, s, \tilde{z}_\rho). \tag{2.8}$$

If $\mathbb{P}(s^\star | s, \tilde{z}_\rho)$ and $\mathbb{P}(\tilde{z}^\star_\rho | s, \tilde{z}_\rho)$ are uniform probabilities, then the MAP estimator is equal

to the ML estimator, given by

$$\bar{s}_{\rho}^{\star} = \arg\max_{s^{\star}\in\mathcal{S}} L_{\rho}(s^{\star}), \tag{2.9}$$

where the ρ-th log-likelihood function, i.e., $L_{\rho}(s^{\star})$, is defined as follows.

$$L_{\rho}(s^{\star}) = \frac{-1}{2}\left(\frac{|\tilde{z}_{\rho}^{\star} - \mu_{\rho}(s^{\star})|^2}{\sigma_{\epsilon,\rho}^2 + \sigma_{\rho}^2(s^{\star})} + \log\left(\sigma_{\epsilon,\rho}^2 + \sigma_{\rho}^2(s^{\star})\right) + \log 2\pi\right). \tag{2.10}$$

2.3 Localization and feature selection

Let Ω be the collection of indices that are associated to the multiple scalar random fields (of the multivariate GP). Provided that all scalar random fields (of the multivariate GP) are independent of each other, we then obtain a computationally efficient ML estimate of the location given the observations of all scalar random fields $\{\tilde{z}_{\rho}|\rho\in\Omega\}$ as follows.

$$\bar{s}_{\Omega}^{\star} = \arg\max_{s^{\star}\in\mathcal{S}} \sum_{\rho\in\Omega} L_{\rho}(s^{\star}), \tag{2.11}$$

where $L_{\rho}(s^{\star})$ is the ρ-th log-likelihood function as given in (2.10).

In this dissertation, a backward sequential elimination technique [61] is used for the model selection. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. To this end, we divide the data set into two segments: one used to learn or train the GP model and the other used to validate the model.

The RMSE is used to measure the performance of GP models. It is defined by the following equation.

$$\mathrm{RMSE}(\Omega) = \sqrt{\frac{1}{n_c}\sum_{i=1}^{n_c}\left\|s_c^{[i]} - \bar{s}_{\Omega}^{\star}\right\|^2}, \tag{2.12}$$

where $\|\cdot\|$ is the Euclidean norm of a vector. In the case that $\Omega = \emptyset$, we define the following.

$$\mathrm{RMSE}(\emptyset) = \sqrt{\frac{1}{n_c}\sum_{i=1}^{n_c}\left\|s_c^{[i]} - \mathrm{median}(s_c)\right\|^2},$$

where median$(\cdot)$ is the median of a random vector. Assume that $\Omega_m = \{1, \cdots, m\}$ is the set of all features. Dupuis et al. [25] reported that the backward sequential elimination outperforms the forward sequential selection. Thus, we use a backward sequential elimination algorithm as follows.

$$\Omega_{\ell-1} = \Omega_\ell - \arg\min_{\rho\in\Omega_\ell} \mathrm{RMSE}(\Omega_\ell - \rho), \forall \ell \in \Omega_m, \tag{2.13}$$

where $\Omega_\ell - \rho = \{p | p \in \Omega_\ell, p \neq \rho\}$.

Finally a subset of features is selected as follows.

$$\Omega_{opt} = \arg\min_{\Omega=\Omega_1,\cdots,\Omega_m} \mathrm{RMSE}(\Omega). \tag{2.14}$$

The optimum subset Ω_{opt} has the minimum RMSE among $\{\Omega_1, \cdots, \Omega_m\}$. The mapping and matching steps of the proposed approaches in this disesrtation are summarized in Algorithm 1 and Algorithm 2, respectively.

2.4 Indoor and outdoor experiments

In this section, we demonstrate the effectiveness of the proposed localization algorithms with experiments using different image features we discussed. We report results on two different data sets collected indoors (Case 1) and outdoors (Case 2).

Algorithm 1 learning maps from a sparse set of panoramic images observed in known locations

Input: #1. training data set includes a set of panoramic images captured from known spatial sites,

Output: #1. a linear transformation from image properties to uncorrelated visual features,
#2. the estimated hyperparameter, the estimated mean and the estimated variance function of each independent visual feature,

1: extract image properties $y^{[i]}$ in the available learning data set.
2: use SVD to make a set of uncorrelated visual features $z^{[i]}$ using (2.1)
3: for each independent visual feature estimate hyperparameters using (2.5)
4: compute the mean function and variance function for each of independent features using (2.6)
5: choose optimal subset of visual features using (2.14) to eliminate some of the visual features that are worthless for the localization goal.

Algorithm 2 localization predictive inference using learned map of visual features.

Input: #1. a linear transformation from image properties to uncorrelated visual features,
#2. the estimated hyperparameter and the estimated mean and variance function of selected visual features,

Output: #1. position of newly captured images.

1: capture new images and obtain image properties $y^{\star}$.
2: compute the selected visual features $z^{\star}$ using (2.1)
3: compute the likelihood function of selected features $\rho \in \Omega_{opt}$ over possible sampling positions using (2.10)
4: determine the estimated position $\vec{s}^{\star}_{\Omega_{opt}}$ using (2.11)

2.4.1 Experimental setups

In Case 1, the Kogeto panorama lens was used to capture 360° images. In total, 207 pairs of exact sampling positions were recorded manually and corresponding captured panoramic images on a regular lattice (7×2.7 m^2) were collected.

In Case 2, we use a vision and GPS data acquisition circuit which consists of an Arduino microcontroller (Arduino MEGA board, Open Source Hardware platform, Italy), a Xsens GPS unit (MTi-G-700, Xsens Technologies B.V., Netherlands), a Raspberry Pi microcontroller (Raspberry Pi model B+, Raspberry Pi Foundation, United Kingdom) and a webcam (Logitech HD webcam C310, Logitech, Newark, CA, U.S.A.) glued to a 360° lens (Kogeto Panoramic Dot Optic Lens, Kogeto, U.S.A.). The data acquisition circuit was secured inside

Figure 2.2 Outdoor trajectory collected from a GPS unit.

the vehicle while the omni-directional camera was fixed on the roof of the vehicle. The vehicle was driven through the surveillance area (Fig. 2.2). The surrounding scenes were recorded by the Raspberry Pi unit while the truth locations measured by the Xsens GPS unit were stored on the Arduino microcontroller. We collected 378 data points, on a 61×86 meter area on the campus of Michigan State University, East Lansing, MI, U.S.A (see Fig. 2.2). Figs. 2.3 and Fig. 2.4 shows the setups for Case 1 and Case 2, respectively.

The data sets are divided into 50% *learning*, 25% *backward sequential elimination (or validation)* and 25% *testing* data subsets. The learning data set is used to estimate the mean functions and the hyperparameters for the covariance functions to build GP models. The validation data set is used to select the best features in order to minimize the localization

Figure 2.3 Dot iPhone Panorama lens (left) and the indoor environment (right) for Case 1.

estimation RMSE and compress the feature. After the training and feature selection, we evaluate the performance of the selected model using the testing data set, which was not used for training or feature selection.

To analyze our results in a statistically meaningful way, we calculate the Bayesian Information Criteria (BIC) index for the model with all features and the one with only selected features in addition to the RMSE. The BIC is a criterion for model selection based on the log likelihood with a penalty on the number of parameters to penalize over-fitting. The model with a smaller BIC index is less likely to be over-fitted [64].

2.4.2 Learning of GP models in an empirical Bayes approach

As illustrative examples for the case of utilizing FFT features of length 128, we apply the proposed algorithm to both data sets. The variance of the random field σ_f^2, the spatial bandwidth $\sigma_{\ell,\rho}^2$, and the noise variance σ_ϵ^2 are estimated for each feature independently. Thus, for the FFT case, $128 \times 4 = 512$ hyperparameters need to be estimated in total for each experimental setup. The hyperparameters for Case 2 are estimated in the same manner.

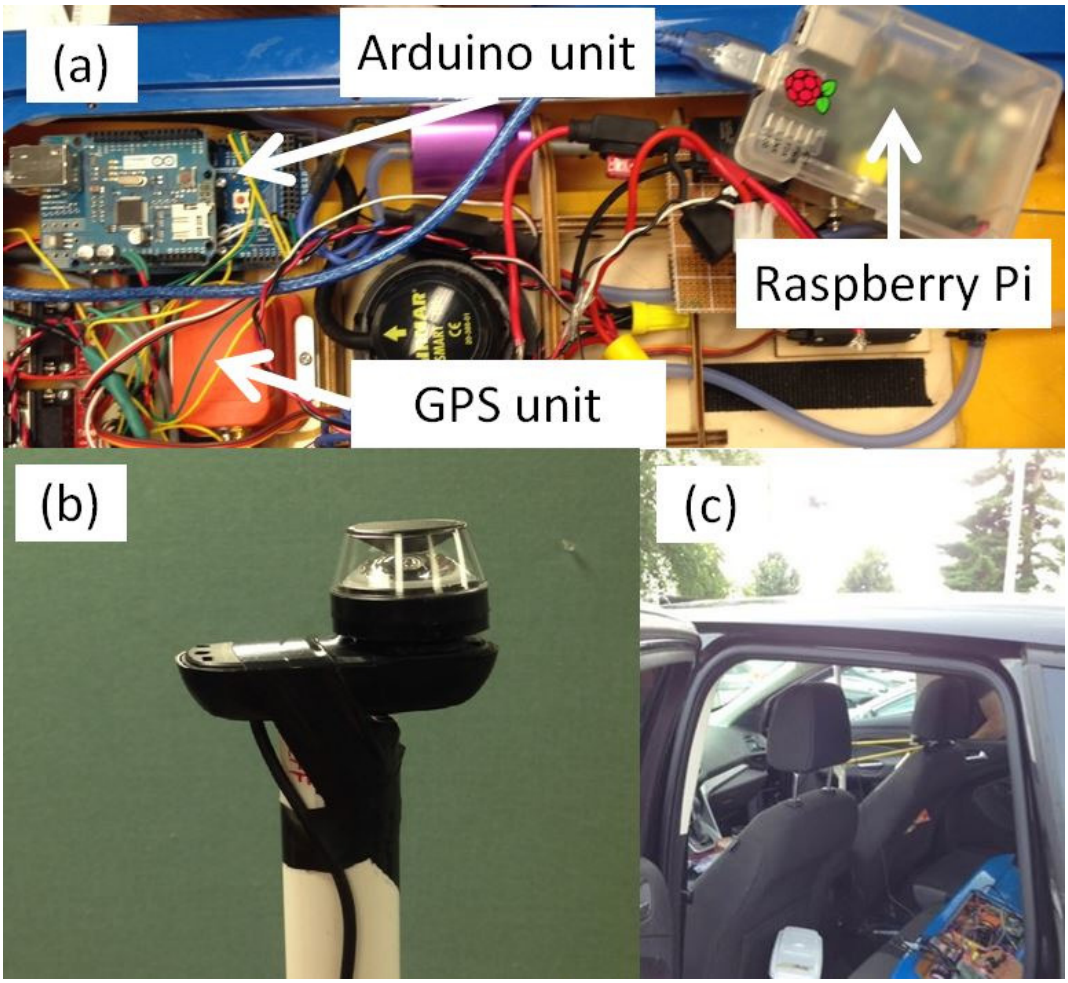


Figure 2.4 (a) Data acquisition circuit, (b) panoramic camera and (c) vehicle used in Case 2.

The 3D plots of the means and variances of the first three GP models for the case of 128 FFT features are shown in Fig. 2.5.

To study the effect of the turning angle of the vehicle (or the yaw angle) on the features, we run the algorithm with another data set in which the collected panoramic images are pre-processed so that the heading of the panoramic image is kept constant using the yaw angles from the GPS unit, denoted as (fixed angle) in Table 2.1.

All inferential algorithms are implemented using Matlab R2013a (The MathWorks Inc., Natick, MA, U.S.A.) on a PC (3.2 GHz Intel i7 Processor).

2.4.3 Localization utilizing the Bag of Words (BOW)

We also compare the GP-based approach with a localization scheme based on the BOW. To have a fair comparison, we feed an identical data set to both of the methods. We utilize the SURF as the image descriptor for our BOW. We define the notation $y^{[i]}$ as a set of SURF points extracted from image i. Notice that the number of SURF points varies for different images. The region around each SURF point is represented by a descriptor vector of 64 length. The SURF points from the whole data set are accumulated and put into the k-means clustering [41]. Each centroid is defined as a codeword and the collection of centroids is defined as the codebook. Each SURF point is mapped into the index of the nearest centroid in the codebook. Therefore, we obtain a histogram of codewords for each image that indicates the appearance frequency of all codewords in the image. Lastly, the test set is classified by applying the k-nearest-neighbor classifier [60] based on the histogram of codewords.

We subsample 25% of the data to be the test set (the same test subset used in Table 2.1), which is associated with a newly defined label set, i.e., $\mathcal{I}_T := \{1, \cdots, n_T\}$. The label of each test data point s^* is assigned to the non-test data points within a 5 meter radius with respect to s^* (see Fig. 2.6). Such relabeled non-test data points are used for training the BOW. Since the BOW is mostly used for classification such as identical scenes recognition [9], we define

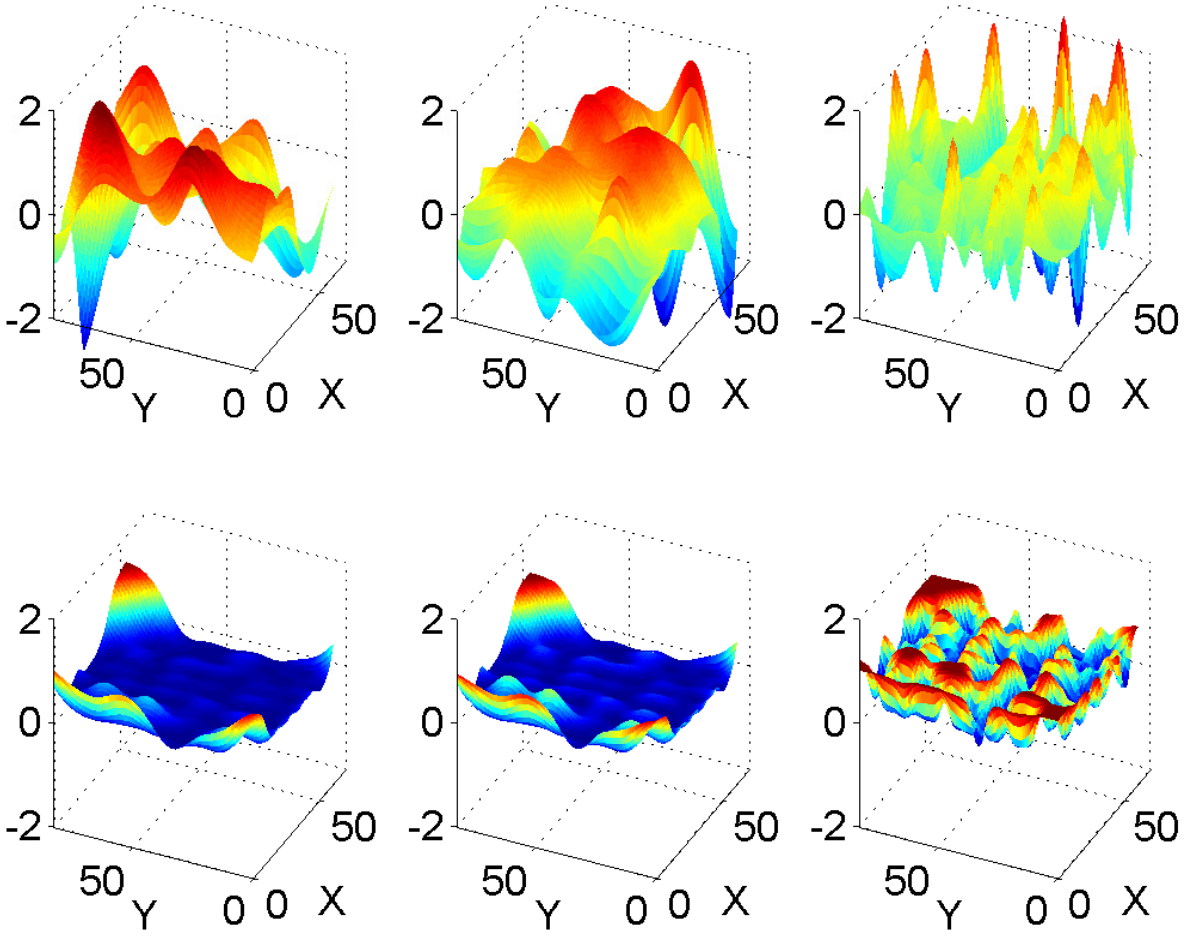


Figure 2.5 GP models for each of first three FFT features from the outdoor data set. The first row shows the means and the second row shows the variances of the GP models.

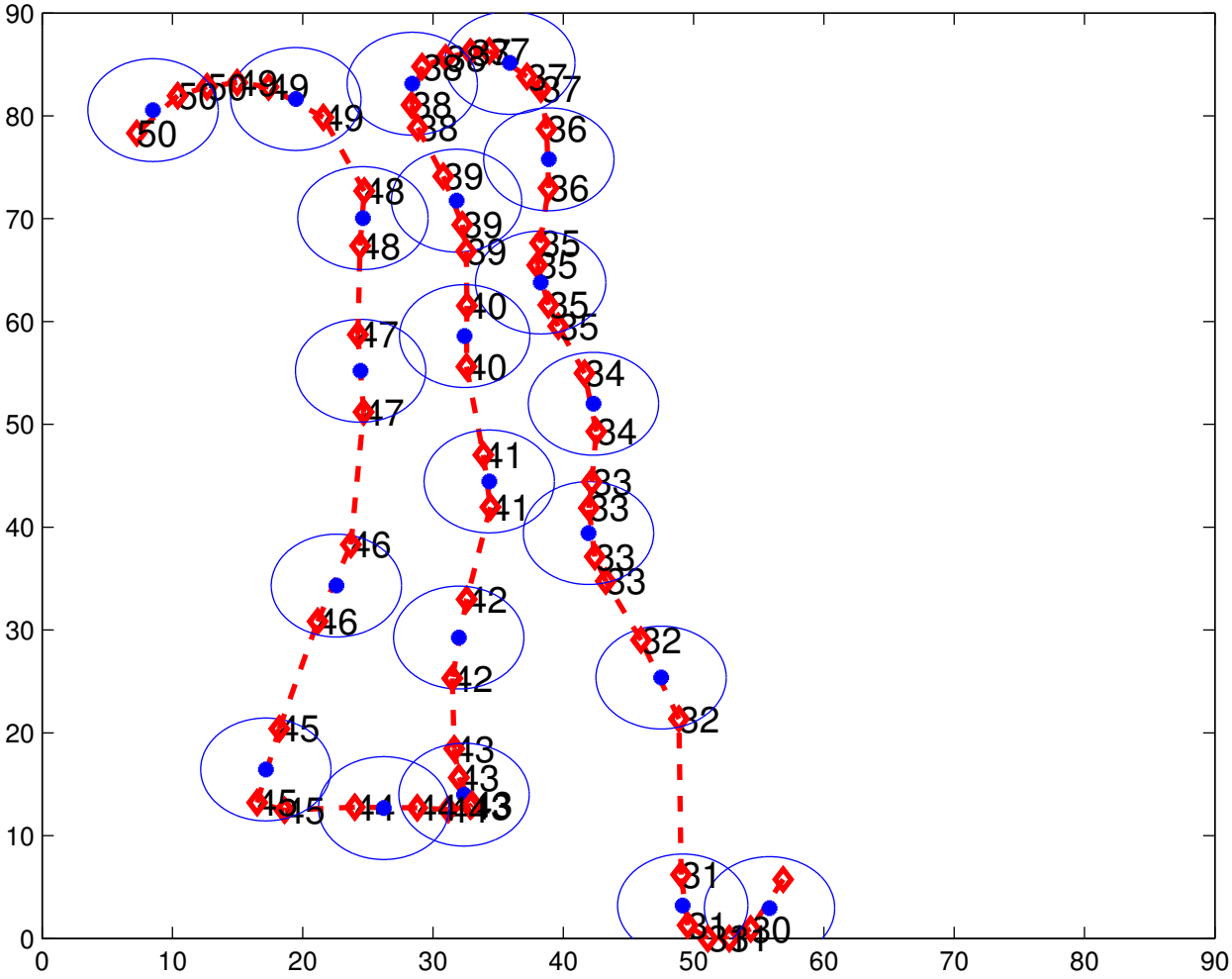


Figure 2.6 Training data set assignment for the BOW. The test points, the training points (with new labels), and the 5m radii are plotted in blue dots, red diamonds, and blue circles, respectively. The training points that do not belong to any test groups are eliminated.

the localization error to compute the RMSE as follows.

Let $s_t(i) \in \mathcal{S}$ be the location of the test point i for all $i \in \mathcal{I}_T$. Let $h^\star(i)$ be the predicted label for the test data point i. Then we define the error at test point i as follows.

$$\text{error}_i = \begin{cases} \|s_t(i) - s_t(h^\star(i))\| & \text{if } i \neq h^\star(i), \\ 0 & \text{if } i = h^\star(i), \end{cases} \tag{2.15}$$

for all $i \in \mathcal{I}_T$.

2.4.4 Experimental results

Our method over different features: The indoor and outdoor performances under different image features are summarized in Table 2.1 and Table 2.2, respectively. We consider three different types of appearance-based features such as the FFT [117], the histogram [40], and the SP decomposition [103]. We calculate the RMSE of our localization estimation from the model on the same validation set and on a separated test set, denoted by "V" and "T" in the tables, respectively. To compare the reduction in the number of features, we use the compression ratio [95]. The compression ratio is defined as:

$$\text{Compression ratio} := \frac{\text{number of original features}}{\text{number of selected features}}.$$

Table 2.1 and Table 2.2 show the appearance-based feature type (column 1), the total number of features (column 2), the optimum number of features along with the compression ratio on the validation set (column 3), the localization RMSE obtained using the total number of features (column 4), and the localization RMSE from the optimum number of features selected by the backward sequential elimination (denoted as "BE") implemented on the test set (column 5), and the localization RMSE from validation set (column 6), the maximum localization error, i.e., $e_{\max}$ taken over the test set (column 7), the BIC indexes of the model using the total number of features (column 8), and the BIC indexes from the optimum number of features (column 9). For Case 2, the FFT and SP features are tested with the data in two situations when the yaw angles are varying and when they are fixed (denoted as (fixed) in Table 2.2) to gauge the effect of yaw angles.

Performance among features: For Case 1, the SP shows the lowest localization RMSE with 4.8 compression ratio. For Case 2, the histogram shows the lowest localization RMSE with 2.7 compression ratio. The predicted trajectory that utilizes the histogram for Case 2 is shown in Fig. 2.7. For all experiments, BIC indexes before and after the feature selection

feature type	# of features		RMSE (meters)			$e_{\max}$ (meters)	BIC index	
	total	opt	All	BE			All	opt
		V	T	T	V	T	$\times 10^3$	$\times 10^3$
FFT	128	77 (1.7)	1.48	1.68	0.94	7.2	16.2	9.5
FFT (noisy)	128	20 (6.4)	1.78	1.89	0.61	7.1	16.7	2.2
Hist	156	9 (17.3)	1.69	1.71	0.55	7.1	18.5	2.2
Hist (noisy)	156	50 (3.1)	1.58	1.64	1.14	6.0	20.5	6.4
SP	72	15 (4.8)	0.67	0.85	0.27	3.4	22.2	4.1
SP (noisy)	72	62 (1.2)	1.63	1.45	0.59	5.8	9.1	7.9

Table 2.1 The localization performance for Case 1

feature type	# of features		RMSE (meters)			$e_{\max}$ (meters)	BIC index	
	total	opt	All	BE			All	opt
		V	T	T	V	T	$\times 10^3$	$\times 10^3$
FFT	128	41 (3.1)	14.5	10.9	4.3	58.5	28.4	8.1
FFT (fixed)	128	25 (5.12)	13.7	13.6	4.8	56.7	28.4	4.7
Hist	156	58 (2.7)	7.5	6.9	3.7	42.4	33.8	11.4
SP	72	35 (2.1)	21.08	18.72	7.86	63.4	15.6	7.1
SP (fixed)	72	28 (2.6)	14.52	14.74	13.19	51.9	15.6	5.5

Table 2.2 The localization performance for Case 2

Feature type	Number of features			RMSE (meters)			
	total	Optimum		GP		BOW	
		unfixed	fixed	unfixed	fixed	unfixed	fixed
FFT	128	41	25	10.97	13.68	-	-
Histogram	156	58	-	6.91	-	-	-
SP	72	35	28	18.72	14.74	-	-
SURF	-	-	-	-	-	23.15	22.07

Table 2.3 The localization performance comparison between the proposed approach and the BOW in Case 2. The RMSE is from the test set.

show the significant improvement that makes the selected model less likely susceptible to over-fitting. From the RMSE on the test data set, all feature types seem to be robust to the varying yaw angles.

Effect of localization noise: To investigate the effect of noisy sampling positions on the method, we added fictitious localization noise generated by a Gaussian white noise process with standard deviation of 0.3048 meters (i.e., 1 foot) to the sampled locations of Case 1, which is denoted by (noisy) in Table. 2.1. As expected, the results show degradation when noisy sampling positions are used due to the sampling uncertainty in the GP learning and prediction processes.

Comparison of Cases 1 and 2: Note that sampling positions of Case 1 (non-noisy data) were recorded exactly while those of Case 2 were noisy due to the uncertainty in the GPS unit. On the other hand, it is clear that Case 2 has the larger RMSE due to the larger scale of the surveillance site compared to Case 1. Together, the results of Case 1 are shown to outperform those of Case 2.

Comparison with the BOW: We compare the performance between our proposed GP-based approach and the BOW in Table 2.3. Table 2.3 shows the feature types (column 1), the total number of features (column 2), the optimum selected number of features from the angle-varying data set (column 3) and the fixed angle data set (column 4), the localization RMSE of the GP-based method from the angle-varying data set (column 5) and the fixed angle data set (column 6), the localization RMSE of the BOW from the angle-varying (column 7) and the fixed angle (column 8) data sets. Since the performance of the BOW highly depends on the clustering results, we run the BOW with different sizes of clusters, and the one that yields the highest classification percentage (80-90%) is chosen to calculate the RMSE using the error defined in (2.15). As discussed, the change in yaw angle does not

show significant effect on the SURF. Table 2.3 shows that our approach outperforms the BOW.

In summary, we achieve significant reduction in the number of features while improving the RMSE when applied to the validation set. Furthermore, we maintain approximately the same RMSE levels when applied to a new test data set.

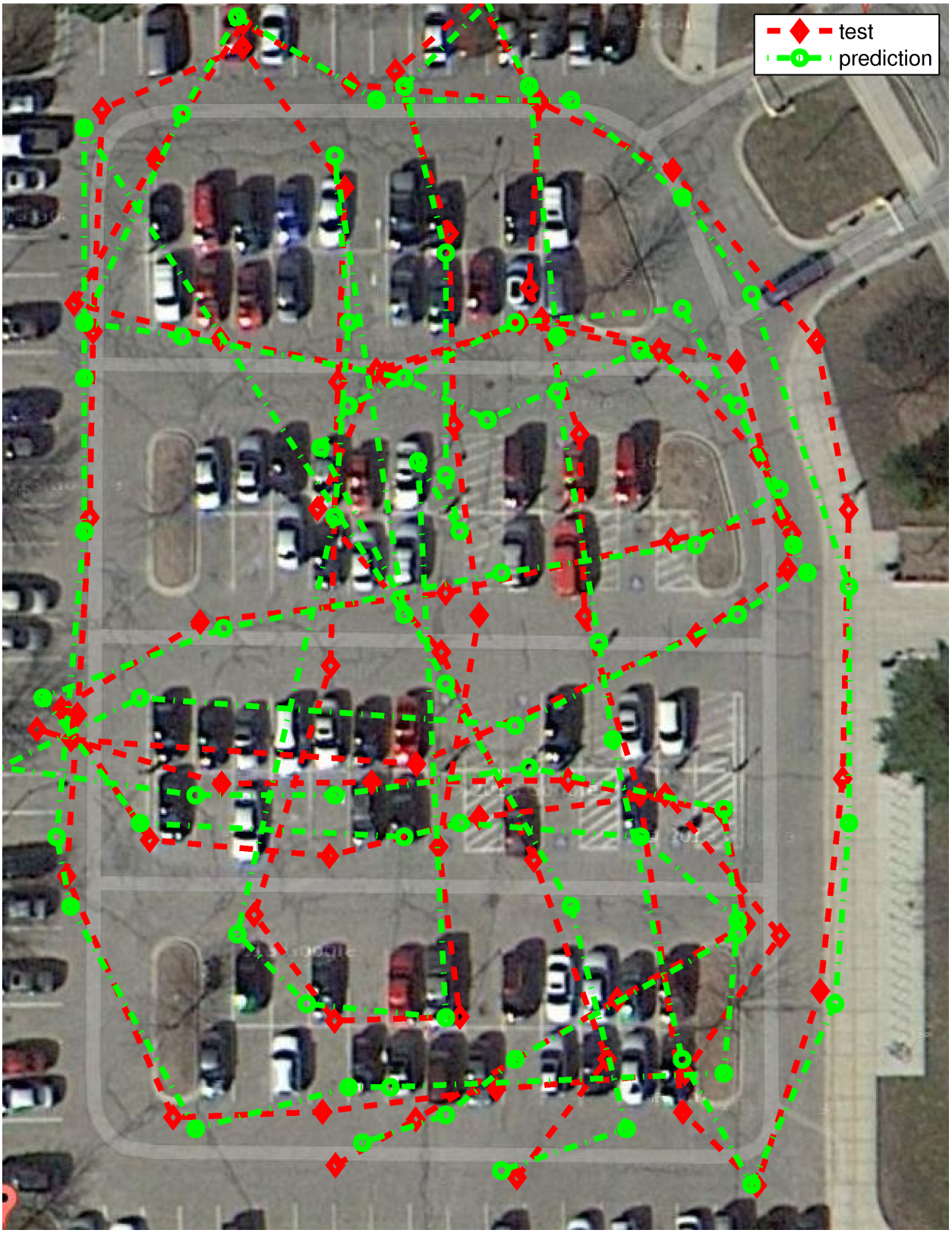


Figure 2.7 Prediction result for Case 2 with the histogram. The test path and the prediction are plotted over the Google Map image in red diamonds and green dots, respectively.

Chapter 3

Feature selection in spatial space

Recall that in chapter 2, we have used the error e between the location X and the inverse mapping of the reconstructed visual feature vector $\hat{Y}$, i.e., $e = \|X - F^{-1}(\hat{Y})\|^2$. Then, the localization is done by finding the location X that minimizes the error e. In this chapter, we approach the problem in the opposite respective: the error function now is changed to $e = \|X - F(Y)\|^2$. Using a direct mapping from visual feature to location has been emerged recently under the terminology "appearance-based localization". Recent study has shown that appearance-based localization techniques are more robust to noise, view obstruction, and image quality inconsistency in comparison with geometrical markers [102]. Localization of a robot relative to its environment using vision information has received extensive attention over past few decades from the robotic and computer vision communities [7, 19]. Since it provides more accurate estimation of positions, its applications are not limited in localization. For instance, minimizing levels of location uncertainties in sensor networks or robotic sensors is important for regression based prediction of environmental fields [14, 55].

The appearance-based method is based on a supervised learning framework that utilizes collections of location coordinates and visual observations. Fundamentally, models are learned from a large number of images at the training step, then newly collected images are recognized [43]. Visual features have been heavily used recently in robotic applications.

In [117], a neural network is used to learn the implicit relationship between the pose displacements of a 6-DOF robot and the observed variations in global descriptors of the image such as geometric moments and Fourier descriptors. In similar studies, gradient orientation histograms [67] and low dimensional representation of the vision data [110] are used to localize mobile robots. In [83], an algorithm is developed for navigating a mobile robot using a visual potential. The visual potential is computed from the image appearance sequence captured by a camera mounted on the robot. A method for recognizing scene categories by comparing the histograms of local features is presented in [70]. Without explicit object models, by using global cues as indirect evidence about the presence of an object, they consistently achieve an improvement over an orderless image representation. Additionally, the SURF feature has been often used for classification type problems such as topological localization (image matching) [112] and place recognition [111]. Se et al. [99] investigated a closely related problem by using the ego-motion technique to determine the pose of camera based on the SIFT features [75]. In [18], the authors developed an x-ray vision system to build a map of a totally unseen territory by utilizing the wifi signal. While the selection of visual features for such applications determines the ultimate performance of the algorithms, this topic has not been fully investigated to date.

In most of the cases, utilizing the full set of features is unlikely to yield optimal outcomes due to the present of redundant features. Additionally, an exhaustive search by executing tasks with all possible combinations of features is computationally unfeasible. Therefore, selecting the most effective features is critical in two categories: (1) to maximize the performance of the tasks and (2) to eliminate redundant features. For example, in localization tasks, the selected features must be robust to local noise and sensitive to different locations over the spatial field. One example of widely used feature selection technique is the Principal component analysis (PCA) [58], which is utilized to find the mapping from image features to the robot locations [98]. There are a number of methods to determine the significant variables in PCA. According to [87], the merit of those methods highly depends on the extent

of correlation of variables. Another example is reported in [8], the author utilized a iterative graph theory algorithm to obtain a Connected Dominating Graph to reduce the number of images needed to be stored in training data set, as well as retain salient SIFT features. However, since those methods are unsupervised learning, the selection process might not be optimized for the localization performance. Thus, we focus our investigation to develop a feature selection method based on a supervised learning framework.

The recent research efforts that are closely related to our problem are summarized as follows. The location for a set of image features from new observations is inferred by comparing new features with the calculated map [12, 78, 79]. In [116], a neural network is used to learn the mapping between image features and robot movements. Similarly, there exists effort on automatically finding the transformation that maximizes the mutual information between two random variables [115].

We first unwrap the omni-directional images into panoramic ones and extract three types of visual features, which are Fast Fourier Transform, color histogram, and SUFT. We concatenate all different types of features together and normalize the feature vectors to eliminate dominancy in value among different features. Then, we treat the robot's coordinates as multivariate targets and train the group LASSO [82] regression model on the training data set. Once the model is trained, we feed the newly captured image into the model to obtain an estimated position. Next, we treat the estimated position as a noisy observation and deploy a filtering estimator, e.g., EKF or PF, to remove the "jumping around" noise in the group LASSO regression outcomes. Preliminary results for indoor and outdoor experiments were reported in [20] and [21], respectively.

The overall structure of this chapter is as follows. We will introduce the visual features that are used in this study in Section 3.1. We will then discuss the LASSO regression in Section 3.2 and the group LASSO regression in Section 3.3. In Section 3.4, we will show how to integrate the EKF into the LASSO based localization performance. Experiment results are discussed in Section 3.5. For notations, we use $\mathbf{A}_{[i,j]}$ for the entry of row i, column j,

$\mathbf{A}_{[i,:]}$ for the i-th row of the matrix $\mathbf{A}$, and $\mathcal{N}(0, \sigma^2)$ as the normal distribution with zero mean and variance σ^2.

3.1 Image features

Inspired by a biological observation that insects and arthropods develop their nervous system based on a wide view sense of sight, recently omni-directional cameras have been heavily studied for navigation [119]. In contrast to conventional cameras with restricted fields of view, panoramic cameras can provide complete views of the surrounding environment in a single image. In order to extract useful features from an omni-directional image, we first unwrap it as follows. The raw image captured by the omni-directional camera is a nonlinear mapping of a 360° panoramic view onto a doughnut shape. We recover the panoramic view by applying a reverse mapping of the pixels from the wrapped view onto a panoramic view [17, 78]. From the panoramic images, we extract three types of visual features. In this chapter, we will use the notation $\mathbf{x}_i$ generally for the collection of all types of the image features that are extracted from image i. In particular, we use the Fast Fourier Transform (FFT) coefficients, the histogram, and SURF as the image features [70]. These feature types and their properties are discussed in Section 2.1.

For the indoor environment, we utilize the FFT and Histogram, since they are more robust to the dynamic change of the surrounding compared to the SURF. For the outdoor experiment, since a large portion of the image is covered by uniform tonal pixels of the sky, the FFT and Histogram features are not sensitive to the change in surrounding scenes. The SURF features capture the local properties of points of interest in an image other than the global properties like the other two. Therefore, we use the SURF for the outdoor experiment. Features from different types are concatenated and normalized.

3.2 The LASSO

Suppose that one image is captured for every sampling time i. Thus, we have the data $(\mathbb{x}_i, y_i), i \in \{1, \cdots, N\}$ where $\mathbb{x}_i = [x_i^{[1]}, \cdots, x_i^{[p]}]^T$ and y_i are the input variable vector and the target, respectively. Define $\mathbf{X} \in \mathbb{R}^{N \times p}$ and $\mathbf{y} \in \mathbb{R}^{N \times 1}$ as the collections of N observations. We assume that the input vector and the target are standardized. Let $\hat{\mathbf{b}} = [\hat{b}_1, \cdots, \hat{b}_p]^T$ be the linear fitting estimate vector that provides the estimated target vector $\hat{\mathbf{y}}$, i.e.,

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\mathbf{b}}. \tag{3.1}$$

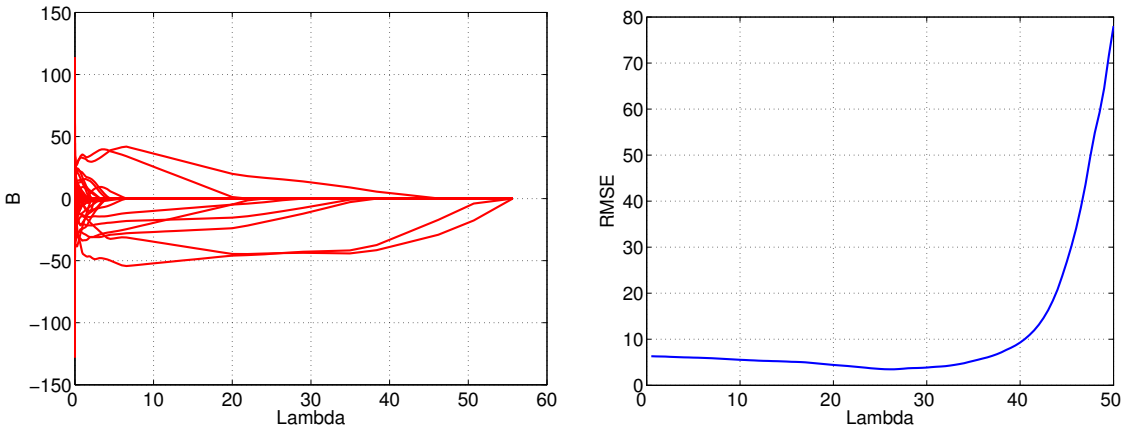
The vector $\hat{\mathbf{b}}$ can be computed by the least squares minimization as:

$$\hat{\mathbf{b}} = \underset{\mathbf{b}}{\operatorname{argmin}} \sum_{i=1}^{N} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|_2^2, \tag{3.2}$$

where $\|.\|_2$ is the ℓ_2 norm. The LASSO regression introduces a penalty term into (3.2) as follows.

$$\hat{\mathbf{b}} = \underset{\mathbf{b}}{\operatorname{argmin}} \left(\frac{1}{2N} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|_2^2 + \lambda \sum_{i=1}^{p} |b_i| \right), \tag{3.3}$$

where $\lambda > 0$ is a penalty parameter. The level of shrinkage applied to the estimated $\mathbf{b}$, i.e., the number of zero entries of $\mathbf{b}$, is determined by λ. For instance, Figure 3.1-(a) shows the evolution of entries of the vector $\hat{\mathbf{b}}$ with λ from 0 (all entries are non-zero) to a final value (all entries are zeros). Notice that as λ increases from 0, we remunerate the least squares estimation accuracy for the sparseness of vector $\mathbf{b}$. The collection of vector $\hat{\mathbf{b}}$ that is corresponding to different values of λ is obtained from the training data set. Then, it is applied to the validation data set to form the error curve. An example of an error curve is shown in Figure 3.1-(b). Finally, the optimal λ is chosen at the minimum of the error curve. In this study, we use the Root Mean Squared Error (RMSE) to compare two trajectories so the optimal λ is chosen at the minimum of the RMSE curve.



(a) (b)

Figure 3.1 (a) Shrinkage of estimate vector entries with respect to different values of λ. (b) error curve of different estimate vectors $\mathbf{b}$ applied on validation data set.

3.3 The group LASSO

In Section 3.2, the estimated target y_i is a scalar quantity. However, since the two coordinates of the robot's location need to be estimated simultaneously, we show how we extend the LASSO regression to estimate a multivariate target, i.e., the group LASSO regression [82].

Let $\mathbf{Y} \in \mathbb{R}^{N \times k}$ as the collection of N target vectors of k dimensions $\mathbf{y}_i$. Define $\hat{\mathbf{B}} \in \mathbb{R}^{p \times k}$ as the estimate matrix of p estimate vector of k dimensions that estimates the target $\hat{\mathbf{Y}}$:

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\mathbf{B}}. \tag{3.4}$$

Note that $k = 2$ in this study, which is corresponding to two spatial coordinates. We define the ℓ_1/ℓ_2 norm as:

$$\|\mathbf{B}\|_{\ell_1/\ell_2} = \sum_{i=1}^{p} \left(\sum_{j=1}^{k} \mathbf{B}_{ij}^2 \right)^{\frac{1}{2}} = \sum_{i=1}^{p} \|\mathbf{B}_{[i,:]}\|_2, \tag{3.5}$$

where $\mathbf{B}_{[i,:]}$ is the i-th row of the matrix $\mathbf{B}$. The estimate matrix $\hat{\mathbf{B}}$ can be calculated as follows.

$$\hat{\mathbf{B}} = \underset{\mathbf{B}}{\operatorname{argmin}} \left(\frac{1}{2N} \|\mathbf{Y} - \mathbf{X}\mathbf{B}\|_F^2 + \lambda \|\mathbf{B}\|_{\ell_1/\ell_2} \right), \tag{3.6}$$

where $\|.\|_F$ denotes the Frobenius norm[1]. According to Obozinski et al. [82], (3.6) is equivalent to the *second-order cone program* (SOCP):

$$\begin{aligned} \hat{\mathbf{B}} = \underset{\mathbf{B},\boldsymbol{\beta}\in\mathbb{R}^{p\times 1}}{\operatorname{argmin}} & \left(\frac{1}{2N}\|\mathbf{Y}-\mathbf{X}\mathbf{B}\|_F^2 + \lambda \sum_{i=1}^{p} \beta_i \right), \\ & \text{subject to } \|\mathbf{B}_{[i,:]}\|_2 \leq \beta_i. \end{aligned} \tag{3.7}$$

Define vec(.) operator as:

$$\text{vec}(\mathbf{B}) = \begin{bmatrix} \mathbf{B}_{[:,1]} \\ \mathbf{B}_{[:,2]} \\ \vdots \\ \mathbf{B}_{[:,k]} \end{bmatrix}.$$

Applying the vec(.) operator to both sides of (3.7) yields:

$$\begin{aligned} & \text{vec}(\hat{\mathbf{B}}) = \\ & \underset{\text{vec}(\mathbf{B}),\boldsymbol{\beta}}{\operatorname{argmin}} \left(\frac{1}{2N}\|\text{vec}(\mathbf{Y}) - (\mathbf{I}\otimes\mathbf{X})\text{vec}(\mathbf{B})\|_2^2 + \lambda \sum_{i=1}^{p} \beta_i \right), \\ & \text{subject to } \|\mathbf{W}_i \text{vec}(\mathbf{B})\|_2 \leq \beta_i, \end{aligned} \tag{3.8}$$

where $\otimes$ is the Kronecker product [36] and $\mathbf{W}_i \in \mathbb{R}^{k\times kp}$ is the weight matrix[2], i.e., $\mathbf{W}_1 = [\mathbf{I}_k \cdots \mathbf{O}_{kp-k}]$, $\mathbf{W}_2 = [\mathbf{O}_k\ \mathbf{I}_k \cdots \mathbf{O}_{kp-2k}], \cdots$.

Note that the ℓ_1/ℓ_2 regularization penalizes the *norm* of the rows of estimate matrix $\mathbf{B}$ in (3.6), which are corresponding to the two coordinates. Thus features that are not significant for the localization of both coordinates are more likely to be eliminated.

[1]The Frobenius norm of a matrix $\mathbf{A}$ is: $\|\mathbf{A}\|_F := \sqrt{\sum_{i,j} \mathbf{A}_{i,j}^2}$

[2]$\mathbf{I}_k$ and $\mathbf{O}_k$ are the identity and zero matrices of size $k \times k$

3.4 The extended Kalman filter

In this section, we show how to integrate the EKF into the group LASSO-based localization as a post-processing step. For the sake of simplicity, we describe in this section the bicycle kinematics model [16] of a front wheel driven vehicle for the outdoor experiment. For the indoor experiment, we utilize the unicycle kinematics model that effects the equation (3.9), and the rest of the EKF shall stay unchanged.

We utilize the bicycle kinematics for a front wheel driven vehicle. Let η_i, u_i and L be the front wheel angle, the rear wheel linear speed at time iteration i and the wheelbase (distance between the two wheel axes) of the vehicle, respectively. Define $\mathbf{x}_i \in \mathbb{R}^3$ as the state vector that includes the position and the heading angle of the robot, e.g., $\mathbf{x}_i = [q_i^{[1]} q_i^{[2]} \psi_i]^T$. Therefore, the state transition equation of the vehicle can be described as follows.

$$\begin{aligned} \mathbf{x}_{i+1} &= \mathbf{x}_i + \Delta i \begin{bmatrix} u_i \cos\psi_i \\ u_i \sin\psi_i \\ \frac{\tan(\eta_i)}{L} u_i \end{bmatrix} + \omega_i \\ &= f(\mathbf{x}_i, u_i, \eta_i) + \omega_i, \end{aligned} \tag{3.9}$$

where $\omega_i \sim \mathcal{N}(0, \Sigma_{\omega_i})$ is the white noise on the location and orientation. Let $\hat{\mathbf{x}}_{i+1}^-, P_{i+1}^-$ be the estimated state and the covariance matrix of the robot before taking the position measurement, respectively. Assume that localization errors for x, y axes and the orientation ψ are independent, thus:

$$P_i = \begin{bmatrix} \sigma_{1,i}^2 & 0 & 0 \\ 0 & \sigma_{2,i}^2 & 0 \\ 0 & 0 & \sigma_{\psi_i}^2 \end{bmatrix}. \tag{3.10}$$

We have the prediction as follows.

$$\begin{aligned} \hat{\mathbf{x}}_{i+1}^{-} &= f(\mathbf{x}_i, u_i, \eta_i) \\ P_{i+1}^{-} &= \nabla f P_i \nabla f^T + \Sigma_{\omega_i}, \end{aligned} \tag{3.11}$$

where ∇f is the Jacobian matrix of the function f with respect to the state vector $\mathbf{x}_i$, e.g.,

$$\nabla f_{x_i} = \begin{bmatrix} 1 & 0 & -\Delta i u_i \sin\psi_i \\ 0 & 1 & \Delta i u_i \cos\psi_i \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.12}$$

where Δi is the sampling period. Since we do not take the measurement of the vehicle's heading angle, the observation has the form:

$$\tilde{\mathbf{q}}_i = M\mathbf{x}_i + e_i, \tag{3.13}$$

where

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$

and $e_i \sim \mathcal{N}(0, \sigma_e^2 I)$. Note that $\tilde{\mathbf{q}}_i = \hat{\mathbf{y}}_i$ in (3.4) since we take the estimation of group LASSO-based localization as the noisy measurement for the EKF, and the observation noise level σ_e is set to be the RMSE of the group LASSO-based localization. The measurement update can be computed as follows.

$$\begin{aligned} \hat{\mathbf{x}}_{i+1} &= \hat{\mathbf{x}}_{i+1}^{-} + K_{i+1}(\tilde{q}_{i+1} - M\hat{\mathbf{x}}_{i+1}^{-}) \\ P_{i+1} &= (I - K_{i+1}M)P_{i+1}^{-}, \end{aligned} \tag{3.14}$$

where the Kalman gain K_{i+1} is:

$$K_{i+1} = P_{i+1}^{-} M^T (M P_{i+1}^{-} M^T + \sigma_e^2 I)^{-1}. \tag{3.15}$$

The prediction and update measurement computations, i.e., equations (3.11) and (3.14), are recursively executed in the localization process.

Note that for the two wheels mobile robot, (3.9) becomes:

$$\begin{aligned}\mathbf{x}_{i+1} &= \mathbf{x}_i + \Delta i \begin{bmatrix} u_i \cos\psi_i \\ u_i \sin\psi_i \\ \eta_i \end{bmatrix} + \omega_i, \\ &= f(\mathbf{x}_i, u_i, \eta_i) + \omega_i,\end{aligned}$$

where η_i is the robot's orientation.

3.5 Experimental study

3.5.1 Indoor Experiment

In this section, we show how we implement our proposed method in an indoor experimental set-up. The mobile robot and the testing environment is shown in Figure 3.2-(a).

We control the robot remotely via wireless communication units while the panoramic vision of the surrounding scene is recorded. In order to track the true positions of the robot, we utilize an overhead camera (Logitech HD webcam C910, Logitech, Newark, CA, U.S.A.) that is mounted on the ceiling. Note that the positions obtained from the ceiling camera are only used for evaluating the method's performance. In summary, we collect the following three data sets all sampled at 0.5 Hz.

- The command inputs, turn rates and collapsed time between samplings $\{u_i, \eta_i, \Delta_i\}$.
- The surrounding scene recorded in the panoramic vision.
- The positions of the mobile robot (ground truth).

We compare the localization results performed by the three following techniques:

- **Open-loop based localization:** We naively apply the recorded command inputs to the robot's kinematics, which is described in (3.9) to obtain the robot's position.

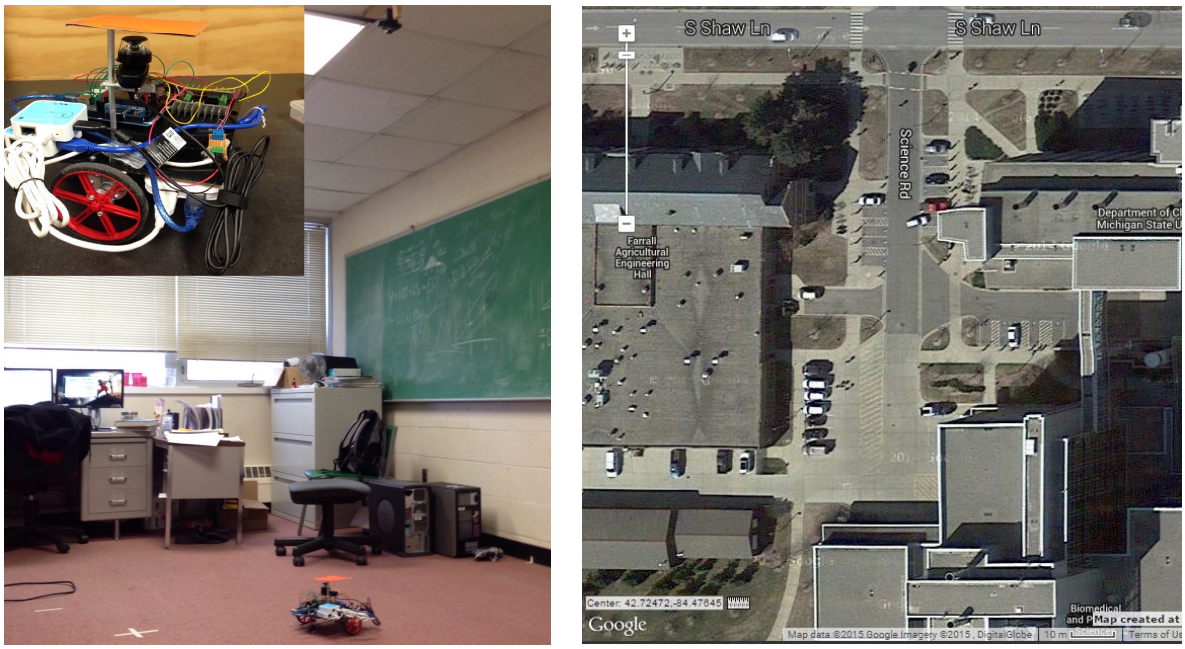
- **Group LASSO-based localization:** We randomly sample without replacement the whole data set and categorize them into three labels: *training*, *validation* and *testing* by corresponding percentages: 50%, 25% and 25%. Then, we apply the group LASSO regression to the labeled data and compare the performance on the test data with other techniques.

- **Group LASSO-based and EKF localization:** We apply the EKF as a post-processing to the test result of the group LASSO-based localization as described in Section 3.4.

- **Group LASSO-based and PF localization:** We apply the Particle Filter (PF) [35] as a post-processing to the test result of the group LASSO-based localization.

Note that for the indoor experiment, the test data is the locations that are randomly chosen from the original trajectory. Therefore, when we run the EKF and the PF to estimate the trajectory of the robot, we assume that the robot do not take measurement at the non-test locations, i.e., estimation with *missing* observations [104]. If an observation is made, the EKF executes the update step by (3.14). Otherwise, it predicts the state vector by open-loop kinematics and propagates the covariance matrix by (3.11). Similarly, for the PF, the weights of the filter density are updated once an observation is made. Otherwise, a uniform probability is assigned for all particles [35].

To illustrate the estimation with missing observation by the EKF, Figure 3.3 shows the evolutions of the first and second entries of the diagonal of matrix P_i ($P_{[1,1]}$ and $P_{[2,2]}$) that represent the uncertainties in x and y axes. Notice that the variance of position coordinates drops whenever an observation, which is indicated by a vertical dashed line in Figure 3.3, is made. This simulates a practical situation where the localization observations are mostly missing and very sparse due to line-of-sight or GPS-denied environments.



(a) (b)

Figure 3.2 (a) Indoor experiment environment and the zoomed-in picture of the mobile robot shown in the upper left corner. (b) Outdoor experiment in the campus of Michigan State University on Google map.

In Table 3.1 we compare the localization results performed by open-loop localization, group LASSO-based localization, group LASSO-based with EKF localization and group LASSO-based with PF localization which are denoted as "Open-loop", "Group LASSO", "Group LASSO+EKF" and "Group LASSO+PF", respectively.

Figure 3.4 shows the evolution of the elements of group LASSO estimate matrix $\mathbf{B}$. Each row of $\mathbf{B}$ consists of two elements that correspond a feature to the location with two coordinates. Each pair of elements is plotted in the same color in Figure 3.4. Notice that elements from a pair (same colors) are both either eliminated or preserved.

Table 3.1 shows the RMSEs of the four methods. The open-loop prediction yields the worst performance, since there is no feedback in the prediction, the error is accumulated. The group LASSO-based localization reduces 50%[3] number of features and yields 29% lower RMSE compared to the open-loop method. By applying the EKF, the group LASSO-based

[3]The elements of the estimate matrix $\hat{\mathbf{B}}$ that are smaller than 1×10^{-3} are considered to be 0.

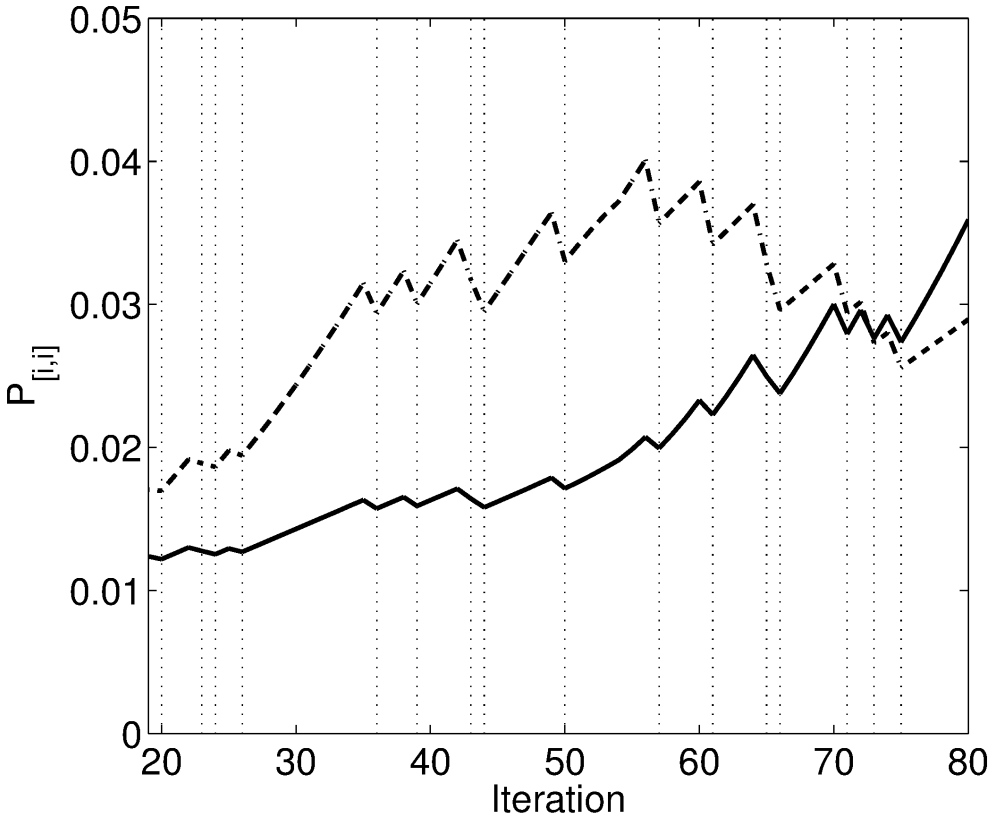


Figure 3.3 Plot of $P_{[1,1]}$ (dashed line) and $P_{[2,2]}$ (solid line) for iterations from 20 to 80 for the group LASSO-based with EKF localization.

	Method	RMSE (meters)	Number of features	Run time (sec) Train	Test
Indoor	Open-loop	0.7039	-	-	0.033
	Group LASSO	0.4993	27/60	2061.7	0.005
	Group LASSO+EKF	0.3158	27/60	2061.7	1.271
	Group LASSO+PF (MC)	(0.39, 0.01)	27/60	2061.7	0.532
Outdoor	Open-loop	7.32	-	-	0.686
	Group LASSO	22.13	49/200	1861.8	29.376
	Group LASSO+EKF	5.08	49/200	1861.8	29.388
	Group LASSO+PF (MC)	(11.87, 2.93)	49/200	1861.8	29.543

Table 3.1 Localization performance comparison

and EKF localization results in the lowest RMSE with 55% reduction compared to the open-loop case. Figure 3.5 shows the true trajectory (solid line), group LASSO-based localization (square dots), group LASSO and EKF localization (red dotted dashed line), and group LASSO and PF localization (black dotted line). Table 3.1 has indicated the excellent performance of our proposed method.

3.5.2 Outdoor Experiment

In this section, we demonstrate the effectiveness of our proposed method using an outdoor experimental study.

Figure 3.6 shows the data acquisition circuit and the surveillance vehicle. We drive the vehicle (Acura RSX 2003) in Michigan State University campus and record the panoramic scene via an omni-directional camera installed on the top of the vehicle. We collect three data sets, which are sampled at 1 Hz: (1) The estimated control inputs and sampling times $\{\eta_i, u_i, \Delta_i\}$, (2) the scene recorded by the omnidirectional camera that is stored in the Raspberry Pi (Raspberry Pi model B+, Raspberry Pi Foundation, United Kingdom), (3) the positions of the vehicle that are tracked by the Xsens GPS (MTi-G-700, Xsens Technologies B.V., Netherlands).

We compare the localization performances based on the following three techniques:

- **Open-loop based localization:** We naively apply the open-loop prediction by using

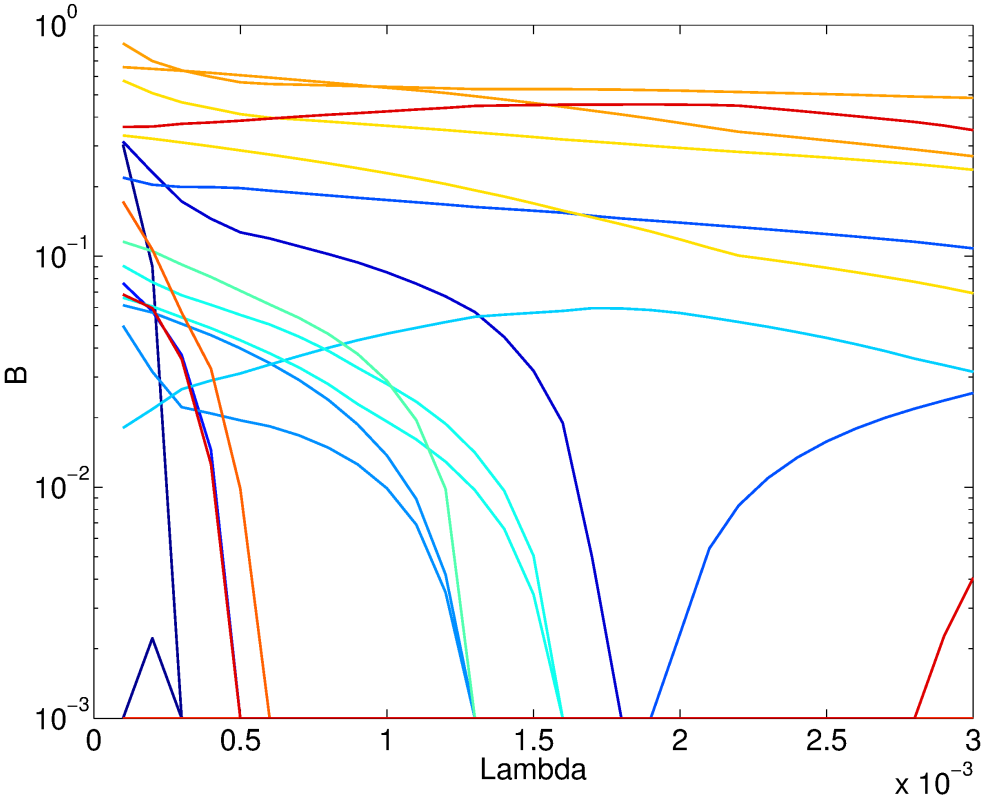


Figure 3.4 Indoor experiment: The evolution of the entries of estimate matrix $\mathbf{B}$ versus the penalty λ. Each pair of features that associate with same coordinate is plotted in same color.

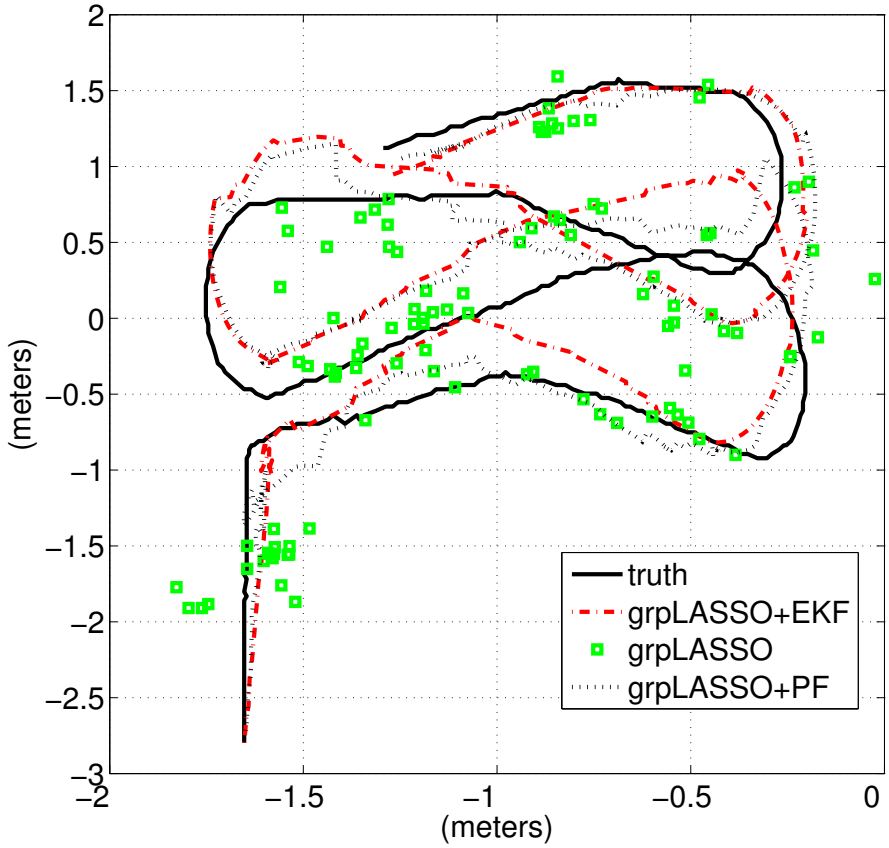


Figure 3.5 Indoor experiment: The true trajectory (black solid line), group LASSO (green squares), group LASSO + PF (dotted black line) and group LASSO + EKF (red dotted dashed line) predictions.

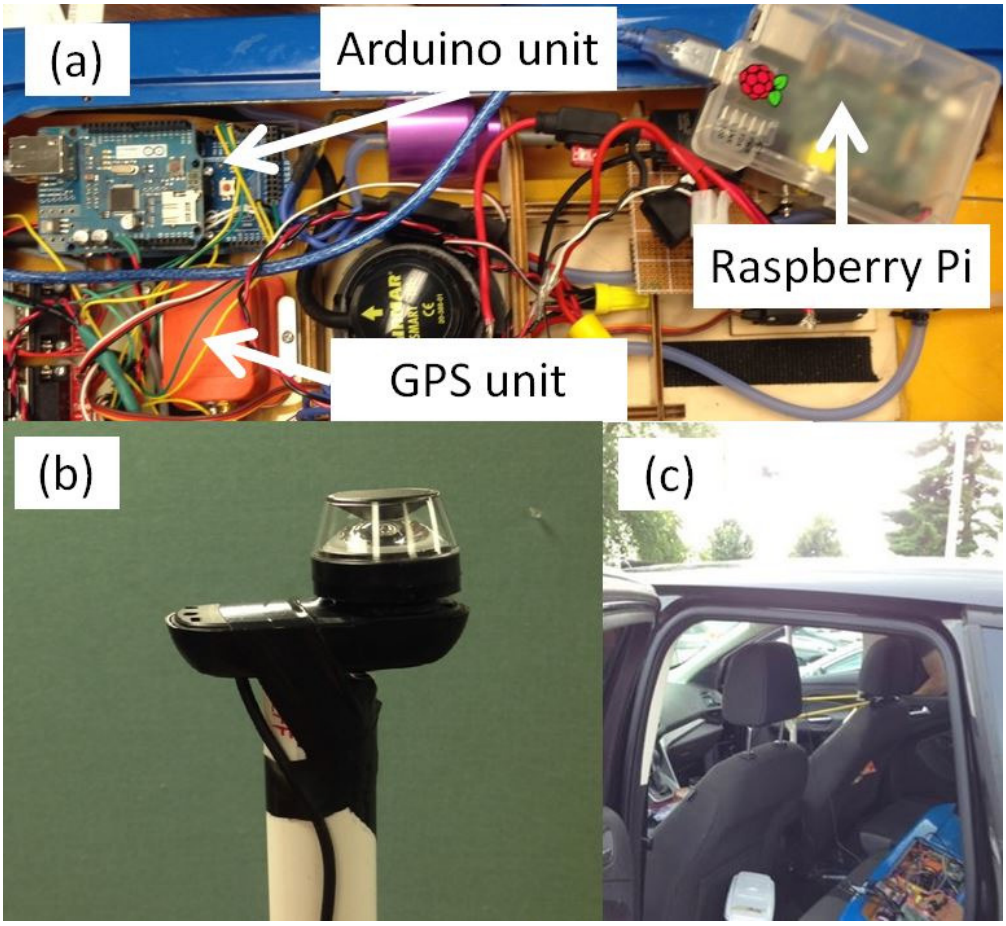


Figure 3.6 (a) Data acquisition circuit, (b) panoramic camera and (c) vehicle equipped with the camera on top.

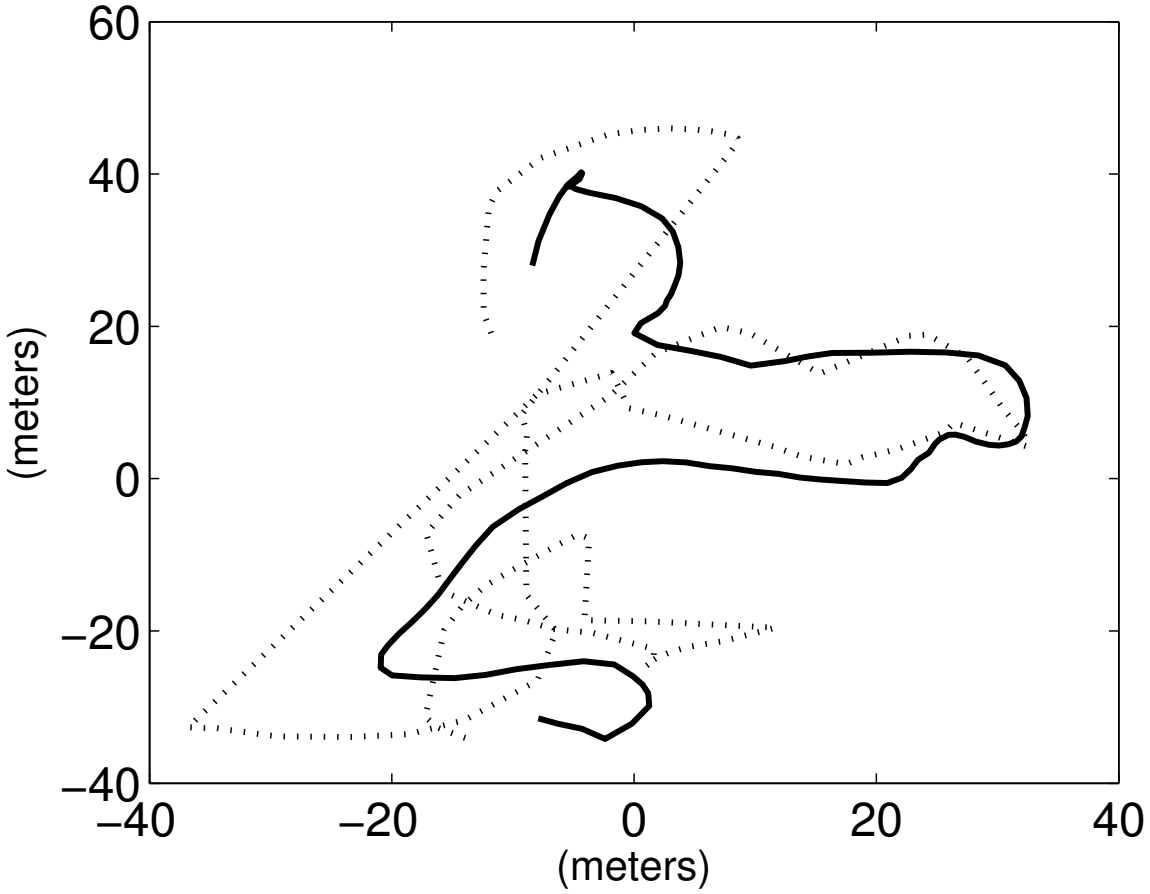


Figure 3.7 The training (dashed) and testing paths (solid lines) are plotted in meters.

the kinematics described in (3.9) and the command inputs.

- **Group LASSO-based localization:** We record two separated trajectories for training and testing, as shown in Figure 3.7. The original data is divided 50-50 for training and validation. The group LASSO regression uses the training data to train the estimate matrix $\mathbf{B}$, the validation data to compute the optimal matrix $\hat{\mathbf{B}}$ and then applies $\hat{\mathbf{B}}$ to estimate the test data $\hat{\mathbf{Y}}$.

- **Group LASSO and EKF localization:** We apply the EKF as a post-processing to the test result of the group LASSO-based localization as described in Section 3.4.

- **Group LASSO and PF localization:** We apply the PF as a post-processing to the test result of the group LASSO-based localization.

For this outdoor experiment, we also use the RMSE to evaluate the performance of the three techniques described above.

Table 3.1 shows the RMSEs of the three methods (column 2) and the number of used features over the initial total (column 3). The group LASSO-based localization reduces 75.5%[4] of the whole features. By applying the EKF, the group LASSO-based localization with the EKF results in the best RMSE with 30.6% reduction compared to the open-loop method. Figure 3.8 shows the true trajectory (black solid line), group LASSO-based localization (green square dots), group LASSO and EKF localization (red dotted dashed line), and group LASSO and PF localization (black dotted line).

Fundamentally the estimation step of the group LASSO-based localization shown in (3.4) is a linear regression. Thus, its estimation error has a smaller bias compared to the open-loop prediction, which gradually deviates from the true trajectory, but exhibits high variance. Therefore, we utilize the EKF (as well as PF) as a low-pass filter to smooth of the estimated trajectory by projecting the noisy outputs of the group LASSO-based localization onto the

[4]The entries of the estimate matrix $\hat{\mathbf{B}}$ that are smaller than 1×10^{-3} are considered to be 0.

vehicle kinematics. Figure 3.9-(a) shows the evolution of the group LASSO regression estimate matrix $\mathbf{B}$. Each pair of features that is corresponding to one location (two coordinates) is plotted in the same color. The optimal $\mathbf{B}$ is plotted in Figure 3.9-(b). Note that the matrix is sparse with a large number of entries having values of zero.

For comparison, we also implement the particle filter (PF) [1] that takes the group LASSO output as measurements. Since the PF involves randomly generation of the particles, adapting the evaluation criteria from [1], we evaluate the performance of the PF by the Monte Carlo (MC) simulation [80] with 100 runs. The statistics of the MC are reported in Table. 3.1 with the mean and standard variation denoted by "(μ, σ)". The estimated path by the group LASSO and PF localization with lowest RMSE among the MC simulations is plotted in black dotted line in Figure 3.8. A snapshot of the PF at sampling time $t = 50$ is shown in Figure 3.10. The weight densities of 800 particles are plotted in gray-scale colored dots. Note that the sum of weight densities over all particles equals to 1. The PF estimated location is computed by averaging the locations of all particles with the corresponding weight densities.

The computational time is reported in Table. 3.1 in seconds. Since the EKF and PF are applied as the post-processing after the group LASSO, their test phase computation times are reported as the sum of the group LASSO and the additional running time to perform the EKF or PF. Generally the train phase requires 30 minutes for both indoor and outdoor data, while the test phase is significantly shorter. The prediction process of the group LASSO includes one matrix multiplication in (3.4) and the visual features extraction. Since the SURF feature involves clustering, it requires more time to extract compared to the FFT and histogram. Therefore, the test phase computational time in the outdoor experiment is longer than in the indoor case.

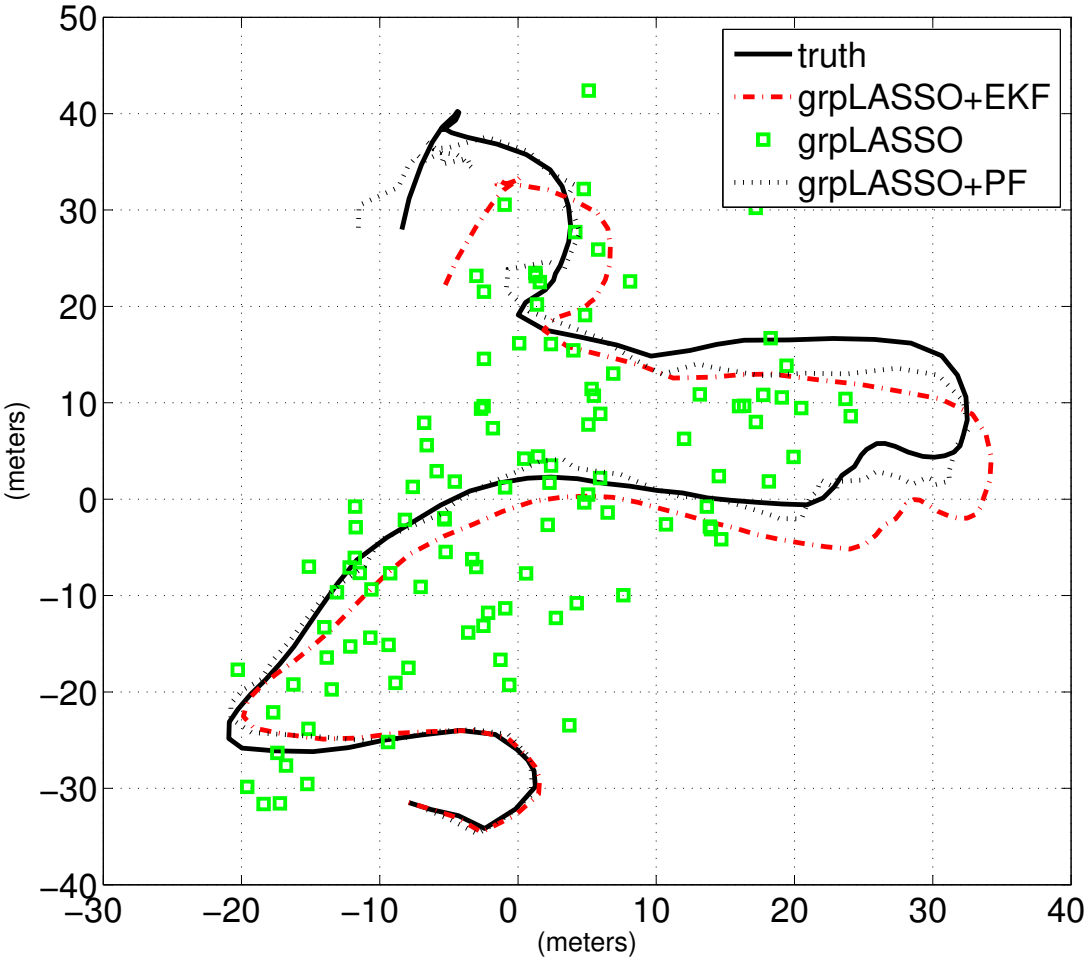
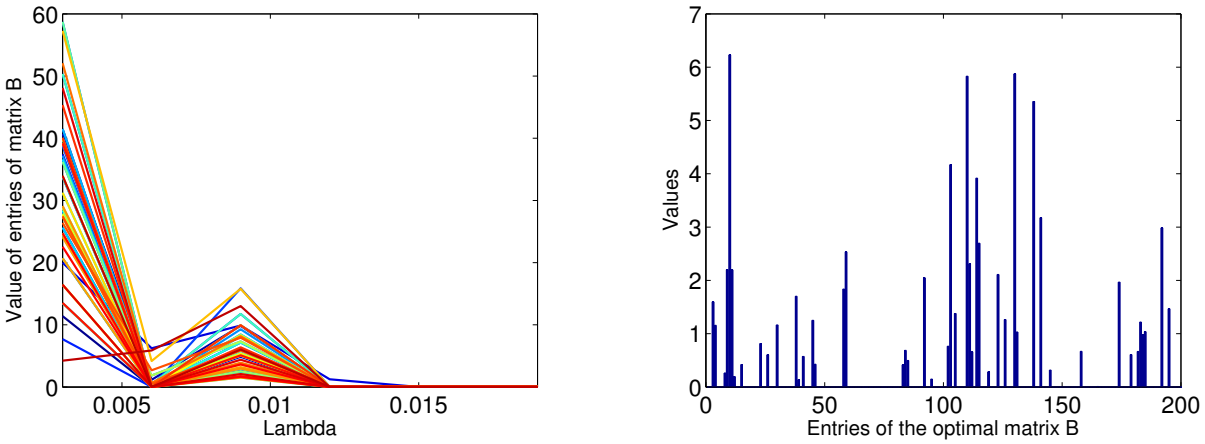


Figure 3.8 Outdoor experiment: The true trajectory (black solid line), group LASSO (green squares), group LASSO + PF (black dotted line), and group LASSO + EKF (red dotted dashed line) predictions are plotted in meters.



(a) (b)

Figure 3.9 Outdoor experiment: (a) The evolution of entries of the estimate matrix $\mathbf{B}$ versus the penalty λ. (b) Overall 200 entries of the optimal matrix $\mathbf{B}$ are plotted in blue bars.

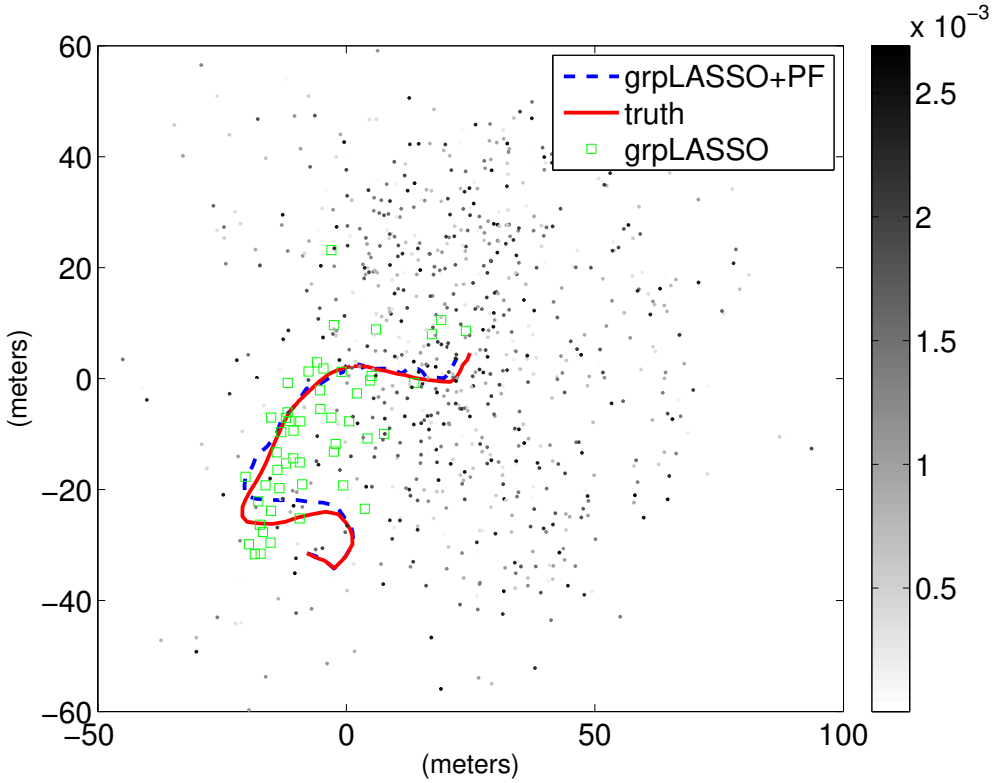


Figure 3.10 The snapshot of the Particle Filter at $t = 50$: The true trajectory (red solid line), group LASSO (green squares) and group LASSO + PF (blue dashed line) predictions are plotted in meters. Each particle is plotted with the color in gray scale corresponding to its probability weight. The sum of the weights of all particles is 1.

Chapter 4

Abdominal Aortic Aneurysm's shape prediction

The aorta is a major artery in which blood circulates through the heart. An aortic aneurysm is identified as an enlargement of the aorta greater than 50% of the normal diameter. The vast majority of aortic aneurysms are in the abdominal region, among these, over 90% occur within the infrarenal aorta [89] [126]. The infraneral aorta lies between the renal branches and the iliac bifurcation. In this region, a diameter greater than 3 cm is considered as an AAA. In most of the cases, AAAs have no symptoms and are found incidentally, but if one ruptures the patient mortality rate can be more than 90% [62] [63]. Because risks from open surgery or endovascular repair outweigh the risk of AAA rupture, surgical treatments are not recommended with AAAs less than 5.5cm in diameter [85]. However, the maximum transverse diameter of an AAA is not a reliable indicator of rupture potential on its own. Therefore, various biomechanical analyses using geometrical factors (e.g., different diameter measurements [34, 65], tortuosity [84], morphological parameters [91], presence of thrombus [6, 76, 125], influence of the spine [27]) have been proposed to reliably predict the risk of rupture. Although recent advances in biomechanics have provided state-of-the-art spatial estimates of stress distributions of AAAs, but these studies do not address the problem

of predicting the shape at a future time and are limited in assessing the time evolution and uncertainty qualification. For clinical treatments and recommendations, a patient-specific predictive tool is required to incorporate the advances in computational modeling. The development of such tool requires a major paradigm-shift since clinical measurements are associated with limited information, uncertainty and incompleteness of the model.

In short, this chapter tackles the following problems:

- *Data-driven dynamic growth model development:* We develop a dynamic model to simulate the AAA's growth that is trained on patient's specific data.

- *Prediction of AAA geometrical changes and their validation:* Comparisons of predictions and the true (not included in the learning phase) scan images are provided to evaluate the accuracy of the proposed scheme.

- *Prediction uncertainty quantification:* The point-wise confidence interval is provided for the predicted AAA surface along with the estimation error.

- *Possible utility of the methodology:* Possible utility of the proposed method is discussed from helping decision making to feature extraction applications.

To the best of our knowledge, this is the first study that predicts the 3D shape of the AAA growth using available (patient-specific) point clouds data in a statistical perspective, which allows uncertainty quantification in the prediction.

Standard notations will be used throughout this paper. Let $\mathbb{R}, \mathbb{R}_{\geq 0}, \mathbb{R}_{>0}$, and $\mathbb{Z}$ be defined as the sets of real, non-negative real, positive real, and integer numbers, respectively. I_n denotes the identity matrix of size n. For column vectors $v_a \in \mathbb{R}^a$,$v_b \in \mathbb{R}^b$, and $v_c \in \mathbb{R}^c$, $\text{col}(v_a, v_b, v_c) := [v_a\, v_b\, v_c] \in \mathbb{R}^{a+b+c}$ stacks all vectors to create one column vector, and $\|v_a\|$ denotes the Euclidean norm (or vector 2-norm) of v_a. Let $\mathbb{E}(z)$ and $\text{Var}(z)$ denote the expectation and the variance of random variable z, respectively. A random vector $z \in \mathbb{R}^q$, which is distributed by a multivariate Gaussian distribution of a mean $\mu \in \mathbb{R}^q$ and a covariance $\Sigma \in \mathbb{R}^{q\times q}$, is denoted by $z \sim \mathcal{N}(\mu, \Sigma)$.

The chapter is organized as follows. In Section 4.1, we discuss the data acquisition that is followed by the prediction model in Section 4.2. Section 4.3 describes our post-process procedures for the estimated point cloud. Geometrical prediction results from our methodology are illustrated in Section 4.4. Finally, we discuss the results in Section 4.5.

4.1 Data Adaptation

In this work, we adapt the point clouds data that is reconstructed by the work reported in [34]. In particular, from the longitudinal patient-specific data comprises 37 computer-tomography scan images of the AAAs obtained from 7 different patients, Gharahi et al. [34] described the segmentation process of the outer surface and image registration. Then, they extract point clouds by randomly sampling from the segmented surfaces. In this work, we utilize the point clouds as the inputs to our method.

Patient ID	Number of Scans	Gender	Age	Time of Scans (Years)
P1	7	Male	68	[0, 1.07, 5.76, 6.70, 7.68, 8.64, 9.10]
P2	6	Male	66	[0, 1.02, 2.04, 2.94, 3.94, 5.85]
P3	5	Male	54	[0, 1.06, 2.07, 3.07, 3.54]
P4	5	Male	62	[0, 0.63, 1.85, 2.87, 3.84]
P5	4	Male	73	[0, 0.27, 0.74, 1.57]
P6	6	Male	70	[0, 2.12, 2.58, 3.28, 3.99, 4.31]
P7	4	Male	54	[0, 1.11, 2.15, 3.20]

Table 4.1 Demographic Data of Patients from [34]

The collection of data sets, which are in the form of point clouds, obtained from 7 patients is denoted as $\{\mathcal{D}_{i,\ell} | i \in \mathcal{I}, \ell \in \mathcal{T}\}$, where $\mathcal{I} := \{P1, P2, P3, P4, P5, P6, P7\}$ and $\mathcal{T} := \{1, \cdots, 7\}$. $\mathcal{I}$ represents the collection of patient IDs and $\mathcal{T}$ contains the available scans of a particular patient. For example, the second scan data of patient F is denoted by $\mathcal{D}_{F,2}$. A demographic data of patients that is reported in [34] is shown again in Table 4.1. As shown in the table, not all of the patients have seven data points and the remaining, non-existing scans are taken as empty data sets in the representation above.

4.2 Model and method

A Hidden Markov model (HMM) is a special case of a state space model, which has been studied extensively in control community [31] and computer science [93]. It is extraordinarily efficient to represent sequential data such as speech recognition, natural language modeling, and analysis of biological sequences [4]. A state space model is a structure that consists of a Markov chain of latent (hidden) variables and observations (visible units). Each observed unit has a conditional distribution conditioned on the corresponding latent variable. When the latent variables are discrete, we have a Hidden Markov model [4]. Inspired by the HMM, we utilize a similar graphic structure. In our model, we consider a perspective that at each data point, we observe the point cloud, which have distribution conditioned on the potential field. Also, the statistics of the potential field (which is hidden from us) can be inferred from its distribution in the past. Therefore, our general scheme is that based on the observed point cloud, we reconstruct the hidden potential field. Then, we infer the statistics of the field at the future time. Finally, we obtain the predicted point cloud from the predicted field. The detailed structure of our proposed method is shown in Fig. 4.1.

4.2.1 Gaussian process regression

In this section, we briefly review Gaussian process regression. A Gaussian process is formally defined as follows [90].

Definition 1: A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.

A Gaussian process is completely specified by its mean and covariance functions. Let $\xi \in \mathcal{Q} := \mathcal{R} \times \mathcal{T} \subset \mathbb{R}^d$ denote the index vector, where $\xi := \begin{bmatrix}\mathbf{x}^T & t\end{bmatrix}^T$ contains the sampling location $\mathbf{x} \in \mathcal{R} \subset \mathbb{R}^{d-1}$ and the sampling time $t \in \mathcal{T} \subset \mathbb{R}_{\geq 0}$.

For an illustrative purpose, we consider a Gaussian process

$$z(x) \sim \mathcal{GP}\left(\mu(\xi), \mathcal{K}(\xi, \xi')\right).$$

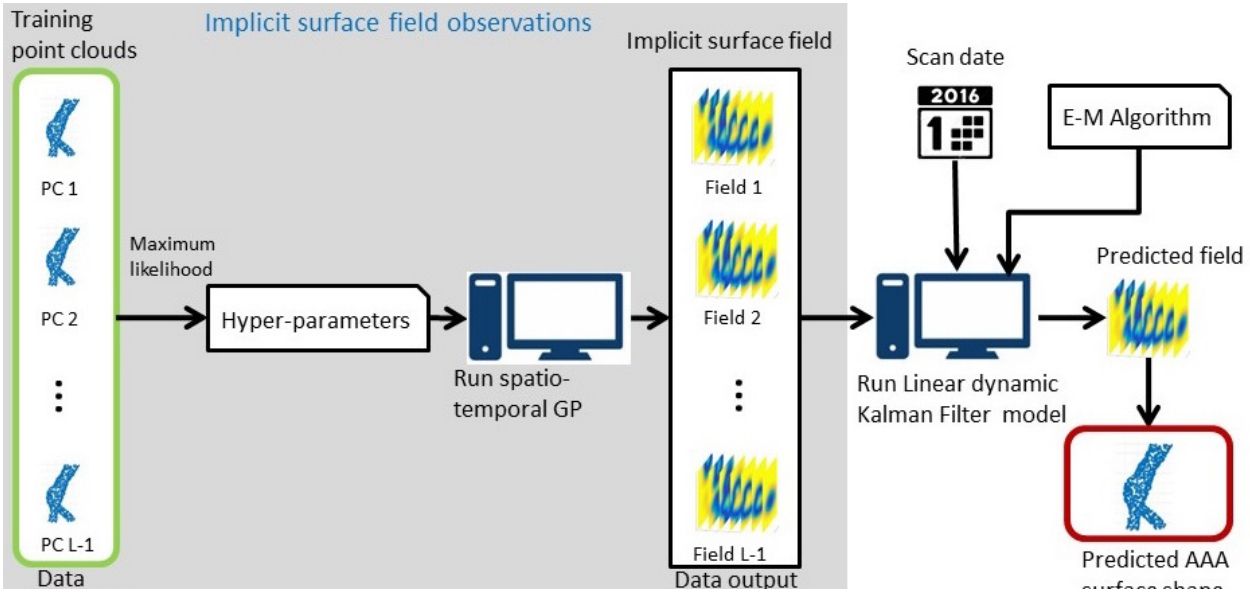


Figure 4.1 Summary of our proposed method: point cloud data is inserted into the spatio-temporal Gaussian process as zero-value observations to generate the field. Then, the temporal evolution of the field is inferred through the dynamic model in (4.8). Finally, the final prediction is computed by utilizing the Kalman Filter in the E-M algorithm.

In general, the mean $\mu(.)$ and the covariance functions $\mathcal{K}(\cdot,\cdot)$ of a Gaussian process can be estimated *a priori* by maximizing the likelihood function [124].

Suppose, we have p noise corrupted observations with $D_e = \left\{(\xi^{(i)}, \tilde{z}^{(i)})|i = 1, \cdots, p\right\}$. Assume that

$$\tilde{z}^{(i)} = z^{(i)} + \epsilon^{(i)}, \tag{4.1}$$

where $\epsilon^{(i)}$ is independent and identically distributed (i.i.d.) white Gaussian noise with variance σ_ϵ^2. $\boldsymbol{\xi}$ is defined as $\boldsymbol{\xi} = \text{col}(\xi^{(1)}, \xi^{(2)}, \ldots, \xi^{(p)})$. The collections of the realizations $\mathbf{z} = \left[z^{(1)}, \ldots, z^{(p)}\right]^T \in \mathbb{R}^p$ and the observations $\tilde{\mathbf{z}} = \left[\tilde{z}^{(1)}, \ldots, \tilde{z}^{(p)}\right]^T \in \mathbb{R}^p$ have the Gaussian distributions

$$\mathbf{z} \sim \mathcal{N}\left(\mu(\boldsymbol{\xi}), K(\boldsymbol{\xi})\right), \quad \tilde{\mathbf{z}} \sim \mathcal{N}\left(\mu(\boldsymbol{\xi}), K(\boldsymbol{\xi}) + \sigma_\epsilon^2 I_p\right),$$

where $K(\boldsymbol{\xi}) \in \mathbb{R}^{p\times p}$ is the covariance matrix of $\mathbf{z}$ and is obtained by $K_{ij}(\boldsymbol{\xi}) = \mathcal{K}(\xi^{(i)}, \xi^{(j)})$ and $I_p \in \mathbb{R}^{p\times p}$ is the identity matrix. We can predict the value z_* of the Gaussian process

at a point ξ_* [90] as

$$z_*|D_e \sim \mathcal{N}\left(\mu_*(\boldsymbol{\xi}), \sigma_*^2(\boldsymbol{\xi})\right), \tag{4.2}$$

where the predictive mean $\mathbb{E}(\mathbf{z}|D_e)$ is

$$\mu_*(\boldsymbol{\xi}) = \mu(\boldsymbol{\xi}) + k^T(\boldsymbol{\xi})\left(\mathbf{K}(\boldsymbol{\xi}) + \sigma_\epsilon^2 I_p\right)^{-1}(\tilde{\mathbf{z}} - \mu(\boldsymbol{\xi})) \tag{4.3}$$

and the predictive variance is given by

$$\sigma_*^2(\boldsymbol{\xi}) = \mathrm{Var}(z_*|D_e) = \sigma^2 - k^T(\boldsymbol{\xi})\left(\mathbf{K}(\boldsymbol{\xi}) + \sigma_\epsilon^2 I_p\right)^{-1} k(\boldsymbol{\xi}). \tag{4.4}$$

Here $k(\boldsymbol{\xi}) \in \mathbb{R}^p$ is the covariance matrix between $\mathbf{z}$ and z_* obtained by $k_j(\boldsymbol{\xi}) = \mathcal{K}(\xi^{(j)}, \xi_*)$ and $\sigma^2 = \mathcal{K}(\xi_*, \xi_*) \in \mathbb{R}$ is the variance at ξ_*.

4.2.2 Discretized Gaussian process space

We investigate the field on a grid that is defined by the discretization of the space. Let

$$\mathcal{S}_c := [x_{\min}^{[1]},\ x_{\max}^{[1]}] \times [x_{\min}^{[2]},\ x_{\max}^{[2]}] \times [x_{\min}^{[3]},\ x_{\max}^{[3]}]$$

be the continuous spatial domain in $\mathbb{R}^3$, where the spatial limitations $\mathbf{x}_{\max}$ and $\mathbf{x}_{\min}$ are determined by coordinates of point clouds in the training data set. We discretized the 3-D continuous space into n spatial sites $\mathcal{S} := \{s^{[1]}, \cdots, s^{[n]}\} \in \mathbb{R}^{n\times 3}$, where $n = h(x_{\max}^{[1]} - x_{\min}^{[1]}) \times h(x_{\max}^{[2]} - x_{\min}^{[2]}) \times h(x_{\max}^{[3]} - x_{\min}^{[3]})$. h is chosen such that $n \in \mathbb{Z}_{>0}$. The collection of the realized value of the implicit surface on the lattice is denoted as $z := (z^{[1]}, \cdots, z^{[n]})^T$, where $z^{[i]} := z(s^{[i]})$.

The prior distribution of z is chosen such that $z \sim \mathcal{N}(\mathbf{1}, \Sigma_0)$, where $\mathbf{1}$ is a column vector of 1 and Σ_0 is the prior covariance matrix, i.e., $\Sigma_0^{[i,j]} := \mathcal{K}(z^{[i]}, z^{[j]})$. Recall that we take p noisy measurement with the model as described in (4.1). Using the GP regression, the posterior

distribution of z on the discrete lattice $\mathcal{S}$ is a Gaussian distribution, i.e., $z \sim \mathcal{N}(\boldsymbol{\mu}_*, \Sigma_*)$ where the posterior mean vector and the covariance matrix are defined as:

$$\boldsymbol{\mu}_* = K^T C^{-1} \tilde{z},\ \Sigma_* = \Sigma_0 - K^T C^{-1} K, \tag{4.5}$$

where $K := \mathcal{K}(\tilde{z}, z) \in \mathbb{R}^{p \times n}$ and $C := \mathcal{K}(\tilde{z}, \tilde{z}) \in \mathbb{R}^{p \times p}$.

In this study, we consider a discretized grid points $\mathcal{S} \subset \mathbb{R}^3$ on the 3-D space of the size $40 \times 40 \times 40$. The range of each dimension will be chosen such that the grid completely covers the point cloud data from all available scans of a patient.

4.2.3 Spatio-temporal Gaussian process

For patient i, we use the covariance function in the form of an exponential kernel function as follows.

$$\begin{aligned} \mathcal{K}(\mathbf{x}, \mathbf{x}', t, t') = \sigma_{f,i}^2 \exp\left(-\frac{|t - t'|^2}{2\sigma_{t,i}^2}\right) \\ \times \exp\left(-\frac{1}{2}\sum_{\rho=1}^{3} \frac{|x^{[\rho]} - x'^{[\rho]}|^2}{\sigma_{\rho,i}^2}\right), \end{aligned} \tag{4.6}$$

where $\mathbf{x} = [x^{[1]}\ x^{[2]}\ x^{[3]}]^T$ is the coordinates vector in the 3-D space. The hyper-parameter vector $\Phi_i := [\sigma_{f,i}\ \sigma_{t,i}\ \sigma_{1,i}\ \sigma_{2,i}\ \sigma_{3,i}]^T$ consists of the function bandwidth, time bandwidth and three spatial bandwidths, respectively. The hyper-parameters can be determined by maximizing the likelihood function. Notice that since the time bandwidth is obtained in a data-driven manner, the value of $\sigma_{t,i}$ could provide an insight of the evolution of the AAA with respect to time. For instance, smaller $\sigma_{t,i}$ implies that the aneurysm changes more rapidly and vice versa. The spatio-temporal Gaussian process regression provides a way to collect implicit surface field observations as shown in Fig. 4.1.

4.2.4 Implicit surfaces

In this section, we describe the definition of an implicit surface [24]. An implicit surface describes an object in a space by mapping coordinates of a location in the space onto a scalar value. The value will indicate if the location belongs to the surface of the object. In particular, let $\mathbf{x} \in \mathbb{R}^3$ be a coordinates vector in a 3-D space. Then, define a function $f : \mathbb{R}^3 \to \mathbb{R}$ as the *implicit surface* (IS):

$$f(\mathbf{x}) \begin{cases} = 0, \text{if } \mathbf{x} \text{ on the surface,} \\ > 0, \text{if } \mathbf{x} \text{ outside the object,} \\ < 0, \text{if } \mathbf{x} \text{ inside the object.} \end{cases} \tag{4.7}$$

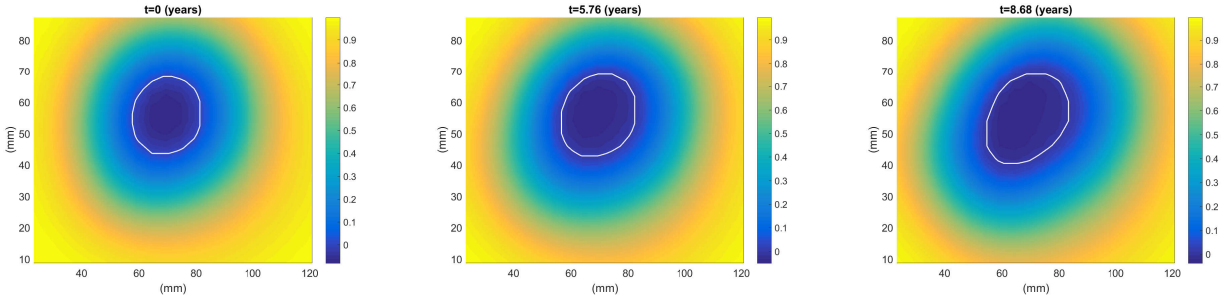


Figure 4.2 Example of an estimated field: cross section views of the 3D field at the same height in z-axis are shown at different times t. The on-surface points (where the field is zero) are labeled in white solid lines. The growth of the AAA in the radial direction can be visualized from top to bottom figures.

For example, Fig. 4.2 shows an estimated 3D field from patient P1 at three different times using the spatio-temporal Gaussian process regression. From top to bottom, the figures show cross section views of the field at the same height in z-axis at three sampling points: at the first scan, after 5.76 years, and after 8.68 years. Those points on the field that are outside of the surface have positive values (brightest yellow) and those that are completely inside the surface have values less than 0 (darkest blue). As shown in the figure, the radial growth of the AAA is reflexed by the spreading of the blue region as the time elapses. Note that a point cloud provides only the points that belong to the surface, i.e., have values of zero in the

IS field. Therefore, it implies *zero-value observations* [24] for the spatio-temporal Gaussian process regression discussed in Section 4.2.3. To implement the spatio-temporal Gaussian process, we utilize a number of built-in functions in the GPML package [90].

4.2.5 Dynamic implicit surface model

Let $f(\mathbf{x}, t)$ be the IS field that represents the aorta surface at the sampling time t with the data (CT scans) available for $t = \{1, \cdots, t\}$. Then, we can define the temporal growth model of the IS field as follows.

$$f(\mathbf{x}, t) = f(\mathbf{x}, t-1) + \Delta_t(A(\mathbf{x}) + W(\mathbf{x})) \tag{4.8}$$

where $A(\mathbf{x})$ is the growth rate of the IS field and $W(\mathbf{x})$ is the zero-mean process noise: $W(\mathbf{x}) \overset{i.i.d}{\sim} \mathcal{N}(\mathbf{0}, \Sigma_w)$. We assume the growth rate is time-invariant for the last several observations. Δ_t is the gap between two successive scan times, i.e., $\Delta_t := \tau_t - \tau_{t-1}$. Note that the uncertainty in the process increases linearly with the gap between observations, i.e., $\Delta_t W(\mathbf{x}) \sim \mathcal{N}(\mathbf{0}, \Delta_t^2 \Sigma_w)$. We assume that the initial field $f(\mathbf{x}, 0)$ has a normal distribution with mean μ_0 and covariance Σ_0, i.e., $f(\mathbf{x}, 0) \sim \mathcal{N}(\mu_0, \Sigma_0)$.

The observations of the IS field is modeled as the evolution of a spatio-temporal GP. The posterior distribution of the field can be treated in an observation model as follows.

$$y(\mathbf{x}, t) = f(\mathbf{x}, t) + V(\mathbf{x}, t), \tag{4.9}$$

where $V(\mathbf{x}, t)$ is the observation noise: $V(\mathbf{x}, t) \overset{i.i.d}{\sim} \mathcal{N}(\mathbf{0}, \Sigma_v(t))$. The noise covariance matrix $\Sigma_v(t)$ can be estimated as the posterior covariance of the GP. We utilize the Expectation-Maximization (E-M) algorithm to estimate $\Sigma_w(\mathbf{x})$ and $A(\mathbf{x})$. A detailed derivation for the Kalman Filter E-M algorithm is provided in Appendix 6.

In a nutshell, we define a likelihood function for the model conditioned on the data. We then derive the expectation of the likelihood function with respect to $A(\mathbf{x})$ and Σ_w, which

is denoted as $\Psi(A, \Sigma_w)$ (see Appendix 6). Hence, for each iteration r of the E-M algorithm the optimized values for $A(\mathbf{x})$ and Σ_w are the ones that maximize $\Psi(A, \Sigma_w)$, which can be found from the first derivative of $\Psi(A, \Sigma_w)$ (see Appendix 6). Since $\Psi(A, \Sigma_w)$ is a continuous function of A and Σ_w, by Theorem 2 in [120], $\Psi(A, \Sigma_w)$ will converge to a stationary value. Furthermore, since the first derivative of $\Psi(A, \Sigma_w)$ is continuous, (A, Σ_w) converges to a stationary point [120]. To illustrate this point, Fig. 4.3-(a) shows the evolution of the distance between $A(r)$ and its equilibrium value A_∞ in terms of the Frobenius norm, i.e., $\|A(r) - A_\infty\|_F$, converges to 0 after 10 iterations. A Similar evolution of Σ_w is shown in Fig. 4.3-(b).

The embedded surface distribution of the IS field is straightforward from the prediction step of the Kalman Filter as follows.

$$\begin{aligned} \mathbb{E}[f(\mathbf{x}, L)|\mathcal{D}_{1:L-1}] &= \mathbb{E}[f(\mathbf{x}, L-1)|\mathcal{D}_{1:L-1}] + \Delta_t A(\mathbf{x}), \\ \mathbb{C}\text{ov}(f(\mathbf{x}, L)|\mathcal{D}_{1:L-1}) &= \mathbb{C}\text{ov}(f(\mathbf{x}, L-1)|\mathcal{D}_{1:L-1}) + \Sigma_w \Delta_t^2. \end{aligned} \tag{4.10}$$

Note that one can alternatively avoid a dynamic model in (4.8) and directly apply a spatio-temporal Gaussian process with a kernel function described in (4.6) to obtain an inference of the field in future time [54]. However, there are two critical rationals that integrating observation regression and dynamic models outperforms the aforementioned method. First of all, since the posterior covariance of a Gaussian process is bounded by prior covariance Σ_0 in (4.5), the uncertainty quantification will eventually reach a limit as we increase the inference time Δ_t further. In contrast, our dynamic model does not impose any limit on the posterior covariance as the term Δ_t^2 is not bounded in (4.10), indicating the fact that as we try to infer at a future time, the uncertainty in the prediction will escalate accordingly. Secondly, naively applying the Gaussian process regression (as well as any other regression model) would overlook the growing trend of AAAs. This problem is resolved by incorporating the term $A(\mathbf{x})$ in the dynamic model (4.8).

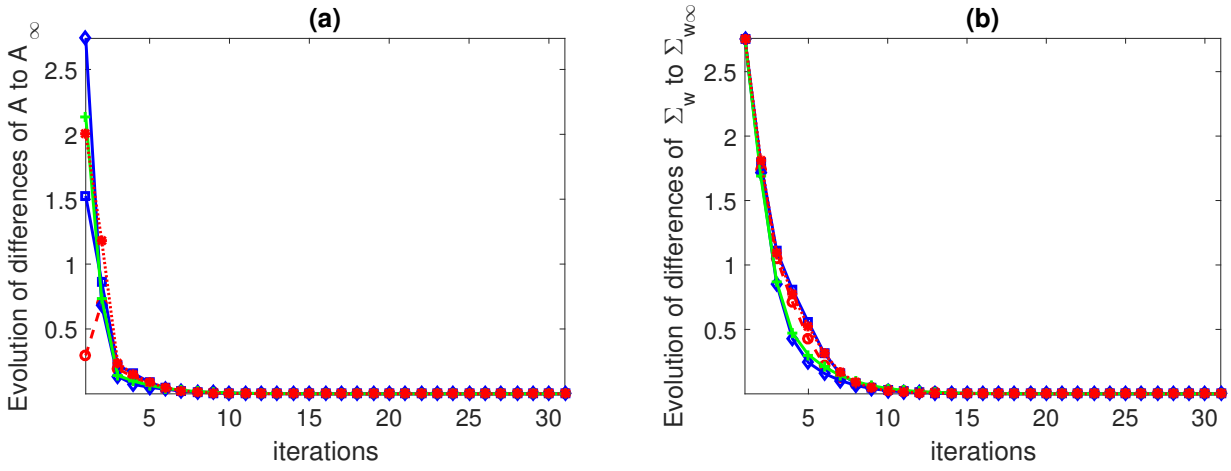


Figure 4.3 Convergence of the distance of (a) A and (b) Σ_w to their equilibrium values (A_∞ and $\Sigma_{w\infty}$), started with respect to different initial values of A and Σ_w.

There are two main computational challenges in our method. First of all, a drawback of the Gaussian process regression is computational complexity with respect to the number of measurements. It can be seen from (4.3) and (4.4) that the calculation of both the predictive mean and predictive variance requires the inversion of covariance matrices whose sizes depend on the number of observations p, i.e., complexity is $\mathcal{O}(p^3)$. Depending on the density and the number of point clouds, the size of points that are observed in our study can be exceptionally large. The second major source of computational consumption is the inversion of covariance matrix in the Kalman Filter algorithm. The Kalman Filter covariance matrix has a dimension of $n \times n$ with n is the number of spatial sites that is discussed in Section 4.2.2. Therefore, the inversion has complexity of $\mathcal{O}(n^3)$. In this study, we use $n = 64,000$ and for those patients with large training data set (more than 4 data points), the E-M algorithm is computationally prohibitive. To resolve these problems, we divide the dense spatial grid $\mathcal{S}$ into sparser sub-grids and apply our method in a distributed manner. The final outcome on $\mathcal{S}$ is merged from predictive results on individual sub-grids. The detail of the efficient Kalman Filter algorithm is reported in Appendix A.

4.3 Surface extraction and post-processing

In the previous section, we have obtained the statistics of predicted IS field that is provided in (4.10). In this section, we briefly show how to extract the predicted surface from the predicted IS field.

Note that the predicted surface is an embedded manifold of the predicted IS field that satisfies (4.7). However, due to the error in computation and uncertainty in the Gaussian process regression, achieving exact zero values of the IS field for on-surface points tuned out to be not practical from our preliminary study. Therefore, we relax the restricted equality condition in (4.7) by setting a near-zero threshold of the field under which the corresponding point is considered to be on-surface. The threshold of the training phase is determined by minimizing the Euclidian distance between the estimated and the true point clouds (see Appendix B). To find the threshold of the test phase, we utilize a fact that each particular value of the threshold classifies the whole 3D lattice $\mathcal{S}$ into a particular spatial binary pattern of two categories: on-surface and off-surface. Therefore, we assign the test phase threshold to the value that yields the most similar spatial pattern to the train phase threshold (see Appendix B).

4.4 Experimental results

In this section, we arrange an experimental study to illustrate the effectiveness of our approach in realistic scenarios using the reconstructed point clouds $\mathcal{D}_{(id,scans)}$ that are adapted from [34]. To validate our approach, the last available data point for each patient $\mathcal{D}_{(id,L)}$ is used as the ground truth whereas the previous data points $\{\mathcal{D}_{(id,j)} : j = \{1, \cdots, L-1\}\}$ of that patient are used to train the model. Then, we compare the prediction results with the existing true data set $\mathcal{D}_{(id,L)}$ for each of the 7 patients using the Hausdorff distance [53]. Thus, we have 7 different patient specific cases with different longitudinal lengths and morphological properties along with inter-scan time interval. For example, we use the last 6

data points of patient H, $\{\mathcal{D}_{H,1}, \cdots, \mathcal{D}_{H,6}\}$, for training, and predict the most current state of the aneurysm, $\hat{\mathcal{D}}_{H,7}$.

The Hausdorff distance is a quantitative measure between two point clouds. However, it does not provides the insight of the spatial shape. In order to obtain a visual estimation of the prediction, we utilize the the Poisson Reconstruction function in MeshLab (National Research Council, Rome, Italy) to render the 3-D structure of the surfaces. The results are shown in Fig. 4.5. Each pair of estimated and true scans are labeled as $\mathcal{D}_{i,j}$ and $\hat{\mathcal{D}}_{i,j}$, respectively.

Patient ID	Number of Scans	Hyperparameters $\Phi_i := [\sigma_{t,i}\ \sigma_{1,i}\ \sigma_{2,i}\ \sigma_{3,i}]$	σ_t (days)	Hausdorff Distance (Full)	(Last-3)
P1	7	$[722.96, 13.69, 12.82, 29.75]$	722.96	17.86	16.16
P2	6	$[1085.38, 7.02, 5.89, 12.47]$	1085.38	11.28	11.53
P3	5	$[683.49, 3.07, 3.15, 7.41]$	683.49	8.32	6.26
P4	5	$[835.32, 5.52, 5.04, 11.85]$	835.32	9.8	12.86
P5	4	$[475.18, 6.09, 5.63, 13.28]$	475.18	9.85	-
P6	6	$[597.75, 2.05, 2.06, 4.91]$	597.75	6.68	6.40
P7	4	$[553.98, 2.40, 2.43, 5.39]$	553.98	8.32	-

Table 4.2 Hyperparameters and Hausdorff Distance with details of each Case

Table 4.2 shows the case IDs (column 1), patient IDs (column 2), number of available scans (column 3), estimated hyper-parameter vectors (column 4), estimated time bandwidth σ_t (column 5), and the Hausdorff distances (column 6 and 7).

For each case, the Hausdorff distance between the predicted and estimated point clouds is calculated. The mean and standard deviation of the overall Hausdorff distances are 10.30 (mm) and 3.64 (mm). The small standard deviation implies high precision of our approach.

4.4.1 Constant growth rate assumption

In what follows, we investigate the effect of the assumption on the constant growth rate $A(\mathbf{x})$ in (4.8).

We observe that increasing number of scans in the training data results in decreasing prediction errors, but only up to a certain point. Then, the error starts to increase. This

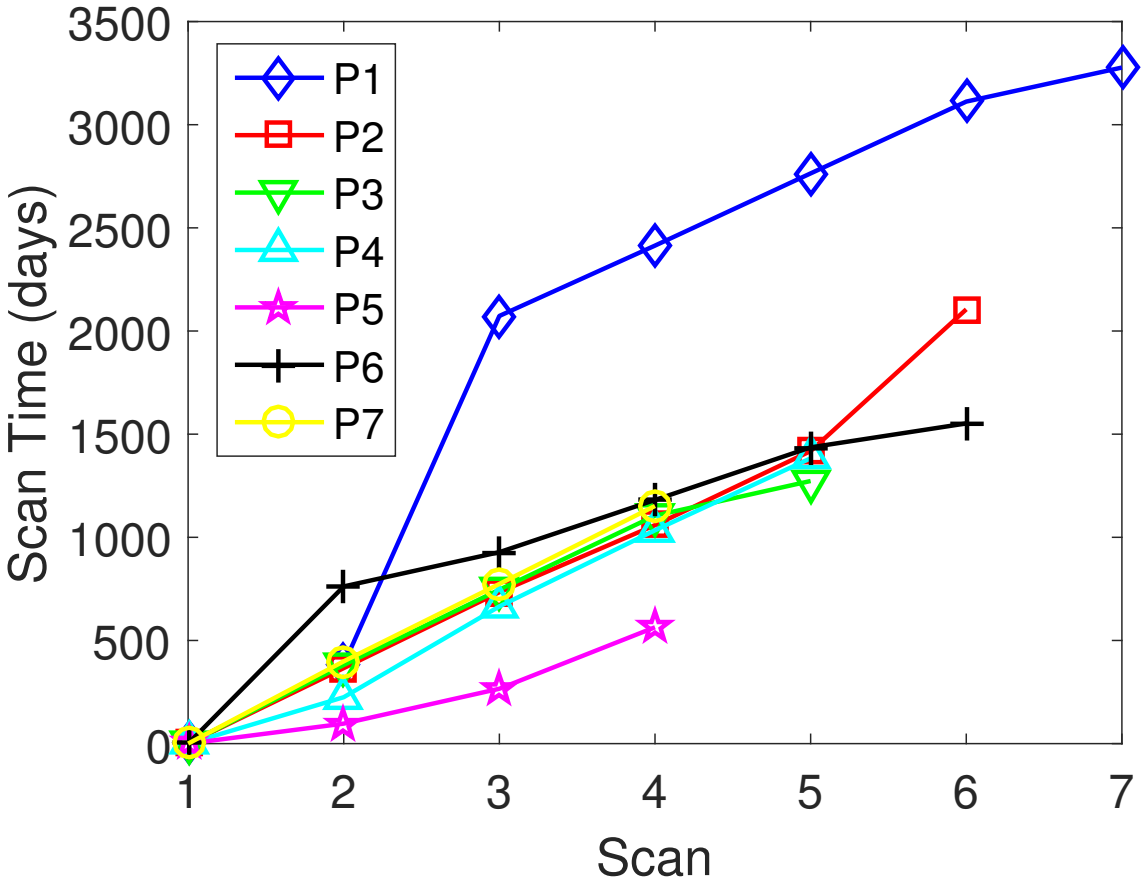


Figure 4.4 The relative times of the scans compared to the first scan are plotted for each patient in days.

is contrary to the common expectation that larger amount of training data yields lower errors. A similar investigation is reported in [107]: the number of measurements does not apparently effect the precision of the predicted results. This observation may be explained by an abrupt increase in the enlargement rate of the aneurysm caused by over-stretched damage of elastin [118]. To test this hypothesis, we conduct a separate study in which we limit the training data set to the *three* most recent data points. In Table 4.2, the results of this study are referred to as "Last-3" whereas, the label "Full" is used when all the available data pionts are used for training. The predicted surfaces and confidence regions are shown in Fig. 4.7 and Fig. 4.8, respectively. The Hausdorff distances in the "Last-3" case provide similar results for those patients with small aneurysms. In contrast, the "Full" case yields decreasing distances for those patients who have larger aneurysms.

4.4.2 Uncertainty qualification

Our approach provides quantification of uncertainty in the prediction, i.e., confidence regions. In order to compute the confidence regions, we sample the IS fields by realizing the posteriori Gaussian distribution with the updated mean vector and covariance matrix computed by (4.10) in a Monte Carlo simulation with 100 runs. We then apply the same procedure as described in Section 4.3 to estimate the AAA surfaces from the sampled IS fields. With a sufficient number of realizations, the set of surfaces forms the confidence region of the prediction. In Fig. 4.6, we visualize the confidence region as follows. First, we set the transparency of 2% for each realized AAA surface in the surface rendering process. We then overlap them in space. Therefore, spatial regions that appear brighter are more likely to be occupied by the AAA surface. The true point clouds are plotted in red dots.

4.5 Discussion

In this section, we discuss the results, the possible utilities and limitations of our approach along with future research directions.

Patient P6 has a high number of scans with regular scanning times and shows to have the lowest Hausdorff distance, i.e., the highest accuracy. Patient P3 yields the second lowest Hausdorff distance. Despite having the same number of scans as patient P6, patient P2 shows a remarkably higher error in prediction. This could be due to an unprecedented longer time (687 days) between the latest adjacent data points for patient P2, as compared to the latest adjacent data points of P6 (116 days) as shown in Fig. 4.4.

We compare our results with those in Powells et al. [11,26] whom use linear and quadratic hierarchical growth models predicting the future size of aneurysms. The hierarchical linear growth model utilizes a zero-mean Gaussian distributed random-effects term to simulate the growing effects of aneurysms. This approach, however, overlooks increasing growth, and it was reported in some cases that the models predict AAA diameter to decrease in a future time [107] which is not realistic. In contrast, the growing effect in our model is obtained by patient-specific fitting based on longitudinal data. Secondly, the hierarchical model focuses only on AAA maximal diameters and does not provide the geometrical shape analysis. On the other hand, our approach addresses a complete treatment of AAA geometrical shape prediction.

The implicit surface is similar to the well-known level set [130] (also known as implicit contour) method in the sense that both methods utilize the evolution of an implicit scalar function to simulate the changes of the interface where the scalar function is zero. However, the level set method relies on the prior (observer's) knowledge in an image such as image intensity, or anatomical model, as a driving force to evolve the function and propagate the interface. Thus, it has not been used so far as a predictive model as proposed.

Compared to the outcomes of other previous works, our results yield following innovative merits. First of all, there has been research effort to predict aneurysm growth and its

uncertainty merely based on a data-driven statistical framework such as non-linear regression [54]. However, naively applying such regression models overlooks a critical fact about AAAs that is their unrelenting increases in diameters [68] by neglecting the deterministic growth dynamics. In this study, we propose a linear dynamic model that simulates an AAA's growth, which will be calibrated by a specific patient's data. Secondly, our method provides a complete prediction of AAA's geometrical structure, unlike recent methods that reduce the data dimensions such as centerline parameterization based approaches [34]. Moreover, each type of parameterizations is associated with a particular topology, which significantly restricts its versatility [88]. To overcome this disadvantage, we develop our model to be free of geometrical parameterization.

4.5.1 Decision making via prediction and confidence regions

The major possible utility of our algorithms is in helping clinicians in conducting medical treatment of an AAA, e.g., monitoring, open surgery or endovascular repair, by providing access to a predicted AAA at a future time. Moreover, the clinicians will also have access to the confidence region of the prediction. This will help them in making a more informed decision for the treatment based on the prediction.

We can observe from Fig. 4.6 that the confidence region covers the true AAA surface better for the case with lower Hausdorff distance. Thus, clinicians can gauge the reliability of the predicted AAA. Note that standard G&R computational models do not provide the uncertainty in their prediction [38, 100, 128]. Therefore, our model shows an significant improvement over them.

4.5.2 Estimated hyper-parameters and model parameters as informative features

Note that besides the final prediction, the hyper-parameters estimated by the spatio-temporal Gaussian process, as shown in Table 4.2, also may provide insightful information. The spatial bandwidths $\{\sigma_{1,i}, \sigma_{2,i}, \sigma_{3,i}\}$ in (4.6) provide information of spatial variation of an AAA's 3D structure. For example, a high value for $\sigma_{1,i}$ implies that the AAA surface varies smoothly whereas a lower value indicates that the surface has high variance in the direction of the first axis. The estimated spatial bandwidths by maximizing the likelihood function are shown in column 4 in Table 4.2. Notice that the estimated spatial bandwidths for the direction of z-axis, i.e., $\sigma_{3,i}$ are normally larger than those for the directions of x and y axes. Since the AAA has a tubular structure, there are less spatial variation along the z-axis compared to the other two coordinates. Additionally, the time bandwidth σ_t represents the temporal changes. Smaller σ_t implies that the AAA shape is likely to change more rapidly with respect to time.

The estimated hyper-parameters and linear model parameters may be viewed as features that may encode the information about the evolution of an AAA. The hyper-parameters estimated for the regression provide a unique patient-specific feature vector which may capture both the temporal and spatial variation patterns of the AAA surface. Collective feature vectors obtained from more patients could be useful in building a classification module capable of detecting patients with imminent danger of rupture [91]. In addition, the estimated linear growth rate $A(\mathbf{x})$ provides information of the migration of the surface in a 3D space. In this work, we utilize the value of $A(\mathbf{x})$ merely for updating the IS field. However, if we view $A(\mathbf{x})$ as a scalar field as shown in Fig .4.9, its partial derivatives with respect to spatial and time may provide an insightful information about the moving surface. This analysis can be obtained by applying a technique that is similar to Lucas-Kanade (LK) optical flow [10] as follows. Assume that we would like to compute the velocity of the movement of the on-surface points $\mathsf{x}(t)$ over time t, i.e., $\frac{d\mathsf{x}}{dt}$. Since on-surface points have zero values of IS field,

we have $f(\mathsf{x}(t), t) = 0$, hence $\frac{\partial f(\mathsf{x}(t),t)}{\partial t} = 0$. Applying the chain rule:

$$\frac{\partial f(\mathsf{x}, t)}{\partial \mathsf{x}} \left(\frac{d\mathsf{x}}{dt}\right) + \frac{\partial f(\mathsf{x}, t)}{\partial t} = 0 \tag{4.11}$$

Note that $\frac{\partial f(\mathsf{x},t)}{\partial \mathsf{x}}$ is the spatial derivative of the IS field at time t and can be estimated by $\frac{\partial \mathbb{E}(f(\mathsf{x},t)|\mathcal{D}_{1:t})}{\partial \mathsf{x}}$ with $\mathbb{E}(f(\mathsf{x}, t)|\mathcal{D}_{1:t})$ given in (4.10). Furthermore, $\frac{\partial f(\mathsf{x},t)}{\partial t}$ is the derivative of the field with respect to time and can be approximated as $A(\mathsf{x})$. Thus, from (4.11), we will be able to compute the velocity of the movement of the on-surface points as:

$$\frac{d\mathsf{x}}{dt} = -\frac{A(\mathsf{x})}{\frac{\partial f(\mathsf{x},t)}{\partial \mathsf{x}}}.$$

4.5.3 Limitations and future research directions

The method is based on an empirical Bayesian method, hyper-parameters are obtained *a-priori* by maximizing the likelihood function [90] for the IS field observations. Therefore, the uncertainties associated with the hyper-parameters are not taken into account. In contrast to the empirical Bayesian framework, the fully Bayesian approach marginalizes the uncertainties in the hyper-parameters with increased complexity. Hence, one future research direction is to develop a fully Bayesian version of our proposed scheme taking into account uncertainties in hyper-parameters [22].

Therefore, as a future work we investigate the implication of using the constant growth rate $A(\mathbf{x})$ in Section 5.2, which could be a limitation. The time-varying or switching $A(\mathbf{x})$ can be explained as a result of mechanical damage of elastin that causes the growth rate to suddenly escalate. However, our dynamic model assumes that with three most recent scans, the growth rate $A(\mathbf{x})$ is constant. Thus, a disruptive growth due to elastin damage is not captured by our dynamic model. In other words, for those cases that experience the abrupt increment, the model with constant $A(\mathbf{x})$ trained on the whole training data, underestimates the growth rate after the point of elastin degradation. Therefore, the linear dynamic model in

(4.8) can be improved by estimating two growth rates, e.g., $A_1(\mathbf{x})$ and $A_2(\mathbf{x})$ and a switching time τ_s to cope with the switched growth rate.

We have shown that our model is able to provide a reliable prediction of AAA evolution, without including the G&R computational model [121, 127]. However, in order to extend it beyond surface prediction to predict the rupture potential (mechanical stress, the effect of thrombus, etc.), we have to combine our approach with the G&R computational model so that both shape evolution and mechanical stress are combined to estimate the rupture potential. The combination of our approach and the G&R computational model will be a computationally and theoretically challenging task given the computational complexity of the model and its unknown parameters. However, a biomechanics-based computation model structure will provide a constraint in space and time, which will help in reducing the size of the confidence region of the predicted AAA at future time. Therefore, our future research would focus on the incorporation of the computational G&R model in our Bayesian framework.

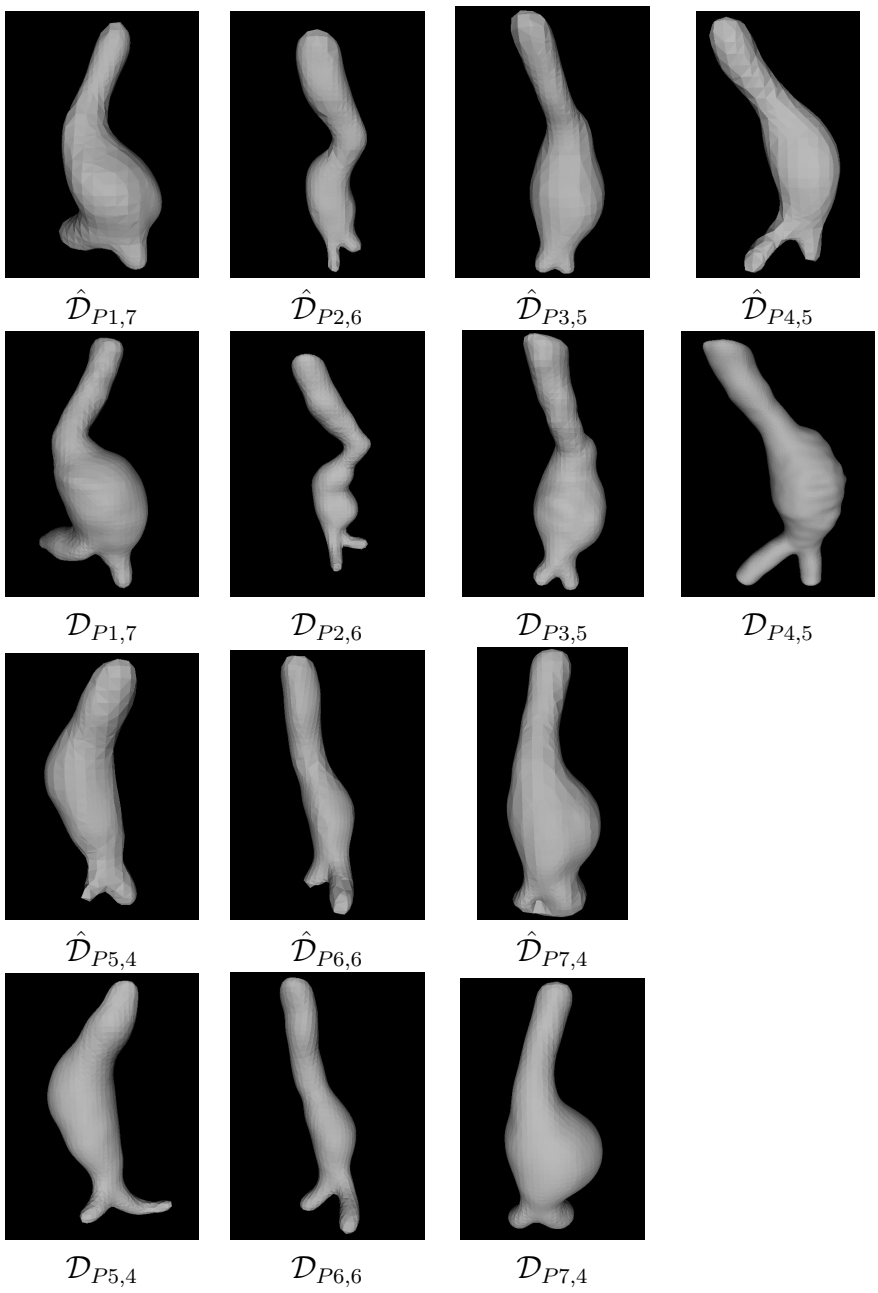


Figure 4.5 Fully trained case using all available longitudinal data: 3-D rendering from the estimated ($\hat{\mathcal{D}}_{i,j}$) and true ($\mathcal{D}_{i,j}$) point clouds: The predicted and true point clouds are used to render the 3-D surfaces using the Poisson reconstruction in Meshlab.

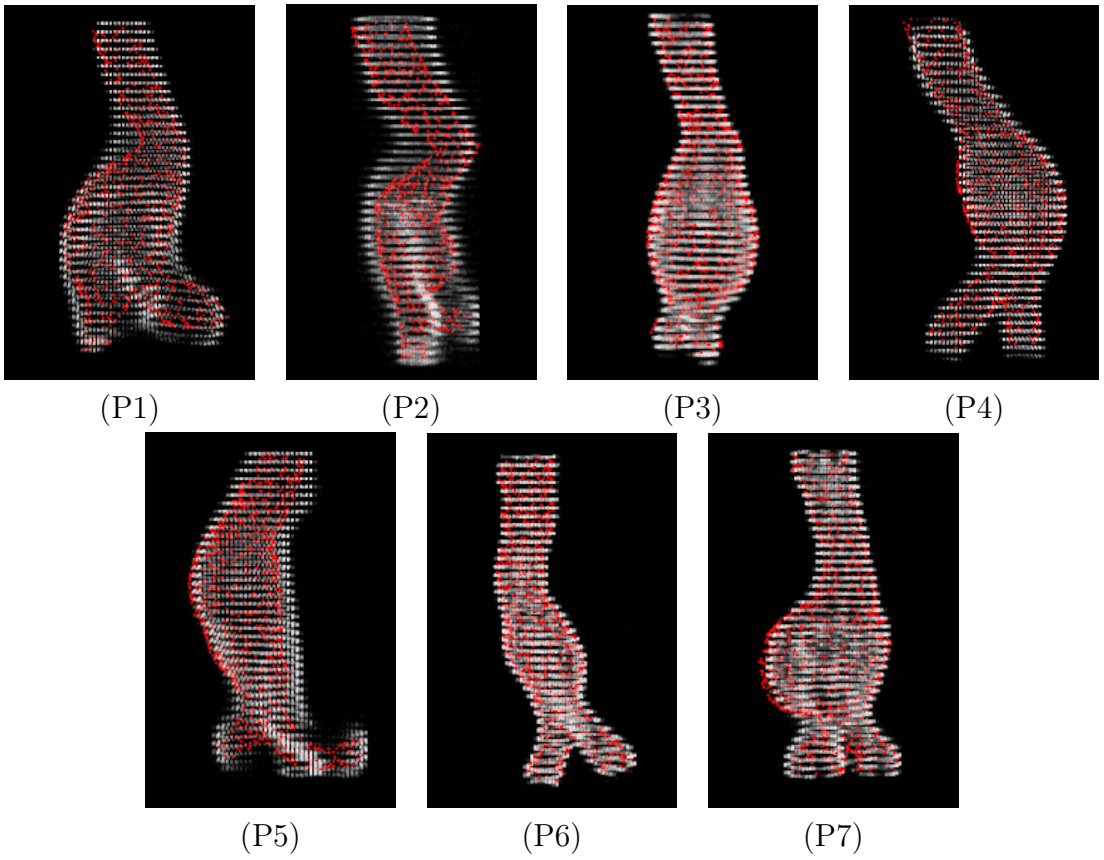


Figure 4.6 Uncertainty quantification for 7 patients for *fully trained* case: The implicit surfaces are regenerated on the grid by realizing the multi-variate Gaussian distribution with the mean vector and covariance matrix computed in (4.5). Then, we apply the same procedure described in Section 4.3 to estimate the AAA surfaces that are corresponding to the realized implicit surfaces. Each realization of the AAA surface from the posterior distribution is plotted with 2% occupancy. Therefore, spatial regions that appear brighter are more likely to be occupied by the AAA surface. The true cloud points are plotted in red dots.

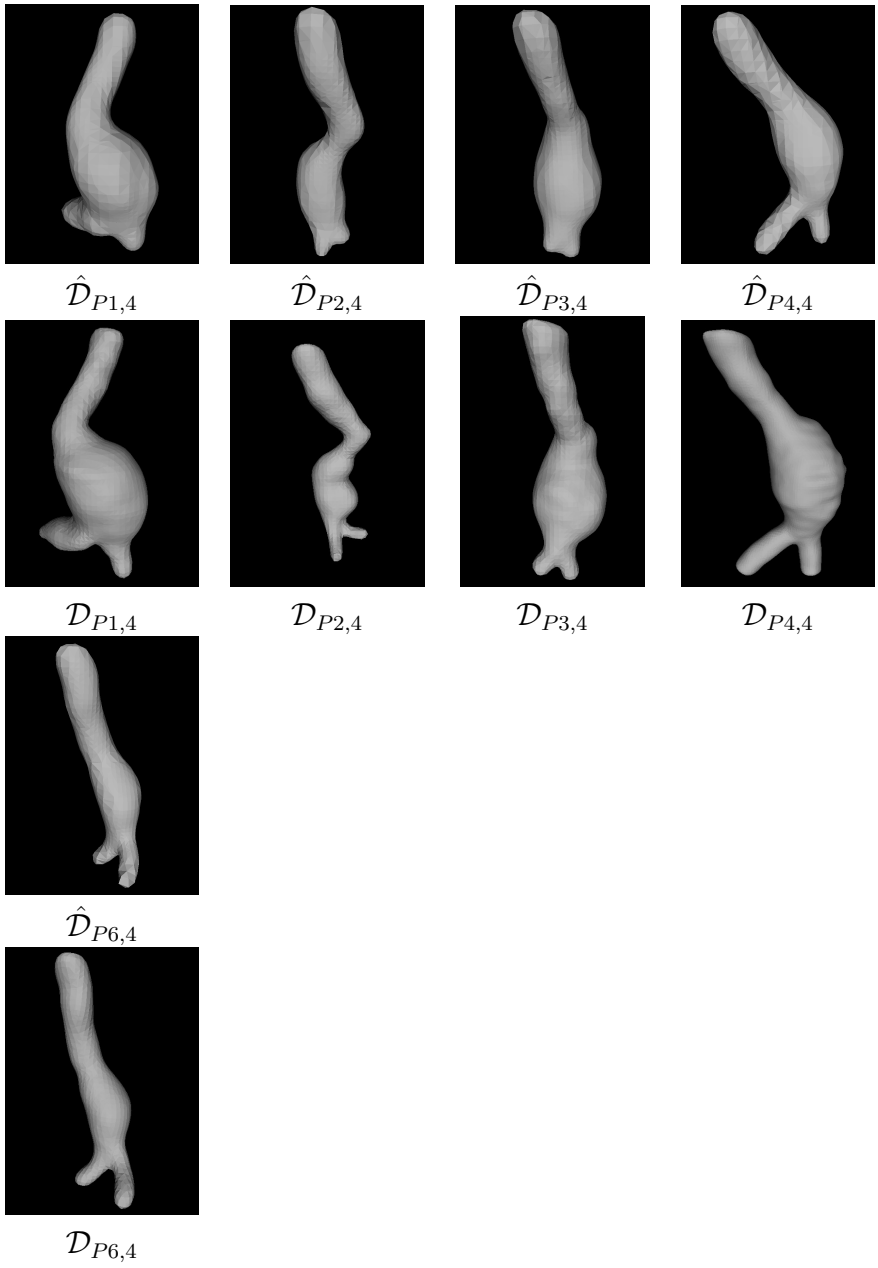
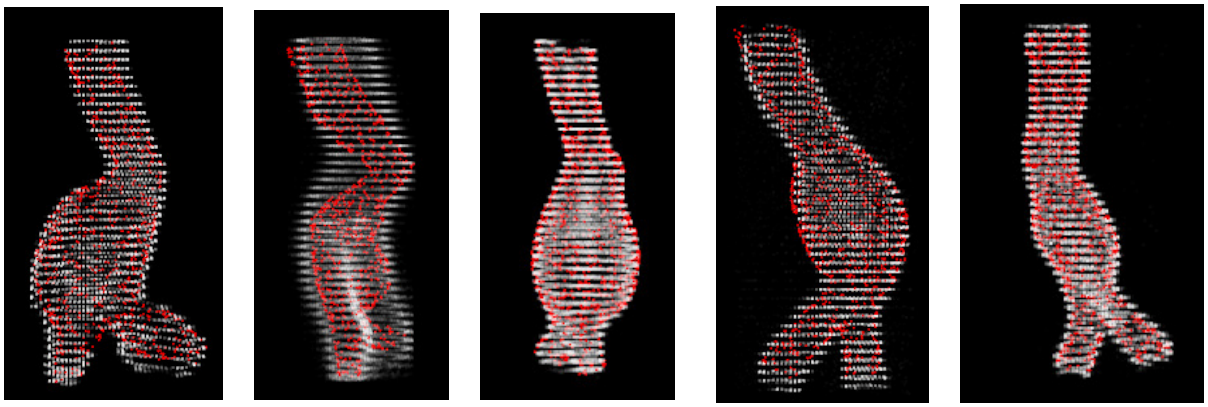


Figure 4.7 Last-3 trained case using the most recent three longitudinal data points: 3-D rendering from the estimated ($\hat{\mathcal{D}}_{i,j}$) and true ($\mathcal{D}_{i,j}$) point clouds: The predicted and true point clouds are used to render the 3-D surfaces using the Poisson reconstruction in Meshlab. Note that we run the Last-3 trained case for patient P1, P2, P3, P4 and P6, since other cases already have 4 scans in total.



(P1) (P2) (P3) (P4) (P6)

Figure 4.8 Uncertainty quantification for 5 patients for *Last-3 trained* case: The implicit surfaces are regenerated on the grid by realizing the multi-variate Gaussian distribution with the mean vector and covariance matrix computed in (4.5). Then, we apply the same procedure described in Section 4.3 to estimate the AAA surfaces that are corresponding to the realized implicit surfaces. Each realization of the AAA surface from the posterior distribution is plotted with 2% occupancy. Therefore, spatial regions that appear brighter are more likely to be occupied by the AAA surface. The true cloud points are plotted in red dots. Note that we run the Last-3 trained case for patient P1, P2, P3, P4 and P6, since other cases already have 4 scans in total.

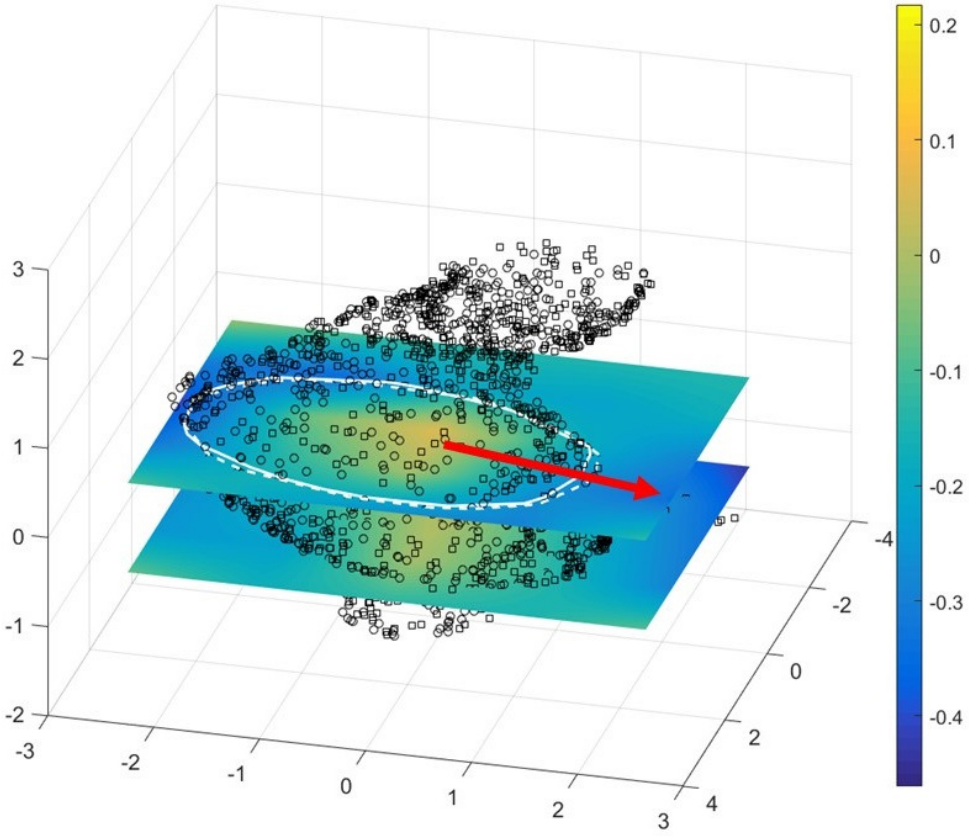


Figure 4.9 An example of usage of $A(\mathbf{x})$ as an informative feature. Two successively shapes, namely previous and current shapes, are plotted in black squared and circled point clouds in standardized unit, respectively. Additionally, the intersections of them with the plane $z = 0$ are plotted in solid (previous) and dashed (current) white lines on the $z = 0$ plane. Furthermore, the cross-section views of the $A(\mathbf{x})$ field at $z = 0$ and $z = -1$ are color plotted on the two planes. The migration of the surface is shown clearly in the direction indicated by the red arrow. Then, the velocity of the migration in the indicated direction can be computed by (4.11).

Chapter 5

Abdominal Aortic Aneurysm's maximum diameters prediction

As discussed in chapter 4, maximum diameter is an imperative criterion to determine for surgical planing of AAAs. One common way to monitor the maximum diameter is the maximum diameter curve (MDC), which is the collection of maximum diameters along the centerline of an AAA. To be considered as an aneurysm, a relative criterion can be used such that the enlargement of the aorta is greater than 50% of the normal diameter. On the other hand, an absolute criterion can also be used. For example, in the infraneral aorta, which is the region lies between the renal branches and the iliac bifurcation, a diameter greater than 3 cm is considered as an AAA.

Due to the severe effects and high mortality rate of the disease, massive databases of both heathy and positively diagnosed have been collected over the decades. While there have been large-scale population-based screening studies, such as Multicenter Aneurysm Screening Study (MASS) in the UK [108], analysis models that is based on longitudinal data has not been properly investigated. In contrast to population-based models, when longitudinal patient-specific analysis models are carefully calibrated based on individual cases those are be able to aid physicians in detecting aneurysms and making decisions. For clinical

treatments and recommendations, a patient-specific predictive tool is required to incorporate the advances in computational modeling and computational capacity. The development of such tool requires a major paradigm-shift since clinical measurements are associated with limited information, uncertainty and incompleteness of the model. In this study, we adopt deep learning to tackle the problem.

Deep learning and deep architectures in general have been transforming an enormous number of research areas, majority in computer vision [77] and natural language processing [15]. Furthermore, deep learning also has been heavily applied in risk prediction based on electronic health record [13], image labeling [44], traffic flow prediction [52], image segmentation [74], medical image segmentation [69], and many other fields. Deep architecture has been investigated since 1980 [32] and has been proved to be more effective and requires less resource compared to a shallow structure of the same size, i.e., same number of nodes. The merits of deep structure come from its ability to reduce redundant works by distributing the tasks through its layers [69]. For instance, the low layers can perform low level tasks like gradients computation or edge detection while the higher layers can perform classification or regression. However, as the networks are constructed in deeper layers, the training becomes prohibitively slow due to the problem of "vanishing gradients" [3]. In particular, when the error is back-propagated from the output layer, it is multiplied by the derivatives of activation function, which is near zero for those saturation nodes. Consequently, the error as the driven force for the gradient decent algorithm is dramatically dissipated that results in extremely slow training rate for those nodes behind the saturated node.

The training problem had remained until 2007 when Hinton proposed a two-stage learning scheme [49]. Firstly, the network is trained in an unsupervised and layer-wise manner in the form of a restricted Boltzmann Machine (RBM), i.e., pre-training. Then, the network is trained again with labeled data, i.e., fine-tuning. However, it is shown that the deep structure only yields high level of generalization and low test error when it is trained on a large available training set. For instance, LeCun et al. [71] utilize a MNIST [72] dataset of

hand-written numbers with $60,000$ samples to train a 3-layer deep network. Unfortunately, in most of cases such large data set of AAA is not available or time-consuming manual segmentation is required. The similar situation exists for other types of medical data set. Thus, up to date, the applications of deep structure in medical data are limited to medical image segmentation.

In this study, we investigate the use of a Deep Belief Network (DBN) to solve the problem of prediction of MDC at a future time in a regression framework. One of the main obstacle to apply deep structure to such a biomedical problem is the shortage of training data set due to variety of reasons such as high cost of data acquisition or disruption in patients' visitings. To cope with the problem of small labeled data set, we propose a method to use a small size data and generate a large artificial data that requires a short time and less computational resource. To the best of our knowledge, this is the first effort to adapt a deep structure in the problem of prediction of AAA.

In Section 5.1, we describe the types of data that our model bases on. In Section 5.3, we discuss how we use the PCM model to general our artificial data set. In Section 5.4, we discuss our deep network structure and the overall prediction model. The results are shown in Section 5.5.

5.1 Data

5.1.1 Real patient based reconstructed data

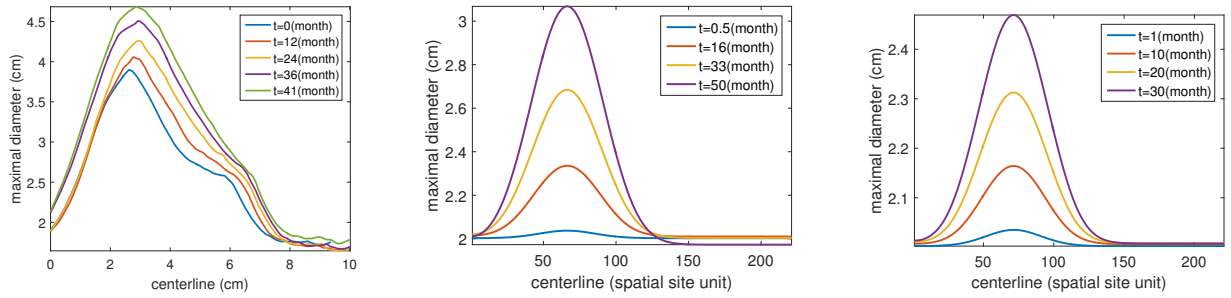
While there are various ways to measure the maximum diameter of an AAA [66], we utilize the results from the work of Gharahi et al. [34]. The method starts with a 3D point cloud of the AAA wall, then a maximally inscribed sphere is defined as the largest sphere within the outer arterial wall surface. The inscribed sphere is moved from the bottom to the top of an AAA. Then, the trace of the movement of such sphere's center point forms the centerline. Finally, the maximum diameters are plotted versus the centerline as shown in Figure 5.1-(a).

Patient ID	Number of Scans	Gender	Age	Time of Scans (Years)
P1	7	Male	68	[0, 1.07, 5.76, 6.70, 7.68, 8.64, 9.10]
P2	6	Male	66	[0, 1.02, 2.04, 2.94, 3.94, 5.85]
P3	5	Male	54	[0, 1.06, 2.07, 3.07, 3.54]
P4	4	Male	73	[0, 0.27, 0.74, 1.57]
P5	6	Male	70	[0, 2.12, 2.58, 3.28, 3.99, 4.31]
P6	4	Male	54	[0, 1.11, 2.15, 3.20]

Table 5.1 Demographic Data of Patients adapted from [34]

We adapt the collection of maximum diameter curves from the authors of [34] as use them as the input to our method.

According to [34], the longitudinal data is collected from 6 patients in total, the demography of which is shown again in Table 5.1 in this dissertation.



(a) (b) (c)

Figure 5.1 Example of 2D profile curves of maximal diameters over the centerline obtained from (a) real patient, (b) the G&R computational model, and (c) the PCM approximation. Note that the two G&R and PCM curves in this example are not based on the real centerline, so their units are not in centimeters, but in spatial site unit. The spatial site can be seen as a 1D mesh in the FEM code.

5.1.2 Artificially generated data

G&R computational model: The G&R computational model [27] is a finite element method (FEM) based vascular model that simulates the mechanical and geometrical state of an aortic aneurysm at a given time. The G&R model utilizes individual microstructural properties of different constituents to compute the stress-stretch state. In the G&R model, the aorta

is consist of three stress-bearing constituents: elastin, collagen fiber families, and vasoactive smooth muscle cells. Each constituent has its own properties and contribution to the strength of the artery's wall. The G&R model connects the constituents to the stress-stretch state of the artery and calculates the evolution of those microstructural properties with a stress-mediated feedback approach.

Elastin contributes resilience and elasticity to the aortic tissue; however, it degenerates over time and be irreplaceable. The degeneration in elastin causes a localized dilation of the aorta, leading to the weakening of the wall. Consequently, it results in an increase of the diameter and the wall stress of the aneurysm. The degeneration of the elastin is specified by the elastin damage function is defined as a modified Gaussian function:

$$d(s) = k_d \exp\left[-\frac{(s-\mu_d)^\alpha}{2\sigma_d^2}\right], \tag{5.1}$$

where s is the coordinate defined on the centerline. μ_d and α are estimated for each patient based on their data as they have specific effects on the shape of the damage function; therefore, on the stress-stretch and geometrical state of the AAA. Since the maximum aortic diameter locates at a close proximity of the maximum damage, μ_d is most likely to be in the vicinity of the centerline locations where the local enlargement is the most severe. We assign the values of 2 or 4 to α, which is determined by minimizing the error between the simulation and the real data. Additionally, k_d is a scaling factor, i.e., $k_d \in [0, 1)$, such that as k_d approaches 1 the degradation of elastin increases leading to the dilatation of the artery. On the other hand, $k_d = 0$ means no degradation of the elastin, which results in the retain of the artery. σ_d is the standard deviation of the Gaussian function; hence, it defines the width of the 2D profile curve of the aneurysm. These three parameters k_d, σ_d, and μ_d directly affect the time evolution of the aneurysm; thus, each unique group of the three parameters yields an unique outcome of the G&R code. One example is shown in Figure 5.1-(b).

PCM approximation model: One common disadvantage of the G&R model is that it is

extremely time- and computational resource- consuming. Therefore, to generate a large data set for the deep network, it is not the optimal option. Alternatively, we use a small number of G&R model simulations to train the PCM and generate a large artificial data set using the PCM. The detail of the PCM is discussed in Section 5.3 and an example is shown in Figure 5.1-(c).

5.2 Prediction of maximum diameter curve

Assume that we have an AAA evolving and for every one year, we are able to obtain an MDC from that particular AAA. Let $f_{t,i}$ be one MDC of the AAA i at scan time t and a collection of $f_{t,i}$ for a span of t provides us the timeline evolution of the AAA i. We define our regression problem as follows. For each data point, we determine the feature vector x_i to be the collection of *three* most recent MDCs, and the prediction target y_i is the MDC for the next scan in a future time:

$$x_i = [f_{t-2,i}, f_{t-1,i}, f_{t,i}],$$

$$y_i = f_{t+1,i}.$$

In the G&R code, the centerline is discretized into a grid size of 221, so the dimensions of our data and label are 663 and 221, respectively.

Given a set of parameter $\gamma = [k_d, \sigma_d, \mu_d]$ and a time span, the G&R model (as well as the PCM approximation code) will produce a time series of MDCs to simulate the growth of AAA during the time span, i.e., an artificially generated collection of $\{x_i, y_i\}$. Next, we normalize each data point and save the normalization scale as a separated data set. Then, the normalization scale data set is used to predict the scale of the future MDC by utilizing the Gaussian process regression. It has been shown that it is difficult to learn the variance for each visible unit [47]. Thus, by normalization process, we transform the original data to be the new one with zero mean and unit variance. Additionally, we normalize the data

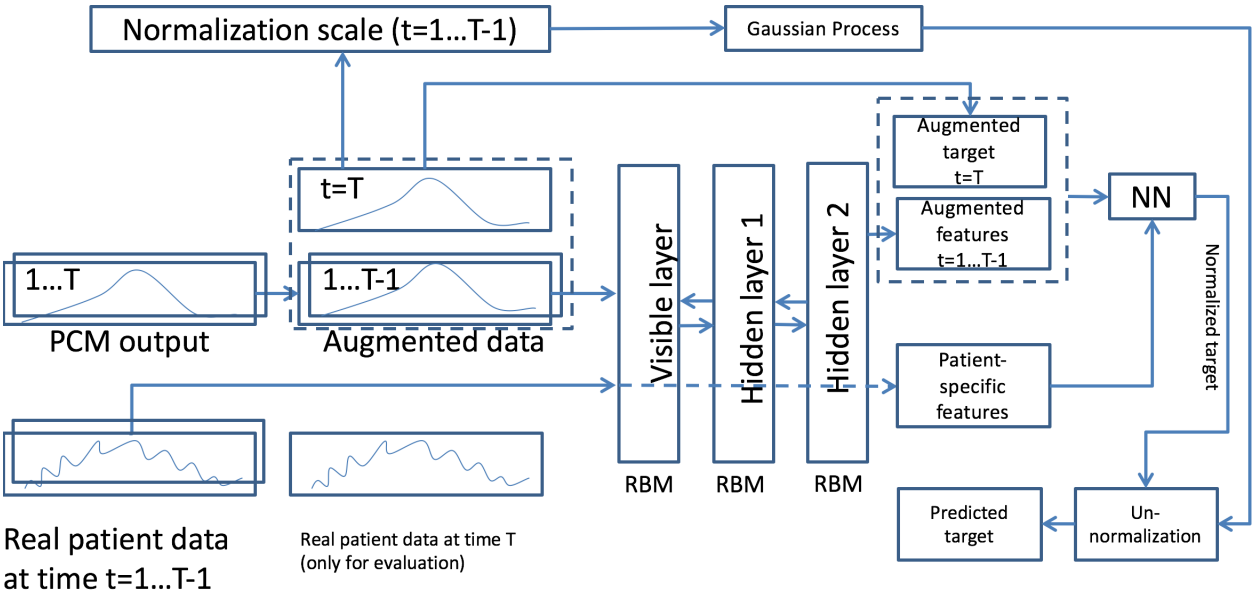


Figure 5.2 Overall diagram of the proposed method.

since the output regression layer utilizes the logistic function, so the outcome is in the form of probability, i.e., in the range [0, 1]. Then, the normalized data is plugged into a Deep Belief Network with a neural network (NN) regression as the output layer. After being pre-trained with the normalized artificial data, the NN is fine-tuned again with the real data set by back-propagation. Finally, the predicted MDC is then transformed back to the normal scale to be the final prediction outcome. The overall method is depicted in Figure 5.2. To compare two MDCs, we use the standard Root Mean Squared Error (RMSE).

Prediction of normalization scale: as aforementioned, we utilize Gaussian process regression to predict the normalization scale of the future scan. The detail of Gaussian process is discussed in Section 4.2.1. We briefly describe our model here. For a data point i, let

$$u_i = [\max(f_{t-2,i}), \max(f_{t-1,i}), \max(f_{t,i})] \text{ and } v_i = \max(f_{t+1,i})$$

be the predictor vector and label value, respectively. Thus, we have a Gaussian process

$$v(u) \sim \mathcal{GP}(\mu(u), \mathcal{K}(u, u')),$$

where $\mu(.)$ and $\mathcal{K}(.,.)$ are the mean and covariance functions, respectively. Thus, the predicted value of v_* has a Gaussian distribution $v_* \sim \mathcal{GP}(\mu_*(u_*), \sigma_*^2(u_*))$ with the mean and variance are computed in (4.3) and (4.4).

5.3 Probabilistic Collocation Method

In this section, we show how we make an approximated outcome of the G&R using the PCM model.

5.3.1 Deterministic input

Consider the G&R model that takes the group of parameters $\gamma = \{k_d, \sigma_d, \mu_d\}$ as the input and produces the simulated MDC y as the output:

$$y = \eta(\gamma), \tag{5.2}$$

where $\eta(.)$ is the G&R computational code. Due to the high demand of computational resource and time of the code, we shall approximate $\eta(.)$ by utilizing a set of N basis functions $\{g_i(\gamma)\}$, with $i = 1, \cdots, N$, such that:

$$\hat{y} = \sum_{i=0}^{N} \beta_i g_i(\gamma), \tag{5.3}$$

where N and β_i are the order of the approximation and regression coefficients. For now, we assume that the set of functions $\{g_i(\gamma)\}$ is known. The regression coefficients $\{\beta_i\}$ can be solved as follows.

We define the residual between the truth and the approximation as follows.

$$R(\{\beta_i\}, \gamma) = \hat{y}(\gamma) - y(\gamma). \tag{5.4}$$

Applying the Ordinary Least Squares estimation to (5.3), we can straightforwardly obtain that the optimal set of coefficients $\hat{\beta}_i$ as follows:

$$\langle g_i(\gamma), R(\{\beta_i\}, \gamma)\rangle \ = \int_\gamma g_i(\gamma) R(\{\beta_i\}, \gamma) d\gamma = 0, \tag{5.5}$$

where $i = 1, \cdots, N$ and $\langle ., . \rangle$ represents the dot product between two deterministic functions. (5.5) can be solved by using a similar idea from the Gaussian quadrature [106] by approximating the integral as:

$$\int_\gamma R(\{\beta_i\}, \gamma) g_i(\gamma) d\gamma \simeq \sum_{j=1}^{N} v_j R(\{\beta_i\}, \tilde{\gamma}_j) g_i(\tilde{\gamma}_j) = 0, \tag{5.6}$$

where v_j and $\tilde{\gamma}_j$ are the weights and abscissas, respectively. If the weights and the basis functions are chosen such that $\prod_{i,j} v_j g_i(\tilde{\gamma}_j) > 0$ for all i and j, the summation in (5.6) can be further approximated as:

$$R(\{\beta_j\}, \tilde{\gamma}_j) = 0, \ j = 0, \cdots, N. \tag{5.7}$$

Note that the quadrature points $\tilde{\gamma}_j$ are also the *collocation points*. (5.7) can be used to find the coefficients $\{\beta_j\}$ by running the model at $N + 1$ different collocation points and solving a system of $N + 1$ equations.

5.3.2 Stochastic input

Suppose that the input γ now is a random vector with a known probability density function (PDF) $\pi(\gamma)$. Thus, (5.5) is transformed into the probability space as follows.

$$\int_\gamma \pi(\gamma) R(\{\beta_i\}, \gamma) g_i(\gamma) d\gamma = 0.$$

Similarly, with the proper choice of v_j and $g_i(\gamma)$, (5.7) becomes:

$$\pi(\gamma)R(\{\beta_j\}, \tilde{\gamma}_j) = 0,$$

where $j = 0, \cdots, N$. Since the PDF function $\pi(\gamma)$ is always positive, (5.7) can still be used to find the coefficients in the stochastic case.

5.3.3 Selection of basic functions and collocation points

Up to this point, we have assumed that the basic functions and the collocation points are given. In this section, we show how to determine them to satisfy the assumptions we have made in the previous section.

Theorem 1. *Consider a quadrature formula:*

$$\int_z W(z)F(z)dz \simeq \sum_{j=1}^{N} w_j F(z_j),$$

where w_j and z_j are the weights and abscissas. Then for a weight function $W(z) = z^\alpha(1-z)^\beta$, there exists an optimal choice of N quadrature points in the sense that the highest possible power of z in a power series expansion of $F(z)$ is correctly integrated when these z-values are used as quadrature points. The optimal quadrature points are the zeros of the polynomial of degree $(N+1)$, i.e., $P_{N+1}^{(\alpha,\beta)}(x)$, that satisfies the following orthogonality condition:

$$\int_z W(z)z^j P_{N+1}^{(\alpha,\beta)}(z)dz = 0, \text{ for } j = 0, \cdots, N. \tag{5.8}$$

The detail proof of Theorem 1 is provided in Chap 3 of [114]. In short, the choice of collocation points as the roots of the next order orthogonal polynomial will make the colloca-

tion method approximation closet to Galerkin's method, which yields the best performance among the Methods of Weighted Residual (MWR) [114].

Corollary 2. *Consider the same quadrature formula in Theorem 1, and a set of $N+1$ orthogonal polynomial functions $\{g_i(z)\}$. If the set of functions satisfies the condition:*

$$\int_z \pi(z) g_i(z) g_{N+1}(z) dz = 0, \ i = 1, \cdots, N, \tag{5.9}$$

then, it also satisfy condition (5.8) *and the zeros of $g_{N+1}(z)$ are the optimal quadrature points.*

The proof of Corollary 2 is straightforward and provided in Appendix C. If we choose the weight function to be the PDF of γ, i.e., $W(\gamma) = \pi(\gamma)$, (5.9) can be used to generate the set $\{g_i(\gamma)\}$ in a recursive manner as follows.

In practice, we define the initial conditions:

$$g_{-1} = 0,$$
$$g_0 = 1,$$

and the orthogonal polynomials can be obtained recursively by solving the equations:

$$\int_\gamma \pi(\gamma) g_i(\gamma) g_{i+1}(\gamma) d\gamma = 0, \ i = 1, \cdots, N.$$

However, for high order polynomials, solving (5.9) manually is time-consuming and error-prone. Thus, the set of basis functions can be computed alternatively and efficiently by using Favard theorem as follows.

Theorem 3. ***(Favard Theorem)*** *If a sequence of polynomials $\{P_i(z)\}$, where $i = 1, \cdots, N$,*

satisfies the recurrence relation:

$$P_i(z) = (z - \alpha_i)P_{i-1}(z) - \gamma_i P_{i-2}(z), \qquad P_{-1} = 0, P_1 = 1, \tag{5.10}$$

where α_i and β_i are real numbers. Then, $\{P_i(z)\}$ are orthogonal polynomials.

In this study, we adapt the work of Zhou et al. [129] that utilizes[1] $\alpha_i = \langle \gamma g_{i-1}, g_{i-1} \rangle$ and $\gamma_i = \sqrt{\langle g_{i-1}, g_{i-1} \rangle}$. The overall PCM algorithm is shown in Algorithm 3. Note that even here we use the same order for all random variables (N-th order), in general the orders can be different. Furthermore, even the PCM has been widely used for approximation of univariate prediction target, i.e., y is scalar, in our approach we extend it to be a multivariate approximation. The extension is straightforward since $\boldsymbol{\beta}$ in (5.11) now shall be a matrix instead of a vector. As a relative comparison, the G&R takes 2 days to run 512 sets of parameters, while PCM takes 20 seconds to produce the same results.

5.3.4 Maximum diameter curve transformation

It is a common practice in training deep structure that the original training data set is augmented to increase its variation and generalization. For instance, the multiple sources of training data sets can be mixed together to form a new one [71] or different view angles are achieved to general more poses of a 3D object [122].

We observe that some of the real MDCs do not start at 2 (cm) due to some truncation in the pre-processing step. Therefore, we generalize the training by augmenting the outcomes of PCM by applying a linear transition to the curve, an example of which is shown in dashed red line in Figure 5.3. Furthermore, since the G&R has only one damage function as shown in (5.1), it can not general data with a secondary local enlargement, which is observed in a

[1]The dot product between two functions $A(z)$ and $B(z)$ with respect to a random variable z with the probability density function $w(z)$ is defined as $\langle A(z), B(z) \rangle = \int_z w(z)A(z)B(z)dz$.

[2]For the sake of notational simplicity, we denote $\{k_d, \sigma_d, \mu_d\}$ as $\{\gamma^{[1]}, \gamma^{[2]}, \gamma^{[3]}\}$ in Algorithm 3.

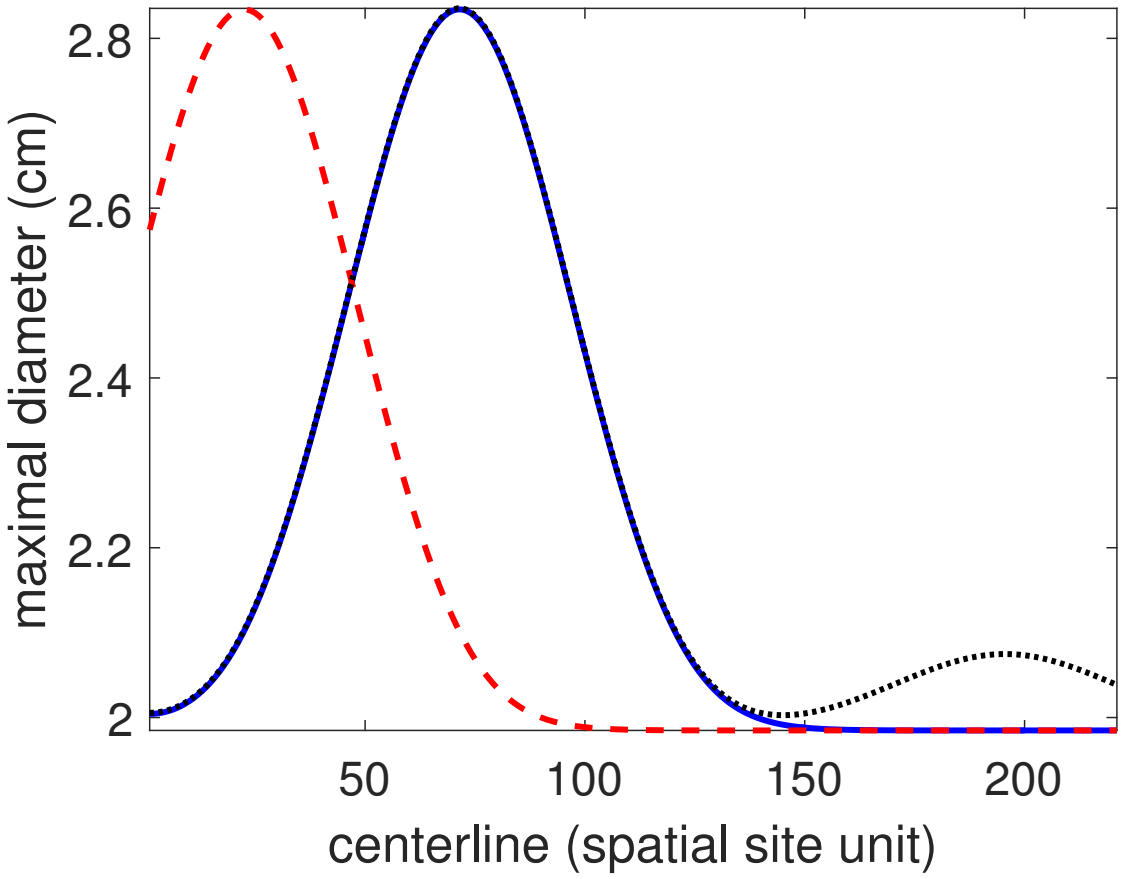


Figure 5.3 Two transformations of a maximum diameter curve: the original, translated, and second peak modified curves are shown in solid blue, dashed red, and dotted black lines, respectively. Note that the first halves of the original and second peak modified curves overlap each other.

number of our patients. Thus, we create a merged MDC by combining two different MDCs together. An example of a merged MDC is shown in dotted black line in Figure 5.3.

5.4 Deep Learning

In this section, we show how we use the artificially generated data from PCM to train the DBN and predict the MDC in a future time. We utilize a standard structure of the DBN [50] with two layers of Restricted Boltzmann Machine (RBM) [47] as shown in Figure 5.4. First of all, the two layers of RBM are pre-trained in an unsupervised manner. Then, we unfold

the two RBMs into an NN. Finally, we fine-tune the NN with the ground truth from both the real and artificially generated data by back-propagation.

5.4.1 Restricted Boltzmann Machine

An RBM is a building block of the DBN and we pre-train the RBM layers as follows.

Assume that we have two types of variables: the *observed* one ($\mathbf{x}$) and the *hidden* one ($\mathbf{h}$). The two variables are governed by an energy function $E(\mathbf{x}, \mathbf{h})$. Assume that both visible and hidden units have binomial distributions, a Boltzmann Machine is an energy-based model that has the energy function as a second-order polynomial [2]:

$$E(\mathbf{x}, \mathbf{h}|\theta) = -\mathbf{b}^T\mathbf{x} - \mathbf{c}^T\mathbf{h} - \mathbf{h}^T W\mathbf{x} - \mathbf{x}^T U\mathbf{x} - \mathbf{h}^T V\mathbf{h},$$

where θ is the collection of the offsets $\mathbf{b}$ and $\mathbf{c}$, and the weights W, U, and V. Thus, any probabilistic density function $P(\mathbf{x})$ (as well as joint and conditional PDFs) can be easily represented by a normalized form of the energy function. For instance, the PDF of $\mathbf{x}$ can be computed as:

$$P(\mathbf{x}) = \sum_{\mathbf{h}} \frac{e^{-E(\mathbf{x},\mathbf{h})}}{Z}, \tag{5.13}$$

where $Z = \sum_{\tilde{\mathbf{x}}} \sum_{\mathbf{h}} E(\tilde{\mathbf{x}}, \mathbf{h}|\theta)$ is the normalization factor and $\tilde{\mathbf{x}}$ is all possible values of the visible vector $\mathbf{x}$. The realization of $\tilde{\mathbf{x}}$ can be considered as reconstructed visible units.

In order to maximize the likelihood function to fit the model to a training data, we can use the log-likelihood gradient $\frac{\partial \log P(\mathbf{x})}{\partial \theta}$ to learn the parameter θ. However, when this energy-based PDF is used to compute the gradient of the log-likelihood, it requires sampling of two conditional probabilities: $P(\mathbf{h}|\mathbf{x})$ and $P(\mathbf{x}, \mathbf{h})$. This can be done via the Monte Carlo Markov Chain (MCMC) sampling [51], which is highly computationally expensive. Therefore, a Restricted Boltzmann Machine learnt by using Contrastive Divergence is normally utilized as a more efficient alternative solution.

The Restricted Boltzmann Machine is introduced by posting an additional condition:

$U = 0$ and $V = 0$. In other word, there are no connection between units in the same layer, either visible or hidden. Note that there is no links among units in the same layer in Figure 5.4. Furthermore, we assume that the visible unit has Gaussian distribution, i.e., $v_i \sim \mathcal{N}(a_i, \sigma_i)$, and the hidden unit has binomial distribution, i.e., $h_j \in \{0, 1\}$. Then, we can define a modified energy function as [47]:

$$E(\mathbf{x}, \mathbf{h}|\theta) = -\sum_{i=1}^{\mathcal{V}} \frac{(x_i - a_i)^2}{2\sigma_i^2} - \sum_{j=1}^{\mathcal{H}} c_j h_j - \sum_{i=1}^{\mathcal{V}} \sum_{j=1}^{\mathcal{H}} w_{ij} \frac{x_i}{\sigma_i} h_j \tag{5.14}$$

The conditional PDF of the visible units given the hidden ones can be computed as:

$$P(\mathbf{x}|\mathbf{h}, \theta) = \frac{e^{-E(\mathbf{x},\mathbf{h}|\theta)}}{\sum_{\mathbf{x}} e^{-E(\mathbf{x},\mathbf{h}|\theta)}}.$$

Note that $P(\mathbf{h}|\mathbf{x}, \theta)$ can be computed in the same manner. Using $E(\mathbf{x}, \mathbf{h}|\theta)$ in (5.14), we have:

$$\begin{aligned} P(h_j = 1|\mathbf{x}, \theta) &= \text{sigm}\left(\sum_{i=1}^{\mathcal{V}} w_{ij} x_i + c_j\right), \\ P(x_i = x|\mathbf{h}, \theta) &= \mathcal{N}\left(a_i + \sigma_i \sum_{j=1}^{\mathcal{H}} h_j w_{ij}, \sigma_i^2\right), \end{aligned} \tag{5.15}$$

where $\text{sigm}(x) = \frac{1}{1+\exp(-x)}$ is the sigmoid function.

The likelihood gradient can be computed by taking the derivative of $P(\mathbf{x})$ in (5.13) with

respect to θ, we have:

$$\begin{aligned}
&\frac{\log P(\mathbf{x})}{\partial\theta} \\
&= -\frac{1}{\sum_{\mathbf{h}} e^{-E(\mathbf{x},\mathbf{h})}} \sum_{\mathbf{h}} e^{-E(\mathbf{x},\mathbf{h})} \frac{\partial E(\mathbf{x},\mathbf{h})}{\partial\theta} \\
&+ \frac{1}{Z} \sum_{\tilde{\mathbf{x}}} \sum_{\mathbf{h}} e^{-E(\tilde{\mathbf{x}},\mathbf{h})} \frac{\partial E(\tilde{\mathbf{x}},\mathbf{h})}{\partial\theta} \\
&= -\sum_{\mathbf{h}} P(\mathbf{h}|\mathbf{x}) \frac{\partial E(\mathbf{x},\mathbf{h})}{\partial\theta} + \sum_{\tilde{\mathbf{x}}} \sum_{\mathbf{h}} P(\tilde{\mathbf{x}},\mathbf{h}) \frac{\partial E(\tilde{\mathbf{x}},\mathbf{h})}{\partial\theta} \\
&= -\mathbb{E}_{P(\mathbf{h}|\mathbf{x})} \left[\frac{\partial E(\mathbf{x},\mathbf{h})}{\partial\theta} \right] + \mathbb{E}_{P(\tilde{\mathbf{x}},\mathbf{h})} \left[\frac{\partial E(\tilde{\mathbf{x}},\mathbf{h})}{\partial\theta} \right].
\end{aligned} \tag{5.16}$$

The expectation $\mathbb{E}_{P(\mathbf{h}|\mathbf{x})}[.]$ is also referred as *positive phase* distribution or data distribution while the other expectation $\mathbb{E}_{P(\tilde{\mathbf{x}},\mathbf{h})}[.]$ is reffered as *negative phase* or model distribution [2].

Optimization of (5.16) involves sampling from $P(\tilde{\mathbf{x}},\mathbf{h})$ and it can be done by running Gibbs sampling until it reaches the equilibrium distribution for each parameter learning update iteration, which is extremely time consuming. Alternatively, Hinton [48] suggested the Contrastive Divergence (CD) learning that minimizes the difference between the data distribution and the one-step (or a finite number of steps) reconstructed distribution instead of minimizes the difference between the data and model distribution directly.

5.4.2 Contrastive Divergence

In a CD-k learning, the gradient of the parameter θ_i is approximated by:

$$\begin{aligned}
\Delta\theta_i &= -\epsilon \frac{\log P(\mathbf{x})}{\partial\theta_i} \\
&= \epsilon \left(\mathbb{E}_{P(\mathbf{h}|\mathbf{x})} \left[\frac{\partial E(\mathbf{x},\mathbf{h})}{\partial\theta_i} \right] - \mathbb{E}_{P_k(\tilde{\mathbf{x}},\mathbf{h})} \left[\frac{\partial E(\bar{\mathbf{x}},\mathbf{h})}{\partial\theta_i} \right] \right),
\end{aligned} \tag{5.17}$$

where $P_k(\tilde{\mathbf{x}},\mathbf{h})$ is the distribution of the reconstructed visible data $\bar{\mathbf{x}}$ after k steps of Gibbs sampling and ϵ is the learn rate. In this study, we utilize CD-1 that involves one full step of

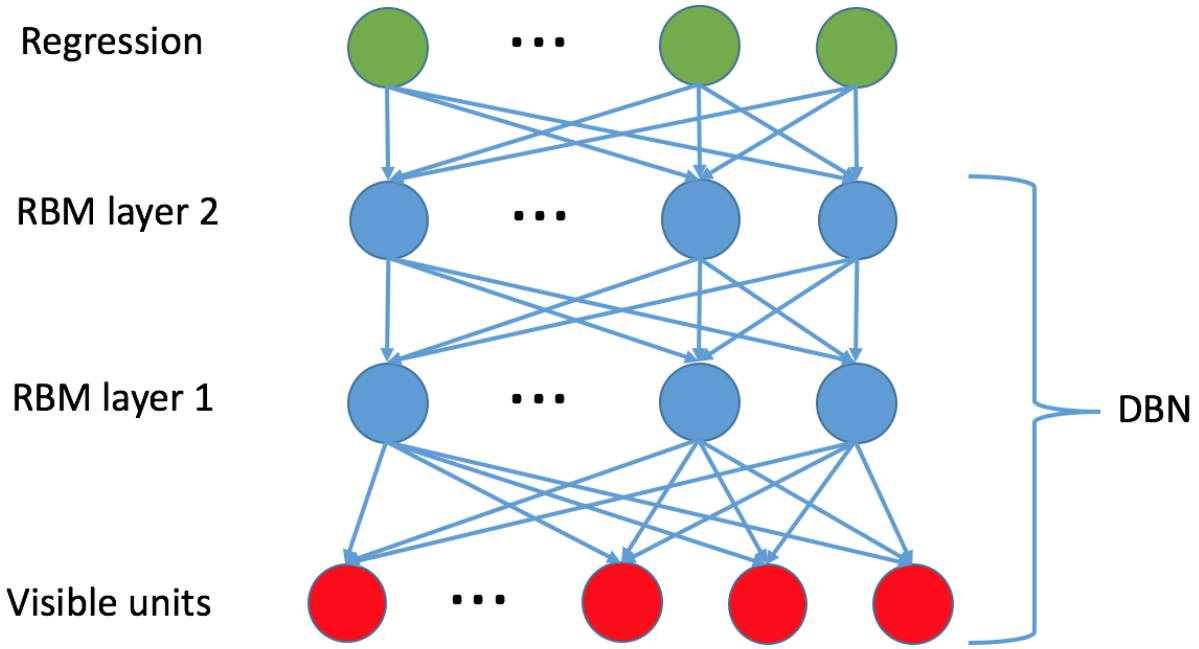


Figure 5.4 Deep architecture of the DBN. Two layers of RBM are trained in an unsupervised manner (pre-trained) using CD-1 algorithm. The top layer utilizes a neural network sigmoid regression for prediction.

Gibbs sampling, i.e., $P_1(\tilde{\mathbf{x}}, \mathbf{h})$. Applying (5.17) to (5.14), we have the learning update rules for each RBM layer as follows.

$$\begin{aligned} w_{ij} &\leftarrow w_{ij} + \epsilon \left(\mathbb{E}_{P(\mathbf{h}|\mathbf{x})} \left[\frac{h_j x_i}{\sigma_i} \right] - \mathbb{E}_{P_1(\tilde{\mathbf{x}},\mathbf{h})} \left[\frac{h_j x_i}{\sigma_i} \right] \right), \\ c_j &\leftarrow c_j + \epsilon \left(\mathbb{E}_{P(\mathbf{h}|\mathbf{x})} \left[h_j \right] - \mathbb{E}_{P_1(\tilde{\mathbf{x}},\mathbf{h})} \left[h_j \right] \right), \\ a_i &\leftarrow a_i + \epsilon \left(\mathbb{E}_{P(\mathbf{h}|\mathbf{x})} \left[-\frac{v_i - a_i}{\sigma_i^2} \right] - \mathbb{E}_{P_1(\tilde{\mathbf{x}},\mathbf{h})} \left[-\frac{v_i - a_i}{\sigma_i^2} \right] \right), \end{aligned} \tag{5.18}$$

where

$$\mathbb{E}_{P(\mathbf{h}|\mathbf{x})}\left[\frac{h_j x_i}{\sigma_i}\right] = \frac{1}{N}\sum_{t=1}^{N}\frac{\tilde{h}_j^{[t]} x_i^{[t]}}{\sigma_i},$$

$$\mathbb{E}_{P_1(\tilde{\mathbf{x}},\mathbf{h})}\left[\frac{h_j x_i}{\sigma_i}\right] = \frac{1}{N}\sum_{t=1}^{N}\frac{P(h_j^{[t]}=1|\tilde{x}_i^{[t]},\theta)\tilde{x}_i^{[t]}}{\sigma_i},$$

$$\mathbb{E}_{P(\mathbf{h}|\mathbf{x})}\left[h_j\right] = \frac{1}{N}\sum_{t=1}^{N}\tilde{h}_j^{[t]},$$

$$\mathbb{E}_{P_1(\tilde{\mathbf{x}},\mathbf{h})}\left[h_j\right] = \frac{1}{N}\sum_{t=1}^{N}P(h_j^{[t]}=1|\tilde{x}_i^{[t]},\theta),$$

$$\mathbb{E}_{P(\mathbf{h}|\mathbf{x})}\left[-\frac{v_i-a_i}{\sigma_i^2}\right] = \frac{1}{N}\sum_{t=1}^{N}\left(-\frac{x_i^{[t]}-a_i}{\sigma_i^2}\right),$$

$$\mathbb{E}_{P_1(\tilde{\mathbf{x}},\mathbf{h})}\left[-\frac{v_i-a_i}{\sigma_i^2}\right] = \frac{1}{N}\sum_{t=1}^{N}\left(-\frac{\tilde{x}_i^{[t]}-a_i}{\sigma_i^2}\right),$$

where N is number of samples and $\tilde{h}_j$ and $\tilde{x}_i$ are sampled with the distributions in (5.15).

5.5 Real case study

In this section, we demonstrate the effectiveness of our proposed predictive model with experiment using the real MDC obtained from [34].

5.5.1 Experimental set-up

For a deep structure, the selection of the number of hidden units, i.e., layers and nodes, is an important factor that determines the performance of the model. As a general rule, if the number of parameters is more than the number of training cases, the model will be overfitted. Therefore, as a rule of thumb for generative model of high-dimensional data, the number of parameters is constrained by several orders of magnitude greater than the dimensionality of the data [47]. In this study, our data has the dimensionality of 663. In order to determine the optimal number of layers, we study the effect of number of hidden

Number of hidden layers	Weights	Training time (s)	RMSE (cm)
1	67063	13.46	4.96
2	77263	17.22	5.08
3	87463	20.39	6.28
4	97663	23.65	9.76
5	107863	26.79	9.79

Table 5.2 Effect of number of layers on the prediction.

Number of nodes		Weights	Training time (s)	RMSE (cm)
RBM-1	RBM-2			
900	36	631599	68.78	5.04
400	100	306763	38.83	4.61
100	100	77263	17.22	5.08
100	400	107563	23.11	5.12
36	900	256803	24.48	7.02

Table 5.3 Effect of number of nodes in a 2-layer DBN on the prediction.

layers on the final prediction, which is shown in Table 5.2. In this analysis, we continuously add one more of layer of 100 hidden nodes to the structure, starting with 1. As shown in the table, the optimal number of weights is for one or two layers, when it is approximately two orders of magnitude of the dimensionality of the data.

In order to test the effect of number of nodes within each layer, we construct a number of 2-layer DBNs with different sets of hidden nodes in each layer. In particular, since there is a remarkable difference in the dimensions of the data and the label, i.e., 663 versus 221, we test the DBN with three distributions of nodes: highly concentrated near the input layer, equally distributed, and highly concentrated near the output layer. As shown in Table 5.3, as the nodes near the input layer decreases, the model fails to capture the representative features in the data, leading to higher prediction error.

One of the problem in our data is that we pre-train the DBN with a large artificially generated data (51200 sample) while the real data (8 samples) for fine-tuning is extremely limited compared to the artificial one. To tackle this problem, we fix the epoch[3] of the

[3]The number of times that the model is trained through the whole training set.

pre-training process to be 1 and increase the number of epochs of the fine-tuning process. By doing this, we improve the portion of generalization capability that is contributed by the real data set over the one from the artificial data as a counter-measurement for the large gap in the numbers of data points between two sources of data. The RMSE and training time for different epochs from 1 to 900 is shown in Figure 5.5. As the epochs is larger than 500, the error starts to increase again as the model starts to overfit the fine-tuning data and the generalization capacity is reduced.

Mixed-effect model: We compare the performance of our proposed method to the nonlinear mixed-effects, which has been used extensively as a powerful growth hierarchical model over the decades [11, 26, 107]. For the mixed-effects model, we utilize a basic form of the growth function as:

$$y_{i,j} = \alpha_0 + (\alpha_1 + b_1)t_{i,j} + (\alpha_2 + b_2)t_{i,j}^2 + \epsilon_{i,j},$$

where $y_{i,j}$ and $t_{i.j}$ are the diameter captured at the time j of patient i, $\mathbf{b} = [b_1, b_2]$ is the random-effects terms and $\mathbf{b} \sim \mathcal{N}(\mathbf{0}, \Sigma_b)$, $\boldsymbol{\alpha} = [\alpha_0, \cdots, \alpha_2]$ is the vector of parameters, and $\epsilon_{i,j}$ is the independent error term, i.e., $\epsilon_{i,j} \sim \mathcal{N}(0, \sigma_w^2)$. $\mathbf{b}$ and $\boldsymbol{\alpha}$ are fitted to the data via the `fminsearch` function in MATLAB.

Neural network with dropouts: dropout is a simple but effective technique to regularize a neural network to obtain a "trimmed" one that is less likely to be overfitting [105]. When dropout is applied, we randomly remove a set of units (from both visible and hidden layers) temporarily from the original network. In this study, we fix each unit with a probability of $p = 0.9$. Then, for training phase, a unit will be presented with the probability p. During the test phase, the weight of a unit will be multiplied by the value p.

5.5.2 Experimental results

The prediction error for 6 cases is shown in Table 5.4 and the plots of predictions are shown in Figure 5.6. The true, predicted by our proposed method, and predicted by the

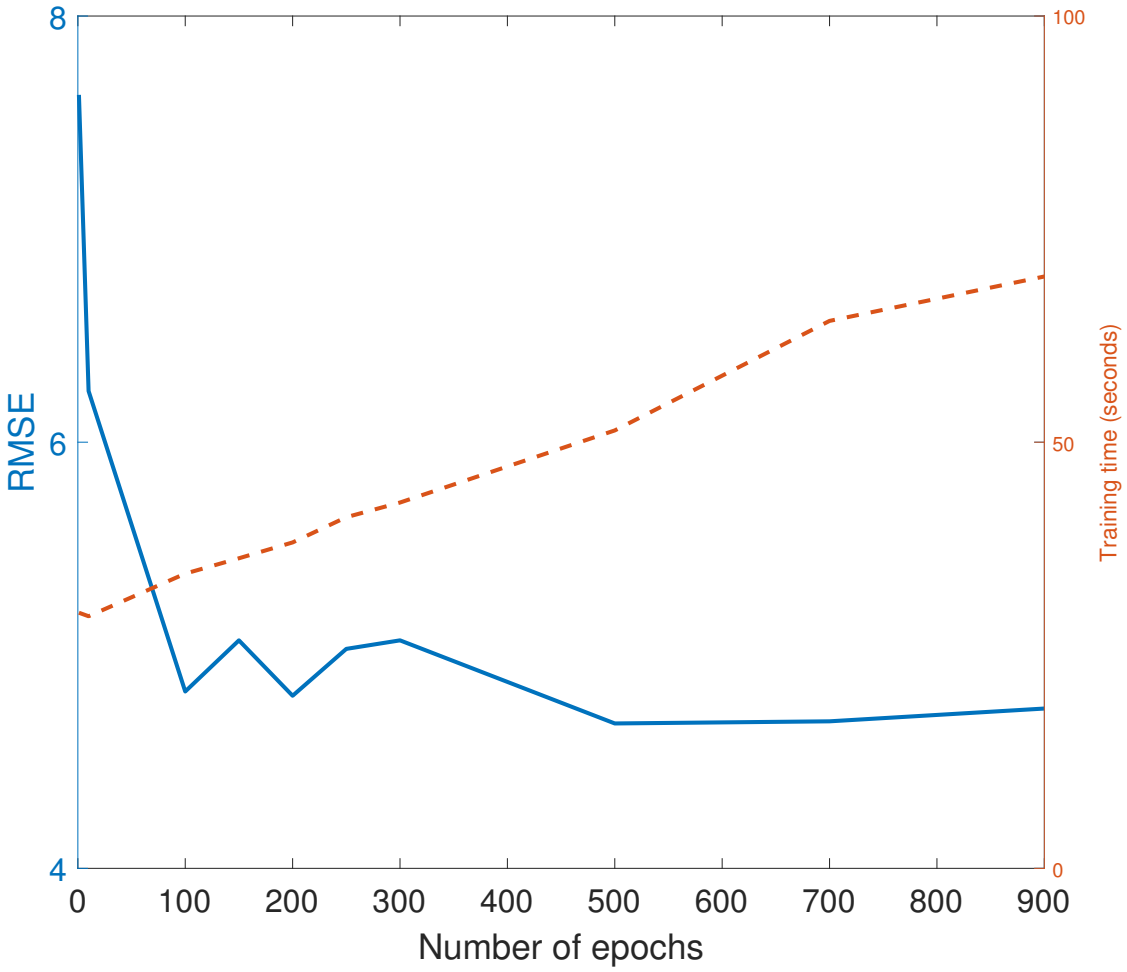


Figure 5.5 Effect of number of epochs to the prediction error. The RMSE and fine-tune training time are plotted in solid blue and dashed red lines, respectively. The training increases linearly with the number of epochs, while the RMSE rapidly decreases for the beginning then starts to increase again for the number of epochs that is larger than 500.

Patient ID	RMSE (cm)			Normalized error		
	DBN	DBN+DO	Mix-effects	DBN	DBN+DO	Mix-effects
P1	3.36	3.10	8.18	0.54	0.57	1.63
P2	4.19	6.3	7.68	0.37	0.46	0.98
P3	1.02	1.55	5.59	0.41	0.42	1.37
P4	2.95	2.81	4.16	0.55	0.55	1.14
P5	3.71	3.4	9.04	0.33	0.29	1.79
P6	5.85	6.46	5.73	1.2	1.31	1.37

Table 5.4 Comparison among different predictive models in terms of RMSE and normalized unit.

mixed-effects model MDCs are shown in solid black, dashed blue, and dotted dashed red lines, respectively. As shown in the table, our proposed method outperforms the mix-effects method with average 45% reduction in RMSE.

Notice that we have used an additional Gaussian process regression to predict the scale of the MDC. Thus, the final predictions are subjected to a bias error due to the Gaussian process regression. For instance, in the case of patient P5, the whole deep learning predicted curve has a consistent offset with respect to the true curve. Notice that patients P2 and P5 have a secondary local enlargements and our proposed method successfully produces accordingly predictions while the mixed-effect method fails to do so.

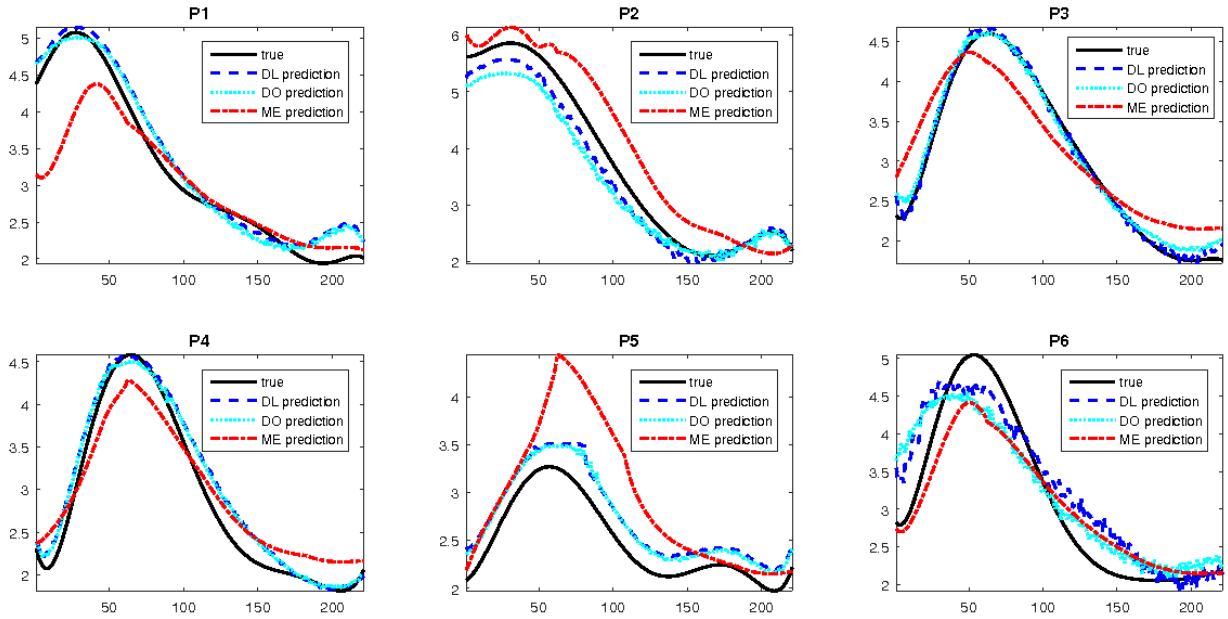


Figure 5.6 The true, predicted by our proposed method, and predicted by the mixed-effects model are shown in solid black ("true"), dashed blue ("DL prediction"), and dotted dashed red ("ME prediction") lines, respectively.

Algorithm 3 The Probabilistic Collocation Method (PCM) model

Part 1: Compute collocation points

1: Initialize $g_{-1}(\gamma) = G_{-1}(\gamma) = 0$ and $g_0(\gamma) = G_0(\gamma) = 1$.
2: **for** $i = 1, \cdots, N$ **do**
3: $\quad G_i(\gamma) = \gamma g_{i-1}(\gamma) - \langle \gamma g_{i-1}(\gamma), g_{i-1}(\gamma)\rangle g_{i-1}(\gamma) - \sqrt{\langle G_{i-1}(\gamma), G_{i-1}(\gamma)\rangle} g_{i-2}(\gamma)$
4: $\quad g_i(\gamma) = \frac{G_i(\gamma)}{\sqrt{\langle G_i(\gamma), G_i(\gamma)\rangle}}$
5: **end for**
6: Find the zeros of $g_{N+1}(\gamma) = 0$ as N collocation points.
7: Repeat steps 1-5 for other random variables. We denote three random variables[2] and three sets of basis functions as $\{\gamma_i^{[1]}, \gamma_i^{[2]}, \gamma_i^{[3]}\}$ and $\{g_j^{[1]}(\gamma), g_j^{[2]}(\gamma), g_j^{[3]}(\gamma)\}$, respectively, where $i = 1, \cdots, N$ and $j = 0, \cdots, N-1$
8: Create a permutation of three groups of collocation points of 3 random variables, i.e., $(\gamma_i^{[1]}, \gamma_j^{[2]}, \gamma_k^{[3]})$ where $i = 1, \cdots, N$, $j = 1, \cdots, N$, and $k = 1, \cdots, N$.

Part 2: Run the computational code at the collocation points

1: **for** $i = 1, \cdots, N$, $j = 1, \cdots, N$, $k = 1, \cdots, N$ **do**
2: $\quad$ Run $\eta(\gamma_i^{[1]}, \gamma_j^{[2]}, \gamma_k^{[3]})$.
3: **end for**

Part 3: Compute the coefficients $\boldsymbol{\beta}$

1: Concatenate the computational outcomes: $\mathbf{y} = \begin{bmatrix} \eta(\gamma_1^{[1]}, \gamma_1^{[2]}, \gamma_1^{[3]}) \\ \vdots \\ \eta(\gamma_N^{[1]}, \gamma_N^{[2]}, \gamma_N^{[3]}) \end{bmatrix}$.
2: Arrange the matrix $K = \begin{bmatrix} g_{N-1}^{[1]}(\gamma_1^{[1]}) g_{N-1}^{[2]}(\gamma_1^{[2]}) g_{N-1}^{[3]}(\gamma_1^{[3]}) & \cdots & g_0^{[1]}(\gamma_1^{[1]}) g_0^{[2]}(\gamma_1^{[2]}) g_0^{[3]}(\gamma_1^{[3]}) \\ \vdots & \ddots & \vdots \\ g_{N-1}^{[1]}(\gamma_N^{[1]}) g_{N-1}^{[2]}(\gamma_N^{[2]}) g_{N-1}^{[3]}(\gamma_N^{[3]}) & \cdots & g_0^{[1]}(\gamma_N^{[1]}) g_0^{[2]}(\gamma_N^{[2]}) g_0^{[3]}(\gamma_N^{[3]}) \end{bmatrix}$.
3: Compute the coefficients:

$$\boldsymbol{\beta} = K^{-1}\mathbf{y}. \tag{5.11}$$

Part 4: Approximate the computational code

1: For any new set of random variable $(\gamma_*^{[1]}, \gamma_*^{[2]}, \gamma_*^{[3]})$, the outcome can be approximated by:

$$\eta^*(\gamma_*^{[1]}, \gamma_*^{[2]}, \gamma_*^{[3]}) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} \boldsymbol{\beta}_{i,j,k} g_i^{[1]}(\gamma_*^{[1]}) g_j^{[2]}(\gamma_*^{[2]}) g_k^{[3]}(\gamma_*^{[3]}). \tag{5.12}$$

Chapter 6

Conclusion and future works

In this chapter, we briefly summarize the main factors of each chapters as well the limitations of the discussed methods. Additionally, we show how the works can be improved in the future with further investigation.

In chapter 2, we present a novel approach to use vision data for the robot localization. The predictive statistics of vision data is learned in advance and used in order to estimate the position of a vehicle, equipped just with an omnidirectional camera in both indoor and outdoor environments. The multivariate GP model is used to model a collection of selected visual features. The locations are estimated by maximizing the likelihood function without fusing combining vehicle dynamics with measured features in order to evaluate the proposed scheme alone. Hence, we believe that the localization performance will be further improved when vehicle dynamics are fused together via Kalman filtering or particle filtering.

One of the limitation of the method is that after the initial training phase, learning is discontinued. If the environment changes, it is desirable that the localization routines adapt to the changes in the environment. Thus, a future research direction is to develop a localization scheme that is adaptive to changes in the environment.

In chapter 3, we introduce a novel appearance-based approach based on group LASSO regression. We have shown the effectiveness of our method in feature selection and localiza-

tion enhancement by combing the group LASSO regression and the EKF. The experiment study shows the significant reduction (75.5%) in the features while improving the localization performance.

Notice that since the direct dynamic model that maps the visual features into the robot's positions, i.e., $\mathbf{q}_i = h(\mathbf{x})$, are not available, we use the LASSO (as well as the group LASSO) as a linear regression approximation. Thus, the observations for the EKF and PF are noisy estimated localization results from the LASSO. The linearity assumption could restrict the localization performance. Therefore, a potential future study is to apply a non-linear regression technique, such as Gaussian process, to relax the linearity condition.

In chapter 4, we have formulated the modeling and growth of the AAA using patient-specific point clouds data in a statistical framework. After utilizing the spatio-temporal Gaussian process observation model to construct the implicit surface field, we develop a dynamic model to infer the evolution of the field at a future time. Finally, we extract the surface from the predicted field for visualization of an aneurysm.

The results of the case studies have shown the efficacy of our proposed scheme by comparing the predicted AAAs with the ground truth. To the best of our knowledge, this is the first study that predicts the growth of the 3D AAA shape by using patient-specific data in a statistical framework and provides uncertainty of the predicted AAA shape. In doing so, the study yields insightful findings as well as highlights the limitations of using such models for studying the nature of the growth of an AAA. Possible clinical applications and limitations of our approach are also discussed along with prospective research directions. With advances in computing technologies and new sampling methods, the use of the Bayesian approach will have a great potential to revolutionize application of computational modeling in the treatment of vascular diseases.

In chapter 5, we propose a novel method that utilizes the Deep Belief Network to predict the growth of an AAA via the maximum diameter curves. The deep structure is pre-trained on a set of simulated data and fine-tuned by the longitudinal data of the patients. The

method is implemented on a real case study with 6 patients. The outcomes validate the effectiveness of our proposed method.

A limitation of the current approach arises from the fact that the use of a additional regression to predict the normalization factor separately increases the bias error. Thus, a future research direction is to accommodate uncertainty in Gaussian process to the final results. For instance, drop-out can be used as an approximated Bayesian technique to represent represent model uncertainty in deep learning [33].

Appendices

Appendix A E-M algorithm

In this section, we show how to use the E-M (Expectation-Maximization) algorithm to fit the linear dynamic model to the training data set.

First of all, we define a log likelihood function conditioned on the available data up to the time T as (for notational simplicity, we omit the $\mathbf{x}$ argument and put the time t in the subscript for the function $f(.)$, i.e., A and f_t imply $A(\mathbf{x})$ and $f(\mathbf{x}, t)$):

$$\begin{aligned}
\log(\mathcal{L}) := & -\frac{1}{2}\log|\Sigma_0| - \frac{1}{2}\left((f_0-\mu_0)^T\Sigma_0^{-1}(f_0-\mu_0)\right) \\
& -\frac{1}{2}\sum_{t=1}^{T}\log|\Delta_t^2\Sigma_w| - \frac{1}{2}\sum_{t=1}^{T}\left((f_t-f_{t-1}-\Delta_t A)^T(\Delta_t^2\Sigma_w)^{-1}(f_t-f_{t-1}-\Delta_t A)\right) \\
& -\frac{1}{2}\sum_{t=1}^{T}\log|\Sigma_v(t)| - \frac{1}{2}\sum_{t=1}^{T}\left((y_t-f_t)^T\Sigma_v(t)^{-1}(y_t-f_t)\right).
\end{aligned} \tag{1}$$

The E-step

Let $\Psi(A, \Sigma_w) = \mathbb{E}_r[\log(\mathcal{L})|\mathcal{D}_{1:T}, A, \Sigma_w]$ be the expected value of the log likelihood function at iteration r, apply the expectation to both sides of (1) and utilize the matrix identity

$$\mathbb{E}[\mathbf{x}^T A\mathbf{x}] = \mathrm{Tr}(A\mathbb{E}[\mathbf{x}\mathbf{x}^T]) \tag{2}$$

we can derive $\Psi(A, \Sigma_w)$ as follows. Consider each line of (1):

We can see straightforwardly that the *first* line of (1) becomes the first line of (4) by simply applying (2).

Apply the identity (2) to the *second* line of (1), we have:

$$\begin{aligned}
& \mathbb{E}\left[-\frac{1}{2}\sum_{t=1}^{T}\log|\Delta_t^2\Sigma_w| - \frac{1}{2}\sum_{t=1}^{T}\left((f_t-f_{t-1}-\Delta_t A)^T(\Delta_t^2\Sigma_w)^{-1}(f_t-f_{t-1}-\Delta_t A)\right)\right] \\
& = -\frac{1}{2}\sum_{t=1}^{T}\log|\Delta_t^2\Sigma_w| - \frac{1}{2}\sum_{t=1}^{T}\mathbb{E}\left[(f_t-f_{t-1}-\Delta_t A)^T(\Delta_t^2\Sigma_w)^{-1}(f_t-f_{t-1}-\Delta_t A)\right]
\end{aligned}$$

Apply (2) to the later sum and expand it, we have:

$$\begin{aligned}
&\sum_{t=1}^{T} \mathrm{Tr}\left\{(\Delta_t^2\Sigma_w)^{-1}\mathbb{E}[(f_t - f_{t-1} - \Delta_t A)(f_t - f_{t-1} - \Delta_t A)^T]\right\} \\
&= \sum_{t=1}^{T} \mathrm{Tr}\left\{(\Delta_t^2\Sigma_w)^{-1}\left(C_t - B_t - B_t^T + E_t + \Delta_t H_t + \Delta_t^2 AA^T\right)\right\},
\end{aligned}$$

where

$$\begin{aligned}
H_t &= (\mathbb{E}[f_{t-1}] - \mathbb{E}[f_t])A^T + A(\mathbb{E}[f_{t-1}] - \mathbb{E}[f_t])^T, \\
C_t &= \mathbb{C}\mathrm{ov}(f_t|\mathcal{D}_{1:T}) + \mathbb{E}[f(t)|\mathcal{D}_{1:T}]\mathbb{E}[f(t)|\mathcal{D}_{1:T}]^T, \\
B_t &= \mathbb{C}\mathrm{ov}(f_t, f_{t-1}|\mathcal{D}_{1:T}) + \mathbb{E}[f_t|\mathcal{D}_{1:T}]\mathbb{E}[f_{t-1}|\mathcal{D}_{1:T}]^T, \\
E_t &= \mathbb{C}\mathrm{ov}(f_{t-1}|\mathcal{D}_{1:T}) + \mathbb{E}[f_{t-1}|\mathcal{D}_{1:T}]\mathbb{E}[f_{t-1}|\mathcal{D}_{1:T}]^T.
\end{aligned} \tag{3}$$

Apply the identity (2) to the *third* line of (1) we have:

$$\begin{aligned}
&-\frac{T}{2}\log|\Sigma_v| - \frac{1}{2}\mathrm{Tr}\left\{\Sigma_v^{-1}\sum_{t=1}^{T}\mathbb{E}[(y_t - f_t)(y_t - f_t)^T]\right\} \\
&= -\frac{T}{2}\log|\Sigma_v| - \frac{1}{2}\mathrm{Tr}\left\{\Sigma_v^{-1}\sum_{t=1}^{T}\left(y_t y_t^T - y_t f_t^T - y_t^T f_t + \mathbb{E}[f_t f_t^T]\right)\right\} \\
&= -\frac{T}{2}\log|\Sigma_v| - \frac{1}{2}\mathrm{Tr}\left\{\Sigma_v^{-1}\sum_{t=1}^{T}\left((y_t - \mathbb{E}[f_t])(y_t - \mathbb{E}[f_t])^T + \mathbb{C}\mathrm{ov}(f_t, f_t)\right)\right\}.
\end{aligned}$$

Finally, combing the three lines together, we have:

$$\begin{aligned}
&\Psi(A, \Sigma_w) = -\frac{1}{2}\log|\Sigma_0| - \frac{1}{2}\mathrm{Tr}\left\{\Sigma_0^{-1}\left(\mathbb{C}\mathrm{ov}(f(0)|\mathcal{D}_{1:T}) + \mathbb{E}(f(0)|\mathcal{D}_{1:T})\mathbb{E}(f(0)|\mathcal{D}_{1:T})^T\right)\right\} \\
&- \frac{1}{2}\sum_{t=1}^{T}\log|\Delta_t^2\Sigma_w| - \frac{1}{2}\sum_{t=1}^{T}\mathrm{Tr}\left\{(\Delta_t^2\Sigma_w)^{-1}\left(C_t - B_t - B_t^T + E_t + \Delta_t H_t + (\Delta_t)^2 AA^T\right)\right\} \\
&- \frac{1}{2}\sum_{t=1}^{T}\log|\Sigma_v(t)| \\
&- \frac{1}{2}\mathrm{Tr}\left\{(\Delta_t\Sigma_v(t))^{-1}\sum_{t=1}^{T}\left((y(t) - \mathbb{E}[f(t)|\mathcal{D}_{1:T}])(y(t) - \mathbb{E}[f(t)|\mathcal{D}_{1:T}])^T + \mathbb{C}\mathrm{ov}(f(t)|\mathcal{D}_{1:T})\right)\right\},
\end{aligned} \tag{4}$$

where H_t, C_t, B_t, and E_t are defined in (3). This is the end of the E-step.

The M-step

In the M-step of iteration r, we find the values of $A(r+1)$ and $\Sigma_w(r+1)$ that maximize (4). Let $M = \Delta_t^2 \Sigma_w$ and take the derivative of (4) with respect to A:

$$
\begin{aligned}
&\frac{\partial \Psi}{\partial A} \\
&= \frac{\partial \mathrm{Tr}}{\partial A} \left\{ \sum_{t=1}^{T} M \left(\Delta_t (\mathbb{E}[f_{t-1}] - \mathbb{E}[f_t]) A^T + \Delta_t A (\mathbb{E}[f_{t-1}]^T - \mathbb{E}[f_t]^T) + \Delta_t^2 A A^T \right) \right\} \\
&= \sum_{t=1}^{T} \left\{ \frac{\partial_{\mathrm{Tr}}}{\partial A} (\Delta_t M \left(\mathbb{E}[f_{t-1}] - \mathbb{E}[f_t] \right) A^T) + \frac{\partial_{\mathrm{Tr}}}{\partial A} (\Delta_t M A (\mathbb{E}[f_{t-1}]^T - \mathbb{E}[f_t]^T)) + \frac{\partial_{\mathrm{Tr}}}{\partial A} (\Delta_t^2 M A A^T) \right\} \\
&= \sum_{t=1}^{T} \left(\Delta_t (M + M^T) (\mathbb{E}[f_{t-1}] - \mathbb{E}[f_t]) \right) + \left(\sum_{t=1}^{T} \Delta_t^2 (M + M^T) \right) A \\
&= (\Sigma_w + \Sigma_w^T) \sum_{t=1}^{T} \Delta_t^3 (\mathbb{E}[f_{t-1}] - \mathbb{E}[f_t]) + (\Sigma_w + \Sigma_w^T) \left(\sum_{t=1}^{T} \Delta_t^4 \right) A = 0.
\end{aligned}
$$

Thus,

$$
A(r+1) = A = \left(\sum_{t=1}^{T} \Delta_t^4 \right)^{-1} \left(\sum_{t=1}^{T} \Delta_t^3 \left(\mathbb{E}[f_t] - \mathbb{E}[f_{t-1}] \right) \right)
$$

Then, take the derivative of (4) with respect to Σ_w yields:

$$
\begin{aligned}
&\frac{\partial \Psi}{\partial \Sigma_w} \\
&= \frac{\partial}{\partial \Sigma_w} \left\{ \sum_{t=1}^{T} \log |\Delta_t^2 \Sigma_w| + \sum_{t=1}^{T} \mathrm{Tr} \left((\Delta_t^2 \Sigma_w)^{-1} (C_t - B_t - B_t^T + E_t + \Delta_t H_t + \Delta_t^2 A A^T) \right) \right\} \\
&= \sum_{t=1}^{T} (\Delta_t^2 \Sigma_w)^{-T} \Delta_t^2 - \sum_{t=1}^{T} \left((\Delta_t^2 \Sigma_w)^{-T} (C_t - B_t - B_t^T + E_t + \Delta_t H_t + \Delta_t^2 A A^T)^T (\Delta_t^2 \Sigma_w)^{-T} \Delta_t^2 \right) \\
&= \left(\sum_{t=1}^{T} 1 \right) \Sigma_w^{-T} - \Sigma_w^{-T} \left(\sum_{t=1}^{N} \Delta_t^{-2} (C_t - B_t - B_t^T + E_t + \Delta_t H_t + \Delta_t^2 A A^T)^T \right) \Sigma_w^{-T} = 0
\end{aligned}
$$

Thus,

$$\Sigma_w(r+1) = \Sigma_w = \frac{1}{T}\left(\sum_{t=1}^{N} \Delta_t^{-2}(C_t - B_t - B_t^T + E_t + \Delta_t H_t + \Delta_t^2 A A^T)\right),$$

with A computed above.

Finally, we have the update for A and Σ_w as follows.

$$A(r+1) = \left(\sum_{t=1}^{T} \Delta_t^4\right)^{-1} \left(\sum_{t=1}^{T} \Delta_t^3 \left(\mathbb{E}[f(t)|\mathcal{D}_{1:T}] - \mathbb{E}[f(t-1)|\mathcal{D}_{1:T}]\right)\right), \tag{5a}$$

$$\Sigma_w(r+1) = \frac{1}{T}\left(\sum_{t=1}^{N} \Delta_t^{-2}(C_t - B_t - B_t^T + E_t + \Delta_t H_t + \Delta_t^2 A A^T)\right). \tag{5b}$$

Note that we have not discussed how to compute $\mathbb{E}[f_t|\mathcal{D}_{1:T}]$, $\mathbb{C}\text{ov}(f_t|\mathcal{D}_{1:T})$, and $\mathbb{C}\text{ov}(f_t, f_{t-1}|\mathcal{D}_{1:T})$, which are needed to compute $A(r+1)$, B_t, and C_t. Those statistical quantities are conditioned the whole data set $\mathcal{D}_{1:T}$ so they need to be computed prior to the sum in (5). Using the Kalman filter smoothing framework [101], first we propagate the means ($\mathbb{E}[f_t]$) and covariances ($\mathbb{C}\text{ov}(f_t)$) through the data. For each step of propagation through the data, the quantities are conditioned on the available data up to that particular step. Then, once we reach the end of data set, we compute backward to find the mean, covariance, and correlation conditioned on the whole data set. For the initial condition, we assume that $f_0|\mathcal{D}_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$, i.e., $\mathbb{E}[f_0|\mathcal{D}_0] = \mu_0$ and $\mathbb{C}\text{ov}(f_0|\mathcal{D}_0) = \Sigma_0$ where μ_0 and Σ_0 are known. The overall Kalman Filter algorithm is shown in Algorithm 4. Computed A and Σ_w are plugged back into the log likelihood expectation (4) then the E-step and M-step are repeated. The whole process continues until the log likelihood expectation converges.

Once $A(\mathbf{x})$ and $\Sigma_w(\mathbf{x})$ are calibrated to the training data set, the predictive distribution of the IS field for the test data set, which is shown in (10) of the main manuscript, is straightforwardly obtained from the forward computation of the Kalman Filter algorithm (the first two lines of the `for` loop of the forward computation section of Algorithm. 4).

Algorithm 4 Kalman Filter updating

Forward computation:

for $t = 1, \cdots, T$ **do**

$\mathbb{E}[f(t)|\mathcal{D}_{1:t-1}] = \mathbb{E}[f(t-1)|\mathcal{D}_{1:t-1}] + \Delta_t A$

$\mathbb{C}\text{ov}(f(t)|\mathcal{D}_{1:t-1}) = \mathbb{C}\text{ov}(f(t-1)|\mathcal{D}_{1:t-1}) + \Sigma_w \Delta_t^2$

$K_t = \mathbb{C}\text{ov}(f(t)|\mathcal{D}_{1:t-1}) \left(\mathbb{C}\text{ov}(f(t)|\mathcal{D}_{1:t-1}) + \Sigma_v(t)\right)^{-1}$

$\mathbb{E}[f(t)|\mathcal{D}_{1:t}] = \mathbb{E}[f(t)|\mathcal{D}_{1:t-1}] + K_t(y(t) - \mathbb{E}[f(t)|\mathcal{D}_{1:t-1}])$

$\mathbb{C}\text{ov}(f(t)|\mathcal{D}_{1:t}) = (I - K_t)\mathbb{C}\text{ov}(f(t)|\mathcal{D}_{1:t-1})$

end for

Backward computation:

In order to compute $\mathbb{E}[f(t)|\mathcal{D}_{1:T}]$, $\mathbb{C}\text{ov}(f(t)|\mathcal{D}_{1:T})$, and $\mathbb{C}\text{ov}(f(t), f(t-1)|\mathcal{D}_{1:T})$, one can use the backward computations:

for $t = T, \cdots, 1$ **do**

$J_{t-1} = \mathbb{C}\text{ov}(f(t-1)|\mathcal{D}_{1:t-1})\mathbb{C}\text{ov}(f(t)|\mathcal{D}_{1:t-1})^{-1}$

$\mathbb{E}[f(t-1)|\mathcal{D}_{1:T}] = \mathbb{E}[f(t-1)|\mathcal{D}_{1:t-1}] + J_{t-1}\left(\mathbb{E}[f(t)|\mathcal{D}_{1:T}] - \mathbb{E}[f(t-1)|\mathcal{D}_{t-1}]\right)$

$\mathbb{C}\text{ov}(f(t-1)|\mathcal{D}_{1:T}) = \mathbb{C}\text{ov}(f(t-1)|\mathcal{D}_{1:t-1}) + J_{t-1}(\mathbb{C}\text{ov}(f(t)|\mathcal{D}_{1:T}) - \mathbb{C}\text{ov}(f(t)|\mathcal{D}_{1:t-1}))J_{t-1}^T$

if $t \neq 1$ **then**

$\mathbb{C}\text{ov}(f(t-1), f(t-2)|\mathcal{D}_{1:T})$

$= \mathbb{C}\text{ov}(f(t-1)|\mathcal{D}_{1:t-1})J_{t-2}^T$

$+ J_{t-1}\left(\mathbb{C}\text{ov}(f(t), f(t-1)|\mathcal{D}_{1:T}) - \mathbb{C}\text{ov}(f(t-1)|\mathcal{D}_{1:t-1})\right) J_{t-2}^T$

end if

end for

For $t = T$: $\mathbb{C}\text{ov}(f(T), f(T-1)|\mathcal{D}_{1:T}) = (I - K_T)\mathbb{C}\text{ov}(f(T-1)|\mathcal{D}_{1:T-1})$.

Appendix B Surface extraction

Threshold Determination

Let $\mathbf{x}_\circ$ be the on-surface points at the time t such that

$$\mathbf{x}_\circ = \{\mathbf{x} \subset \mathcal{S} : f(\mathbf{x}, t) \leq \lambda\},$$

where λ is the threshold value. To determine the threshold for the training data, we run an exhaustive search through the range of possible values of the IS field. For each candidate threshold, we reconstruct the training point cloud. Finally, we choose the training threshold to be the one that yields the most similar reconstructed point clouds.

To evaluate the similarity between two point clouds, we use the Hausdorff distance $H(.,.)$ [53] that is defined as follows.

$$H(P, O) = \max\left(h(P, O), h(O, P)\right), \tag{6}$$

where

$$h(P, O) = \max_{\hat{\mathbf{x}} \in P} \min_{\bar{\mathbf{x}} \in O} \|\hat{\mathbf{x}} - \bar{\mathbf{x}}\|.$$

For the exhaustive search, we gradually raise the threshold λ from 0 to 1, the final threshold is chosen such that the corresponding point cloud yields the lowest Hausdorff distance. Fig. A.1 shows one example of patient P5. The values of IS field are plotted with respect to all points on the grid. The threshold for training and test fields are indicated by the black and dashed solid lines, respectively.

Binary encoding and matching

Note that each particular value of the threshold classifies the whole 3D lattice into a unique spatial pattern of two categories: on-surface and off-surface. Therefore, we utilize that fact to assign the test threshold (the threshold value for the predicted IS field) to the value that yields the most similar spatial pattern to the training data. In particular, we

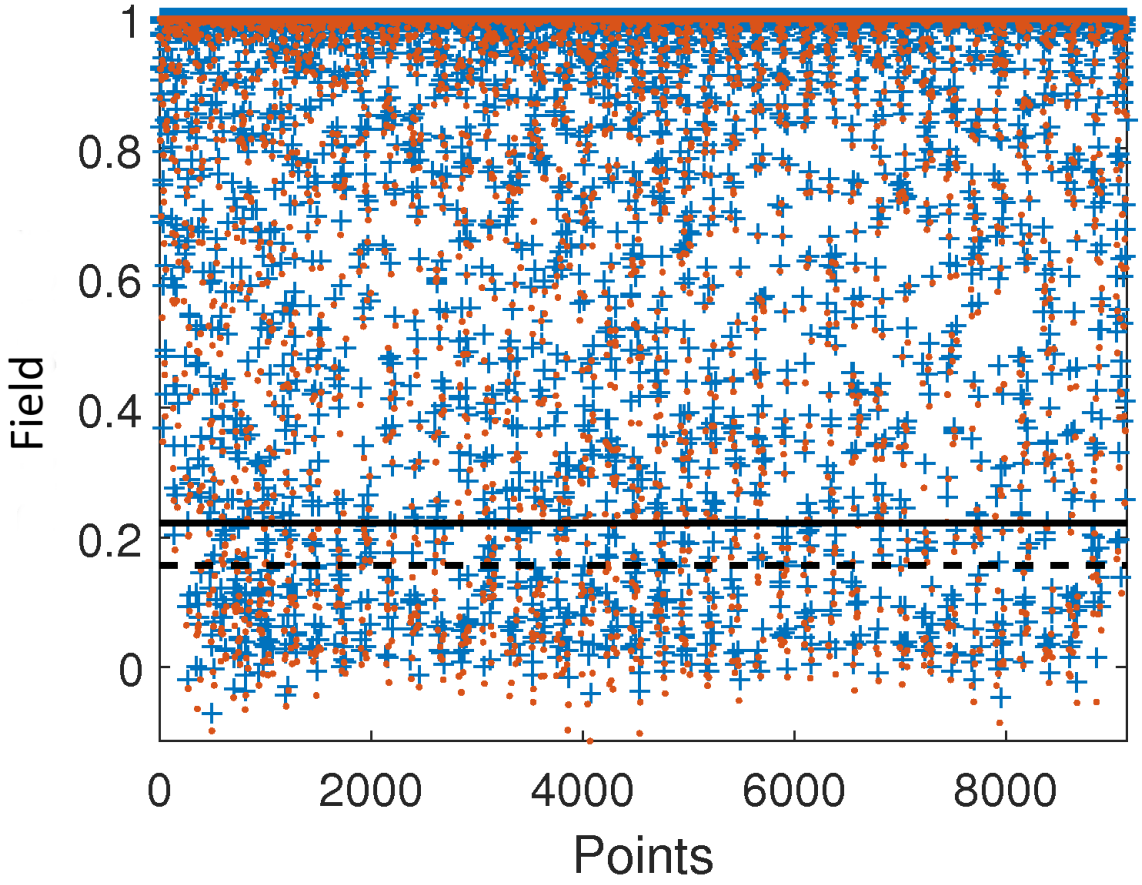


Figure A.1 Threshold determination for patient P5: the field of points on the lattice for the training and test fields are plotted in blue pluses and red dots, respectively. The thresholds for training and test data are shown in solid and dashed lines, correspondingly.

encode two spatial site $\mathcal{S}$'s that are classified by training and test thresholds into two binary vectors. Then, we select the test threshold that yields the closest binary vector to the training one. The detail of the process is discussed as follows.

First, we map on-surface points onto a binary vector as follows.

$$g(\mathbf{x}, \lambda) = \begin{cases} 1 \text{ if } f(\mathbf{x}, L) \leq \lambda, \\ 0 \text{ if } f(\mathbf{x}, L) > \lambda. \end{cases}$$

Let $\mathbf{v}_t(\mathbf{x}_{1:n}, \lambda) = [g(\mathbf{x}_1, \lambda), \cdots, g(\mathbf{x}_n, \lambda)]^T$ be the binary vector obtained at the scan time t with threshold λ, where $\mathbf{x}_{1:n}$ is the collection of coordinates of the spatial site $\mathcal{S}$.

Then, we use the Jaccard indexes [92] to estimate the similarity between two binary vectors. The Jaccard index can be computed as follows.

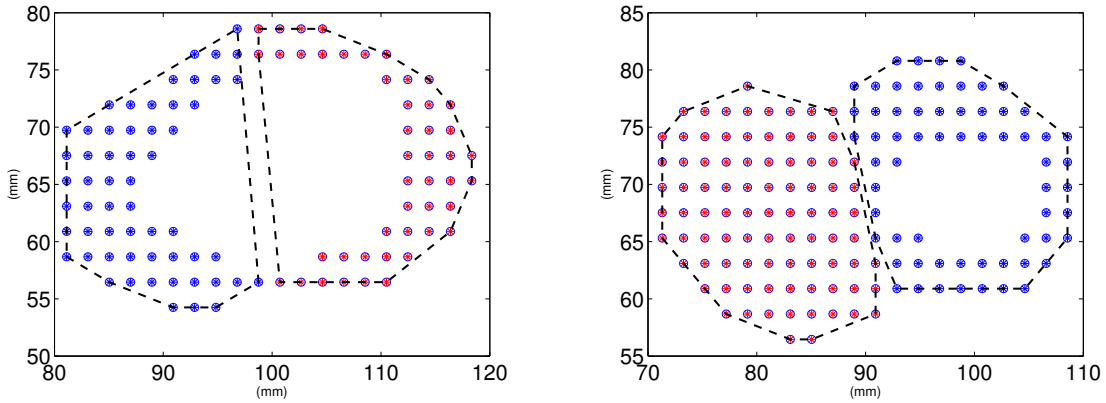
$$J(A, B) = \frac{|A \cap B|}{|A \cup B|},$$

where A, B are two sample sets. The threshold that yields lowest Jaccard index is chosen to be the test threshold, i.e.,

$$\hat{\lambda}_L = \arg\min_{\lambda_L} \{J(\mathbf{v}_{L-1}(\mathbf{x}_{1:n}, \lambda_{L-1}), v_L(\mathbf{x}_{1:n}, \lambda_L))\}.$$

Point cloud post-processing

In this section, we discuss the final step for surface refining. Figs. A.2 show the cross section of the predicted surface at the heights $z = 86.9$ (mm), which is at the middle of the AAA, and $z = 76.21$ (mm) where the AAA starts to branch out into two segments. The selected points by the test threshold are plotted in blue circles. Note that inner points that locate near the surface can be falsely classified to be on-surface. To eliminate those inner points, we apply two connected component operators in the x-y plane of each cross section: k-means clustering [42] and convex hull [37].



(a) (b)

Figure A.2 Cross view of predicted surface of the AAA at two different vertical heights: (a) $z = 86.9$ (mm) and (b) $z = 76.21$ (mm). The point clouds and two clusters are plotted in circles and red/blue stars, respectively. The surface points that are selected by the convex hull are connected with the black dashed line. As shown in (b), without the clustering, two branches will be falsely merged into one.

First, we cluster the points into *two* clusters, that are shown in red and blue stars in Figs. A.2. The number of clusters is set to 2 due to the specific structure of the AAA, which has a tubular structure with two branches at the end. Note that clustering the points into two group prevents the convex hull operator from falsely merging two branches into one large tube.

Then, we apply the convex hull operator to filter out only the points that lie on the boundary, which are shown connected with dashed lines in Figs. A.2. Those points that do not lie on the dashed black line are removed. Notice that without clustering in Fig. A.2-(b), the convex hull operator will falsely merge two sections into one. After this post-processing step, we finally obtain the predicted point cloud of the AAA's surface.

Appendix C Proof of Corollary 2

We illustrate the proof for $N = 2$, then the proof for an arbitrary N is straightforward. Let

$$g_0 = 1,$$

$$g_1 = ax + b,$$

$$g_2 = cx^2 + dx + e.$$

So, applying (5.9), we have:

$$\begin{aligned}
\int_x \pi(x) g_0(x) g_3(x) dx &= \int_x \pi(x) g_3(x) dx = 0, \\
\int_x \pi(x) g_1(x) g_3(x) dx &= \int_x \pi(x)(ax + b) g_3(x) dx = 0, \\
\int_x \pi(x) g_2(x) g_3(x) dx &= \int_x \pi(x)(cx^2 + dx + e) g_3(x) dx = 0.
\end{aligned}$$

Rearrange the terms, we have an equivalent system of equations:

$$\begin{aligned}
(1 + b + e) \int_x \pi(x) g_3(x) dx = 0 &\rightarrow \int_x \pi(x) g_3(x) dx = 0, \\
(a + d) \int_x \pi(x) x g_3(x) dx = 0 &\rightarrow \int_x \pi(x) x g_3(x) dx = 0, \\
(c) \int_x \pi(x) x^2 g_x(x) dx = 0 &\rightarrow \int_x \pi(x) x^2 g_3(x) dx = 0.
\end{aligned}$$

which is the condition (5.8). ■

BIBLIOGRAPHY

Bibliography

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

[2] Yoshua Bengio. Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[3] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[4] Christopher M Bishop et al. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.

[5] P. Blaer and P. Allen. Topological mobile robot localization using fast vision techniques. In *Proceedings of the IEEE International Conference on Robotics and Automation,*, volume 1, pages 1031–1036, 2002.

[6] D. Bluestein, K. Dumont, M. De Beule, J. Ricotta, P. Impellizzeri, B. Verhegghe, and P. Verdonck. Intraluminal thrombus and risk of rupture in patient specific abdominal aortic aneurysm FSI modelling. *Computer Methods in Biomechanics and Biomedical Engineering*, 12(1):73–81, 2009.

[7] Francisco Bonin-Font, Alberto Ortiz, and Gabriel Oliver. Visual navigation for mobile robots: A survey. *Journal of intelligent and robotic systems*, 53(3):263–296, 2008.

[8] Olaf Booij, Zoran Zivkovic, and Ben Krose. Sparse appearance based modeling for robot localization. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1510–1515. IEEE, 2006.

[9] T. Botterill, S. Mills, and R. Green. Speeded-up bag-of-words algorithm for robot localisation through scene recognition. In *Image and Vision Computing New Zealand, 2008. IVCNZ 2008. 23rd International Conference*, pages 1–6, Nov 2008.

[10] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.

[11] Anthony R Brady, Simon G Thompson, F Gerald R Fowkes, Roger M Greenhalgh, Janet T Powell, et al. Abdominal aortic aneurysm expansion risk factors and time intervals for surveillance. *Circulation*, 110(1):16–21, 2004.

[12] A. Brooks, A. Makarenko, and B. Upcroft. Gaussian process models for indoor and outdoor sensor-centric robot localization. *IEEE Transactions on Robotics*, 24(6):1341–1351, 2008.

[13] Yu Cheng, Fei Wang, Ping Zhang, and Jianying Hu. Risk Prediction with Electronic Health Records: A Deep Learning Approach.

[14] Sungjoon Choi, Mahdi Jadaliha, Jongeun Choi, and Songhwai Oh. Distributed Gaussian Process Regression Under Localization Uncertainty. *Journal of Dynamic Systems, Measurement, and Control*, 137(3):031002, 2015.

[15] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[16] John J Craig. *Introduction to robotics: mechanics and control*, volume 3. Pearson Prentice Hall Upper Saddle River, 2005.

[17] Cédric Demonceaux, Pascal Vasseur, and Yohan Fougerolle. Central catadioptric image processing with geodesic metric. *Image and Vision Computing*, 29(12):840–849, 2011.

[18] Saandeep Depatla, Lucas Buckland, and Yasamin Mostofi. X-ray vision with only wifi power measurements using rytov wave models. *IEEE Transactions on Vehicular Technology*, 64(4):1376–1387, 2015.

[19] Guilherme N. DeSouza and Avinash C. Kak. Vision for mobile robot navigation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 24(2):237–267, 2002.

[20] Huan N Do, Jongeun Choi, Chae Young Lim, and Tapabrata Maiti. Appearance-based localization using Group LASSO regression with an indoor experiment. In *2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 984–989. IEEE, 2015.

[21] Huan N Do, Jongeun Choi, Chae Young Lim, and Tapabrata Maiti. Appearance-based outdoor localization using Group LASSO regression. In *2015 Dynamic Systems and Control (DSCC)*. IEEE, in press.

[22] Huan N Do, Mahdi Jadaliha, Mehmet Temel, and Jongeun Choi. Fully Bayesian Field Slam Using Gaussian Markov Random Fields. *Asian Journal of Control*, 2015.

[23] Vladislavs Dovgalecs, Rémi Mégret, and Yannick Berthoumieu. Multiple feature fusion based on co-training approach and time regularization for place classification in wearable video. *Advances in Multimedia*, 1, 2013.

[24] Stanimir Dragiev, Marc Toussaint, and Michael Gienger. Gaussian process implicit surfaces for shape estimation and grasping. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2845–2850. IEEE, 2011.

[25] Y. Dupuis, X. Savatier, and P. Vasseur. Feature subset selection applied to model-free gait recognition. *Image and Vision Computing*, 31(8):580 – 591, 2013.

[26] P Eriksson, S Jormsjö-Pettersson, AR Brady, H Deguchi, A Hamsten, and JT Powell. Genotype–phenotype relationships in an investigation of the role of proteases in abdominal aortic aneurysm expansion. *British Journal of Surgery*, 92(11):1372–1376, 2005.

[27] Mehdi Farsad, Shahrokh Zeinali-Davarani, Jongeun Choi, and Seungik Baek. Computational growth and remodeling of abdominal aortic aneurysms constrained by the spine. *Journal of Biomechanical Engineering*, 137(9):091008, 2015.

[28] Mehdi Farsad, Shahrokh Zeinali-Davarani, Jongeun Choi, and Seungik Baek. Computational growth and remodeling of abdominal aortic aneurysms constrained by the spine. *Journal of Biomechanical Engineering*, 137(9):091008–091008, 2015.

[29] João Carlos Aparício Fernandes and José Alberto B Campos Neves. Using conical and spherical mirrors with conventional cameras for 360 panorama views in a single image. In *Proceedings of the IEEE International Conference on Mechatronics*, pages 157–160, 2006.

[30] B. Ferris, D. Fox, and N. Lawrence. WiFi-SLAM using Gaussian process latent variable models. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2480–2485, 2007.

[31] Bernard Friedland. *Control system design: an introduction to state-space methods*. Courier Corporation, 2012.

[32] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.

[33] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *arXiv preprint arXiv:1506.02142*, 2, 2015.

[34] H Gharahi, B. A. Zambrano, C Lim, J Choi, W Lee, and S Baek. On growth measurements of abdominal aortic aneurysms using maximally inscribed spheres. *Medical Engineering & Physics*, 37(7):683–691, 2015.

[35] RB Gopaluni. A particle filter approach to identification of nonlinear processes under missing observations. *The Canadian Journal of Chemical Engineering*, 86(6):1081–1092, 2008.

[36] Alexander Graham. *Kronecker products and matrix calculus: with applications*, volume 108. Horwood England, UK, 1981.

[37] Ronald L. Graham. An efficient algorith for determining the convex hull of a finite planar set. *Information Processing Letters*, 1(4):132–133, 1972.

[38] Andrii Grytsan, Paul N Watton, and Gerhard A Holzapfel. A Thick-Walled Fluid–Solid-Growth Model of Abdominal Aortic Aneurysm Evolution: Application to a Patient-Specific Geometry. *Journal of Biomechanical Engineering*, 137(3):031008, 2015.

[39] Baofeng Guo and Mark S Nixon. Gait feature subset selection by mutual information. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 39(1):36–46, 2009.

[40] Efstathios Hadjidemetriou, Michael D. Grossberg, and Shree K. Nayar. Multiresolution histograms and their use for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7):831–847, 2004.

[41] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Applied statistics*, pages 100–108, 1979.

[42] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Applied Statistics*, pages 100–108, 1979.

[43] J.B. Hayet, F. Lerasle, and M. Devy. Visual landmarks detection and recognition for mobile robot navigation. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–313–II–318 vol.2, June 2003.

[44] Xuming He, Richard S Zemel, and Miguel Á Carreira-Perpiñán. Multiscale conditional random fields for image labeling. In *Computer vision and pattern recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE computer society conference on*, volume 2, pages II–695. IEEE, 2004.

[45] Tinne Tuytelaars Herbert Bay, Andreas Ess and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[46] Dave Higdon, James Gattiker, Brian Williams, and Maria Rightley. Computer model calibration using high-dimensional output. *Journal of the American Statistical Association*, 103(482):570–583, 2008.

[47] Geoffrey Hinton. A practical guide to training restricted Boltzmann machines. *Momentum*, 9(1):926, 2010.

[48] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[49] Geoffrey E Hinton. Learning multiple layers of representation. *Trends in cognitive sciences*, 11(10):428–434, 2007.

[50] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[51] Geoffrey E Hinton, Terrence J Sejnowski, and David H Ackley. *Boltzmann machines: Constraint satisfaction networks that learn.* Carnegie-Mellon University, Department of Computer Science Pittsburgh, PA, 1984.

[52] Wenhao Huang, Guojie Song, Haikun Hong, and Kunqing Xie. Deep architecture for traffic flow prediction: deep belief networks with multitask learning. *IEEE Transactions on Intelligent Transportation Systems*, 15(5):2191–2201, 2014.

[53] Daniel P. Huttenlocher, Gregory A. Klanderman, and William J Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.

[54] A Ijaz, J Choi, W Lee, and S Baek. Prediction of abdominal aortic aneurysms using sparse gaussian process regression. In *ASME 2013 Summer Bioengineering Conference*, pages V01BT28A011–V01BT28A011. American Society of Mechanical Engineers, 2013.

[55] Mahdi Jadaliha, Yunfei Xu, Jongeun Choi, Nicholas S Johnson, and Weiming Li. Gaussian process regression for sensor networks under localization uncertainty. *IEEE Transactions on Signal Processing*, 61(2):223–237, 2013.

[56] Gijeong Jang, Sungho Lee, and Inso Kweon. Color landmark based self-localization for indoor mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 1037–1042, 2002.

[57] Woo Yeon Jeong and Kyoung Mu Lee. Visual slam with line and corner features. In *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2570–2575. IEEE, 2006.

[58] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[59] Christopher Kanan and Garrison W Cottrell. Color-to-grayscale: does the method matter in image recognition? *PloS one*, 7(1):e29740, 2012.

[60] James M Keller, Michael R Gray, and James A Givens. A fuzzy k-nearest neighbor algorithm. *Systems, Man and Cybernetics, IEEE Transactions on*, (4):580–585, 1985.

[61] Ji-Hyun Kim. Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Computational Statistics & Data Analysis*, 53(11):3735–3745, 2009.

[62] Ahmed Klink, Fabien Hyafil, James Rudd, Peter Faries, Valentin Fuster, Ziad Mallat, Olivier Meilhac, Willem JM Mulder, Jean-Baptiste Michel, Francesco Ramirez, et al. Diagnostic and therapeutic strategies for small abdominal aortic aneurysms. *Nature Reviews Cardiology*, 8(6):338–347, 2011.

[63] HW Kniemeyer, T Kessler, P U Reber, H B Ris, H Hakki, and M K Widmer. Treatment of ruptured abdominal aortic aneurysm, a permanent challenge or a waste of resources? prediction of outcome using a multi-organ-dysfunction score. *European Journal of Vascular and Endovascular Surgery*, pages 190–196, 2000.

[64] Sadanori Konishi and Genshiro Kitagawa. Bayesian information criteria. *Information Criteria and Statistical Modeling*, pages 211–237, 2008.

[65] Nikolaos Kontopodis, Stella Lioudaki, Dimitrios Pantidis, George Papadopoulos, Efstratios Georgakarakos, and Christos V Ioannou. Advances in determining abdominal aortic aneurysm size and growth. *World Journal of Radiology*, 8(2):148, 2016.

[66] Nikolaos Kontopodis, Eleni Metaxa, Michalis Gionis, Yannis Papaharilaou, and Christos V Ioannou. Discrepancies in determination of abdominal aortic aneurysms maximum diameter and growth rate, using axial and orhtogonal computed tomography measurements. *European Journal of Radiology*, 82(9):1398–1403, 2013.

[67] J. Kosecka, Liang Zhou, P. Barber, and Z. Duric. Qualitative image based localization in indoors environments. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–3–II–8, 2003.

[68] Harrie Kurvers, Frank J Veith, Evan C Lipsitz, Takao Ohki, Nicholas J Gargiulo, Neal S Cayne, William D Suggs, Carlos H Timaran, Grace Y Kwon, Soo J Rhee, et al. Discontinuous, staccato growth of abdominal aortic aneurysms. *Journal of the American College of Surgeons*, 199(5):709–715, 2004.

[69] Matthew Lai. Deep learning for medical image segmentation. *arXiv preprint arXiv:1505.02000*, 2015.

[70] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178, 2006.

[71] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[72] Yann LeCun, Corinna Cortes, and Christopher JC Burges. The MNIST database of handwritten digits, 1998.

[73] Anne Long, Laurence Rouet, Jes S Lindholt, and Eric Allaire. Measuring the maximum diameter of native abdominal aortic aneurysms: review and critical analysis. *European Journal of Vascular and Endovascular Surgery*, 43(5):515–524, 2012.

[74] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[75] DavidG. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[76] G Martufi, M Lindquist Liljeqvist, N Sakalihasan, G Panuccio, R Hultgren, J Roy, and TC Gasser. Local Diameter, Wall Stress and Thrombus Thickness Influence the Local Growth of Abdominal Aortic Aneurysms. *European Journal of Vascular and Endovascular Surgery*, 48(3):349, 2014.

[77] Roland Memisevic and Geoffrey Hinton. Unsupervised learning of image transformations. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[78] Emanuele Menegatti, Takeshi Maeda, and Hiroshi Ishiguro. Image-based memory for robot navigation using properties of omnidirectional images. *Robotics and Autonomous Systems*, 47(4):251–267, 2004.

[79] Emanuele Menegatti, Mauro Zoccarato, Enrico Pagello, and Hiroshi Ishiguro. Image-based Monte Carlo localisation with omnidirectional images. *Robotics and Autonomous Systems*, 48(1):17–30, 2004.

[80] Christopher Z. Mooney. *Monte carlo simulation*. Number 116. Sage, 1997.

[81] Mark Nixon and Alberto S Aguado. *Feature extraction & image processing*. Academic Press, 2008.

[82] Guillaume Obozinski, Martin J Wainwright, Michael I Jordan, et al. Support union recovery in high-dimensional multivariate regression. *The Annals of Statistics*, 39(1):1–47, 2011.

[83] Naoya Ohnishi and Atsushi Imiya. Appearance-based navigation and homing for autonomous mobile robot. *Image and Vision Computing*, 31(6):511 – 532, 2013.

[84] Suguna Pappu, Alan Dardik, Hemant Tagare, and Richard J. Gusberg. Beyond fusiform and saccular: a novel quantitative tortuosity index may help classify aneurysm shape and predict aneurysm rupture potential. *Annals of Vascular Surgery*, 22:88–97, 2008.

[85] Kosmas I. Paraskevas, Dimitri P. Mikhailidis, Vassilios Andrikopoulos, Nikolaos Bessias, and Sir Peter R. F. Bell. Should the size threshold for elective abdominal aortic aneurysm repair be lowered in the endovascular era? Yes. *Angiology*, 61(7):617–619, 2010.

[86] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

[87] Pedro R. Peres-Neto, Donald A. Jackson, and Keith M. Somers. How many principal components? stopping rules for determining the number of non-trivial axes revisited. *Computational Statistics and Data Analysis*, 49(4):974 – 997, 2005.

[88] Ly Phan, Katherine Courchaine, Amir Azarbal, David Alan Vorp, Cindy Grimm, and Sandra Rugonyi. A geodesics-based surface parameterization to assess aneurysm progression. *Journal of Biomechanical Engineering*, 138(5):054503, 2016.

[89] Carol M Porth. *Essentials of pathophysiology: Concepts of altered health states*. Lippincott Williams & Wilkins, 2010.

[90] Carl Edward Rasmussen. *Gaussian processes for machine learning*. MIT Press, 2006.

[91] Samarth S Raut, Santanu Chandra, Judy Shum, and Ender A Finol. The role of geometric and biomechanical factors in abdominal aortic aneurysm rupture risk assessment. *Annals of Biomedical Engineering*, pages 1–19, 2013.

[92] Raimundo Real and Juan M Vargas. The probabilistic basis of Jaccard's index of similarity. *Systematic Biology*, pages 380–385, 1996.

[93] Robert Roesser. A discrete state-space model for linear image processing. *IEEE Transactions on Automatic Control*, 20(1):1–10, 1975.

[94] Eric Royer, Maxime Lhuillier, Michel Dhome, and Jean-Marc Lavest. Monocular vision for mobile robot localization and autonomous navigation. *International Journal of Computer Vision*, 74(3):237–260, 2007.

[95] David Salomon. *Data compression: the complete reference*. Springer Science & Business Media, 2004.

[96] Timo Schairer, Benjamin Huhle, Philipp Vorst, Andreas Schilling, and Wolfgang Straser. Visual mapping with uncertainty for correspondence-free localization using Gaussian process regression. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4229–4235, 2011.

[97] D Scharstein and A.J Briggs. Real-time recognition of self-similar landmarks. *Image and Vision Computing*, 19(11):763 – 772, 2001.

[98] S. Se, D.G. Lowe, and J.J. Little. Vision-based global localization and mapping for mobile robots. *Robotics, IEEE Transactions on*, 21(3):364–375, June 2005.

[99] Stephen Se, David Lowe, and Jim Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *The international Journal of robotics Research*, 21(8):735–758, 2002.

[100] A Sheidaei, S C Hunley, S Zeinali-Davarani, L G Raguin, and S Baek. Simulation of abdominal aortic aneurysm growth with updating hemodynamic loads using a realistic geometry. *Medical engineering & physics*, 33(1):80–88, 2011.

[101] Robert H Shumway and David S Stoffer. An approach to time series smoothing and forecasting using the EM algorithm. *Journal of Time Series Analysis*, 3(4):253–264, 1982.

[102] Robert Sim and Gregory Dudek. Comparing image-based localization methods. In *IJCAI*, pages 1560–1562, 2003.

[103] Eero P. Simoncelli and William T. Freeman. The steerable pyramid: a flexible architecture for multi-scale derivative computation. *2nd IEEE International Conference on Image Processing.*, 3(3):444–447, 1995.

[104] Bruno Sinopoli, Luca Schenato, Massimo Franceschetti, Kameshwar Poolla, Michael I Jordan, and Shankar S Sastry. Kalman filtering with intermittent observations. *Automatic Control, IEEE Transactions on*, 49(9):1453–1464, 2004.

[105] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[106] Arthur H Stroud and Don Secrest. Gaussian quadrature formulas. 1966.

[107] MJ Sweeting and SG Thompson. Making predictions from complex longitudinal data, with application to planning monitoring intervals in a national screening programme. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 175(2):569–586, 2012.

[108] SG Thompson, HA Ashton, L Gao, MJ Buxton, and RAP Scott. Final follow-up of the Multicentre Aneurysm Screening Study (MASS) randomized trial of abdominal aortic aneurysm screening. *British Journal of Surgery*, 99(12):1649–1656, 2012.

[109] Sebastian Thrun. Bayesian landmark learning for mobile robot localization. *Machine Learning*, 33(1):41–76, 1998.

[110] A. Torralba, K.P. Murphy, W.T. Freeman, and M.A. Rubin. Context-based vision system for place and object recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 273–280, 2003.

[111] Iwan Ulrich and Illah Nourbakhsh. Appearance-based obstacle detection with monocular color vision. In *Proceedings of the National Conference on Artificial Intelligence*, pages 866–871, 2000.

[112] C. Valgren and A. Lilienthal. Incremental spectral clustering and seasons: Appearance-based localization in outdoor environments. In *IEEE International Conference on Robotics and Automation, 2008. ICRA 2008.*, pages 1856–1861, May 2008.

[113] I. Vallivaara, J. Haverinen, A. Kemppainen, and J. Roning. Simultaneous localization and mapping using ambient magnetic field. In *Proceeding of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 14–19, September 2010.

[114] John Villadsen and Michael L Michelsen. *Solution of differential equation models by polynomial approximation*, volume 7. Prentice-Hall Englewood Cliffs, NJ, 1978.

[115] N. Vlassis, R. Bunschoten, and B. Krose. Learning task-relevant features from robot data. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 499–504, 2001.

[116] Gordon Wells and Carme Torras. Assessing image features for vision-based robot positioning. *Journal of Intelligent and Robotic Systems*, 30(1):95–118, 2001.

[117] Gordon Wells, Christophe Venaille, and Carme Torras. Vision-based robot positioning using neural networks. *Image and Vision Computing*, 14(10):715–732, 1996.

[118] J S Wilson, S Baek, and J D Humphrey. Importance of initial aortic properties on the evolving regional anisotropy, stiffness and wall thickness of human abdominal aortic aneurysms. *Journal of The Royal Society Interface*, 9(74):2047–2058, 2012.

[119] Niall Winters, José Gaspar, Gerard Lacey, and José Santos-Victor. Omni-directional vision for robot navigation. In *Proceedings. IEEE Workshop on Omnidirectional Vision, 2000.*, pages 21–28. IEEE, 2000.

[120] CF Jeff Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, pages 95–103, 1983.

[121] Jiacheng Wu and Shawn C Shadden. Coupled simulation of hemodynamics and vascular growth and remodeling in a subject-specific geometry. *Annals of Biomedical Engineering*, 43(7):1543–1554, 2015.

[122] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015.

[123] Y. Xu and J. Choi. Adaptive sampling for learning Gaussian processes using mobile sensor networks. *Sensors*, 11(3):3051–3066, March 2011.

[124] Yunfei Xu and Jongeun Choi. Adaptive sampling for learning Gaussian processes using mobile sensor networks. *Sensors*, 11(3):3051–3066, 2011.

[125] Byron A. Zambrano, Hamidreza Gharahi, ChaeYoung Lim, Farhad A. Jaberi, Jongeun Choi, Whal Lee, and Seungik Baek. Association of intraluminal thrombus, hemodynamic forces, and abdominal aortic aneurysm expansion using longitudinal CT images. *Annals of Biomedical Engineering*, 44(5):1502–1514, May 2016.

[126] A R Zankl, H Schumacher, U Krumsdorf, H A Katus, L Jahn, et al. Pathology, natural history and treatment of abdominal aortic aneurysms. *Clinical Research in Cardiology*, 96(3):140–151, 2007.

[127] S Zeinali-Davarani and S Baek. Medical image-based simulation of abdominal aortic aneurysm growth. *Mechanics Research Communications*, 42:107–117, 2012.

[128] Shahrokh Zeinali-Davarani, Azadeh Sheidaei, and Seungik Baek. A finite element model of stress-mediated vascular adaptation: application to abdominal aortic aneurysms. *Computer Methods in Biomechanics and Biomedical Engineering*, 14(9):803–817, 2011.

[129] Yi Zhou, Yan Wan, Sandip Roy, Clark Taylor, Craig Wanke, Dinesh Ramamurthy, and Junfei Xie. Multivariate probabilistic collocation method for effective uncertainty evaluation with application to air traffic flow management. *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, 44(10):1347–1363, 2014.

[130] Feng Zhuge, Geoffrey D Rubin, Shaohua Sun, and Sandy Napel. An abdominal aortic aneurysm segmentation method: level set with region and statistical information. *Medical Physics*, 33(5):1440–1453, 2006.