Ergodic Coverage and Active Search in Constrained Environments

Hadi Salman

CMU-RI-TR-18-11
May 2018

Thesis Committee:
Howie Choset, Co-chair
Matthew Travers, Co-chair
Michael Kaess
Arun Srivatsan Rangaprasad

Submitted in partial fulfillment of the requirements for the degree of
Masters of Science in Robotics

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

In this thesis, we explore sampling-based trajectory planning applied to the task of searching for objects of interest in constrained environments (e.g., a UAV searching for a target in the presence of obstacles). We consider two search scenarios: in the first scenario, accurate prior information distribution of the possible locations of the objects of interest is available, thus we cover this information distribution until we find the target. In the second scenario, no or crude prior information distribution exists about the locations of the objects of interest, thus we actively search for those objects by updating the information distribution.

For the first search scenario we explore ergodic coverage. Ergodic coverage is an approach for trajectory planning in which a robot is directed such that the percentage of time spent in a region is in proportion to the probability of locating targets in that region. We extend the ergodic coverage algorithm to robots operating in constrained environments and present a formulation that can capture sensor footprints and avoid obstacles in the domain. We demonstrate through simulation that our formulation easily extends to coordination of multiple robots equipped with different sensing capabilities to perform ergodic coverage of a domain.

For the second search scenario, we investigate palpation-based tumor search and develop an approach that guides robots to automatically localize and find the shapes of tumors and other stiff inclusions present in anatomies. Our approach uses Gaussian processes to model the stiffness distribution, and active learning integrated with sampling-based trajectory planning to direct the palpation path of the robot. Our approach provides the flexibility to avoid obstacles in the search domain, incorporate uncertainties in robot position and sensor measurements, and include prior information about the locations of stiff inclusions while respecting the robot kinematic constraints. The proposed framework is evaluated via simulation and experimentation on three different robot platforms: 6-DoF industrial arm, da Vinci Research Kit (dVRK), and the Insertable Robotic Effector

Platform (IREP). Results show that our approach can accurately estimate the locations and boundaries of stiff inclusions while reducing exploration time.

Acknowledgement

I would like to start by thanking God for the grace of completing my masters degree. I am very thankful to my advisors, Professor Howie Choset and Professor Matthew Travers, for their amazing guidance and support over the period of two years. Special thanks to Elif Ayvali who was a postdoctoral researcher in our lab, and who mentored me during my first year. Elif helped me shape my thesis, and her fantastic work, guidance, ideas, and support were crucial for me to successfully finish this thesis. I would also like to thank Arun Srivatsan and Nicolas Zevallos for their help and support - every experiment in this thesis has these brilliant guys as part of it. I am very thankful to my friends in the Biorobotics laboratory who always supported me and made my life much easier with their big hearts and wonderful smiles. Thanks for Professor Michael Kaess for providing me with valuable feedback for this thesis document, and for being on my committee. Furthermore, I would like to thank people in the ARMA lab at Vanderbilt university and in the LCSR lab at Johns Hopkins university for the great benefits I gained through collaborating with them. Finally, I thank my family who have always loved and supported me especially during the two years of my masters.

Contents

Chapter 1

Introduction

Consider the scenario where a group of unmanned aerial vehicles (UAVs) are searching for targets as shown in Fig. 1.1[1]. The UAVs collectively maintain a belief distribution of the locations of the targets which is overlaid in blue. We are interested in figuring out how to plan trajectories for these UAVs in order to efficiently find the targets. In this thesis, we explore sampling-based techniques for robot trajectory planning for the sake of finding objects of interests in a domain. Throughout this thesis, we consider two types of search that bascially differ in the nature of information that is available to the agents.

The first type of search applies to scenarios in which accurate prior information distribution of the possible locations of the objects of interest is available and is static. Thus, in order to find a target of interest, one would favor covering regions where the prior distribution indicates high likelihood of locating the target. Probabilistic approaches were developed to exploit this prior information to direct the search to take less time [1–3]. Some of these approaches incorporate a coverage metric, called ergodicity, as an objective to guide the exploration with respect to a desired spatial distribution of robot trajectories [4, 5]. For example, given a probability density function (PDF) defined over the search domain, and indicating where targets of interest might be located, an ergodic coverage strategy directs the robots to distribute their time searching regions of the domain in proportion to the probability of locating targets in those regions as shown in Fig. 1.1.

The second type of search that we consider in this thesis applies to scenarios where no or crude prior information distribution exists about the locations of the objects of interest. Therefore, the agents actively search for those objects by collecting sensor measurements, updating an information distribution that encodes the locations of the targets of interest, and using this information distribution to guide the search of where to go and collect sensor measurements next for efficient search. As a focal application, we consider palpation-based tumor search where a robot is equipped with a force sensor and is allowed to probe an organ and collect stiffness measurements in order to locate stiff inclusions such as tumors as shown in Fig. 1.2. The robot is assumed to have no or crude prior information of the locations of tumors. This is essential when performing minimally invasive surgery (MIS) where often there is a loss of haptic understanding of the anatomy. In order to restore the lost information, several works in literature have focused on developing miniature tactile and force sensors [6–11]. Surgeons typically rely on palpation to develop a haptic understanding of the anatomy [12]. They analyze the force and deflection feedback from palpation to localize tumors and

[1]The figure is taken from Guillaume Sartoretti's website: http://guillaume.sartoretti.science/

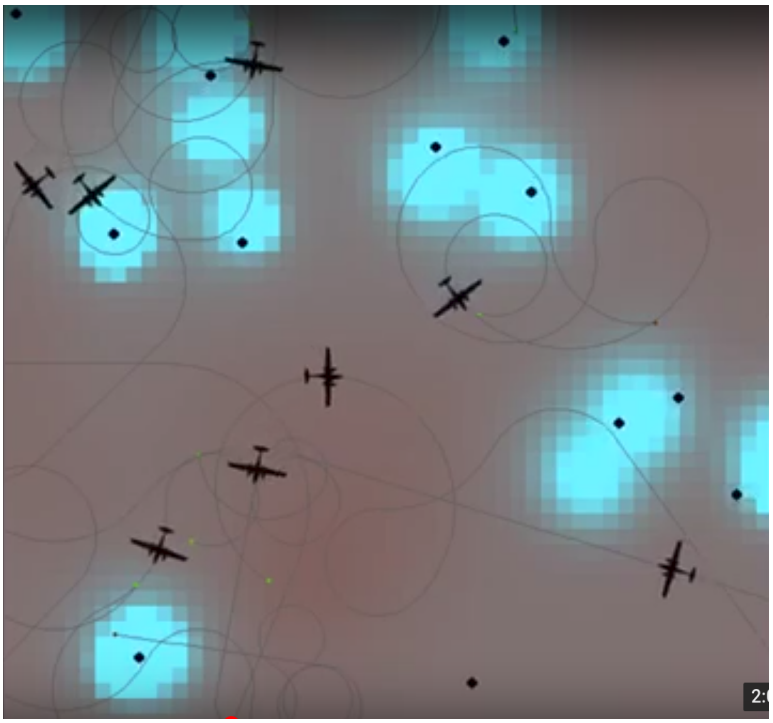

Figure 1.1: Multiple agents (aeroplanes) performing ergodic coverage to search for and locate targets (black dots). The agents start with covering a prior distribution of the locations of targets (uniform distribution), and update this distribution as they observe new targets. The belief distribution of the locations of targets (which is the desired distribution to be covered by the agents) is overlaid on the above map in cyan color.

sensitive anatomy such as nerve bundles, tendons and arteries. Information from palpation can help surgeons in forming a better understanding of the surgical scene and in achieving a correlation between pre-operative imaging information and the surgical scene. Thus, it is important to devise strategies that use these sensory information in an efficient way to help surgeons build a better understanding of the anatomy. Such strategies include methods that guide the robot where to go and probe next for fast and efficient tumor localization, and are referred to as *active search* methods [13, 14].

This thesis explores and extends techniques for ergodic coverage and active search. The first part of the thesis develops a framework for sampling-based ergodic coverage framework that tackles limitations of previous ergodic coverage algorithms. The second part of the thesis presents a trajectory optimized active search framework that integrates state-of-the-art active search methods with sampling-based trajectory planning with application on palpation-based tumor search[2].

[2]Supplementary videos for this thesis can be found at this link. A presentation of this thesis can be found at the following link.

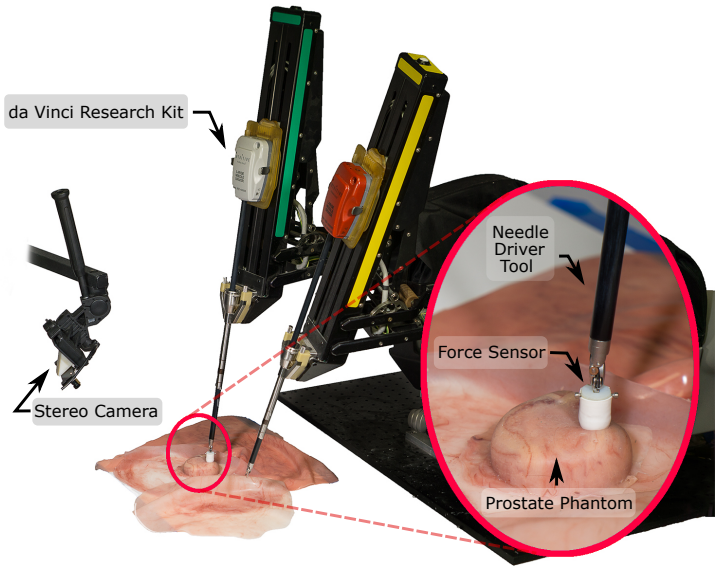

Figure 1.2: Experimental setup showing the dVRK robot with a miniature force sensor attached to the end-effector. A stereo camera overlooks the workspace of the robot. A phantom prostate with embedded stiff inclusion is placed in the workspace of the robot.

1.1 Related Work

In this section, we present some of the work that exists in the literature involving ergodic coverage and active tumor search.

1.1.1 Ergodic Coverage

Ergodic theory is the statistical study of time-averaged behavior of dynamical systems [15]. This useful notion has been previously used to distribute multiple agents in an exploration domain such that agents' trajectories cover the domain. Formally, ergodicity is measured with respect to a given target probability distribution. For example, to equally favor all the possible states in the exploration domain, one needs to specify the target probability distribution as a uniform distribution.

What makes ergodic coverage useful is that it overcomes the drawbacks of alternative uniform coverage algorithms, such as the lawnmower algorithm in which agents scan an area by going back and forth in parallel lines [16–19]; ergodic coverage can be easily implemented for irregular domains and non-uniform target probability distributions. Also, it assumes perfect communication among agents. Agents' motions are planned sequentially by taking into account the motion of every other agent. Therefore, it is robust to agent failure and to adding new agents. Furthermore, ergodic coverage is naturally multi-scale i.e., covers large scale regions followed by smaller regions [4].

The ergodic coverage algorithm, originally formulated by Mathew and Mezić [4], applies to simple kinematic systems and has a natural extension to centralized multi-robot systems. The formulation, however, treats a robot as point mass, and robot's sensor footprint is represented as a Dirac delta function for computational tractability. Additionally, previous imple-

mentations of the algorithm in [4,5] do not consider the obstacles in the exploration domain. To exclude the coverage of certain areas, Mathew and Mezic [4] set the value of the desired spatial distribution of trajectories to zero in those areas. However, the robots still visit those areas although the percentage of time spent is small. We have recently employed a potential field-based approach to extend the formulation in [4], which considers first-order and second-order systems, to avoid circular obstacles in a planar domain [20]. However, generalization of the derivations for other systems and types of obstacles are nontrivial. Constraints corresponding to arbitrary obstacles can be non-smooth, and, as a consequence, gradient-based optimization methods employed in aforementioned works require special differentiation to guarantee convergence of the solution [21].

1.1.2 Tumor Search

One of the active search applications that we consider in this thesis is tumor search. Many groups have looked into using surgical robots for autonomously exploring an organ with discrete probing motion [22, 23], rolling motion [24] and cycloidal motion [25] to localize tumors which are characterized by high stiffness compared to that of normal tissue. These works commonly direct the robot along a predefined path that scans the entire organ or region of interest [22, 23, 26, 27]. Some of these works [25, 28] use adaptive grid resolution to increase palpation resolution around boundaries of regions of interest marked by high stiffness gradients [25, 28].

In order to reduce the exploration time, Bayesian optimization-based approaches have been developed for tumor localization by directing the exploration to stiff regions [12,29–32]. These approaches model tissue stiffness as a distribution defined on the surface of the organ where each point on the surface is associated with a random variable. Bayesian optimization is then used to find the global maxima of the stiffness distribution. The assumption is that finding the global maxima of the stiffness distribution correspond to locating the stiff inclusions, portions of the tissue that probably does not belong. Ayvali *et al.* [29] sequentially select the next location to probe the organ, and predict the stiffness distribution and the location of the global maximum after every measurement, while Chalasani *et al.* [30] update after collecting several samples over finite time along a trajectory that directs the robot to the high stiffness regions. In a more recent work, Chalasani *et al.* [32] incrementally estimate local stiffness and geometry while the organ is palpated along predefined trajectories or under telemanipulation. Garg *et al.* [12] direct the exploration to areas where the predicted stiffness values are within a percentage of the current estimated maximum to favor locations around the maximum and not just at the expected maximum.

However, none of these approaches explicitly encode the goal of extracting the boundary of the stiff inclusion. The only goal that is encoded through a Bayesian optimization framework is to find the global maximum. As a consequence, the robot ends up mainly exploring around high stiffness regions before expanding to the boundary of the inclusion and other regions. Prior works commonly demonstrate results using a single stiff inclusion (single maximum) [12,30]. When multiple inclusions are present (multiple global and local maxima) the Bayesian optimization algorithm is initialized with a coarse grid to ensure exploration of all regions [29].

1.2 Approach and Contributions

There are two key contributions of this thesis: (1) development of a sampling-based ergodic coverage framework that tackles the limitations of previous frameworks [33] [34], and (2) development of a trajectory-optimized active search framework for tumor localization.

1.2.1 Ergodic Coverage in Constrained Environments

We construct a new ergodic coverage objective that can take into account the robots' sensor footprint. We also formulate the ergodic coverage algorithm within a sampling-based trajectory planning framework that lends itself to obstacle avoidance. Our formulation allows a robot with a wide sensor footprint to perform coarse ergodic coverage of the domain, while a robot with narrow sensor footprint performs dense ergodic coverage of the same domain.

1.2.2 Trajectory Optimized Active Search for Tumor Localization

We develop a tumor search framework that leverages state-of-the-art active search methods as the objective to optimize robot's trajectories and explicitly encodes search of stiff regions and their boundaries. Compared to the existing works on active search [13, 35], our formulation also incorporates constraints due to the robot's motion model and restricted areas in the search domain. We show experimental results with a variety of robotic platforms both using discrete probing and palpation along a continuous path that is optimized using sampling-based trajectory planning.

Chapter 2

Background

2.1 Ergodic Coverage

Mathew and Mezić [4] introduced a metric to quantify the ergodicity of a robot's trajectory with respect to a given PDF. They start by defining the time-average statistics distribution Γ of a robot's trajectory, $\gamma : (0,t] \to X$. This distribution quantifies the fraction of time spent at a point, $\boldsymbol{x} \in X$, where $X \subset \mathbb{R}^d$ is a d-dimensional domain, and is defined as

$$\Gamma_t(\boldsymbol{x}) = \frac{1}{t}\int_0^t \delta(\boldsymbol{x} - \gamma(\tau))d\tau, \tag{2.1}$$

where δ is the Dirac delta function.

Mathew and Mezić [4] define a PDF $\xi(\boldsymbol{x})$ – also referred to as the desired coverage distribution – defined over the domain. The ergodicity of a robot's trajectory with respect to $\xi(\boldsymbol{x})$ is then defined as

$$\Phi(t) = \sum_{k=0}^{m} \lambda_k \left|\Gamma_k(t) - \xi_k\right|^2, \tag{2.2}$$

where $\lambda_k = \frac{1}{(1+|k|)^s}$ is a coefficient that places higher weights on the lower frequency components and $s = \frac{d+1}{2}$. The $\Gamma_k(t)$ and ξ_k are the Fourier coefficients of the distributions $\Gamma_t(\boldsymbol{x})$ and $\xi(\boldsymbol{x})$ respectively, i.e.,

$$\begin{aligned} \Gamma_k(t) &= \langle \Gamma, f_k \rangle = \frac{1}{t}\int_0^t f_k(\gamma(\tau))d\tau \\ \xi_k &= \langle \xi, f_k \rangle = \int_X \xi(\mathbf{x}) f_k(\mathbf{x}) d\mathbf{x} \end{aligned} \tag{2.3}$$

where $f_k(\boldsymbol{x}) = \frac{1}{h_k}\prod_{i=1}^{m} \cos\left(\frac{k_i \pi}{L_i} x_i\right)$ is the Fourier basis functions that satisfy Neumann boundary conditions on the domain X, $m \in \mathbb{Z}$ is the number of basis functions, and $\langle \cdot, \cdot \rangle$ is the inner product with respect to the Lebesgue measure. The term h_k is a normalizing factor.

We note that in the scenario where multiple robot are covering a domain, the above equations still hold with minor modifications. Consider N robots, the time average statistics distribution shared among these robots is defined as

$$\Gamma_t(\boldsymbol{x}) = \frac{1}{Nt}\sum_{j=1}^{N}\int_0^t \delta(\boldsymbol{x} - \gamma_j(\tau))d\tau$$

and the corresponding Fourier coefficients are calculated as,

$$\Gamma_k(t) = \frac{1}{Nt} \int_0^t \sum_{j=1}^{N} f_k(\gamma_j(\tau)) d\tau$$

The goal of ergodic coverage is to generate optimal controls $\boldsymbol{u}^*(t)$ for a robot, whose dynamics is described by a function $\dot{\boldsymbol{q}}(t) = g(\boldsymbol{q}(t), \boldsymbol{u}(t))$ such that

$$\begin{aligned} \boldsymbol{u}^*(t) &= \arg\min_{\boldsymbol{u}} \Phi(t), \\ \text{subject to } \dot{\boldsymbol{q}}(t) &= g(\boldsymbol{q}(t), \boldsymbol{u}(t)) \\ \|\boldsymbol{u}(t)\| &\leq u_{max}, \end{aligned} \tag{2.4}$$

where $\boldsymbol{q} \in Q$ is the state space and $\boldsymbol{u} \in U$ denotes the set of controls. Note that $\Phi(t)$ has $\gamma(t)$ encoded in, and $\gamma(t)$ is the actual robot trajectory described by the robot dynamics, which in turn is a function of $\boldsymbol{u}(t)$.

Mathew and Mezić [4] mainly consider first-order, $\dot{\boldsymbol{q}}(t) = \boldsymbol{u}(t)$, and second-order systems, $\ddot{\boldsymbol{q}}(t) = \boldsymbol{u}(t)$. They derive closed-form solutions that approximate the solution to Eq. (2.4) by discretizing the exploration time and solving for the optimal control input that maximizes the rate of decrease of Eq. (2.4) at each time-step. Miller and Murphey [5] use a projection-based trajectory optimization method that solves a first-order approximation of Eq. (2.2) at each iteration using linear quadratic regulator techniques [36].

2.2 Cross-Entropy Trajectory Planning

Cross-Entropy trajectory planning is a sampling-based method that was developed by Kobilarov [37] for path planning. This method is based on sampling in the space of trajectories. The idea is to construct a probability distribution over the set of feasible paths and to perform the search for an optimal trajectory through importance sampling [38], which is an efficient technique to estimate a complex distribution by sampling from another easy-to-sample-from distribution.

Consider a robot whose dynamics is described by the function $g\colon Q \times U \to TX$, such that

$$\dot{\boldsymbol{q}}(t) = g(\boldsymbol{q}(t), \boldsymbol{u}(t)) \tag{2.5}$$

In addition, the robot is subject to constraints such as actuator bounds and obstacles in the environment. These constraints are expressed through the vector of inequalities

$$F(\boldsymbol{q}(t)) \geq 0, \tag{2.6}$$

The goal is to compute the optimal controls $\boldsymbol{u}^*$ over a time horizon $t \in (0, t_f]$ that minimize a cost function such that

$$\begin{aligned} \boldsymbol{u}^*(t) &= \arg\min_{\boldsymbol{u}} \int_0^{t_f} C(\boldsymbol{u}(t), \boldsymbol{q}(t)) dt, \\ \text{subject to } \dot{\boldsymbol{q}}(t) &= g(\boldsymbol{q}(t), \boldsymbol{u}(t)), \\ F(\boldsymbol{q}(t)) &\geq 0, \\ \boldsymbol{q}(0) &= \boldsymbol{q}_0, \end{aligned} \tag{2.7}$$

where $\boldsymbol{q_0}$ is the initial state af the robot and $C\colon U \times Q \to \mathbb{R}$ is a given cost function.

2.2.1 Trajectory Parameterization

Following the notation in [37], a trajectory defined by the controls and states over the time interval $[0,T]$ is denoted by the function $\pi\colon [0,T] \to U \times Q$, i.e. $\pi(t) = (\boldsymbol{u}(t), \boldsymbol{q}(t))$ for all $t \in [0,T]$. The space of all trajectories originating at $\boldsymbol{q}_0$ and satisfying Eq. (2.5) is given by

$$P = \{\pi\colon t \in [0,T] \to (\boldsymbol{u}(t), \boldsymbol{q}(t)) \,|\dot{\boldsymbol{q}}(t) = g(\boldsymbol{q}(t), \boldsymbol{u}(t)), \\ \boldsymbol{q}(0) = \boldsymbol{q}_0, T > 0.\} \tag{2.8}$$

Let us consider a finite-dimensional parameterization of trajectories in terms of vectors $\boldsymbol{z} \in Z$ where $Z \subset \mathbb{R}^{n_z}$ is the parameter space. Assuming that the parameterization is given by a function $\varphi : Z \to P$ according to $\pi = \varphi(z)$, the $(\boldsymbol{u}, \boldsymbol{q})$ tuples along a trajectory parameterized by $\boldsymbol{z}$ are written as $\pi(t) = \varphi(\boldsymbol{z}, t)$. One choice of parameterization is to use motion primitives defined as $\boldsymbol{z} = (\boldsymbol{u}_1, \tau_1, ..., \boldsymbol{u}_j, \tau_j)$ where each $\boldsymbol{u}_i$, for $1 \leq i \leq j$, is a constant control input applied for duration τ_i. Another option is to use RRT or probabilistic roadmaps (PRM) to sample trajectories as demonstrated in [37].

Now, we can define a cost function, $J\colon Z \to \mathbb{R}$, in terms of the trajectory parameters as

$$J(\boldsymbol{z}) = \int_0^T C(\varphi(\boldsymbol{z}, t))dt \tag{2.9}$$

Eq. (2.7) can be restated as finding the optimal $(\boldsymbol{u}^*, \boldsymbol{q}^*) = \varphi(\boldsymbol{z}^*)$ such that

$$\boldsymbol{z}^* = \underset{\boldsymbol{z} \in Z_{con}}{\arg\min} J(\boldsymbol{z}). \tag{2.10}$$

where the constrained parameter space $Z_{con} \subset Z$ is the set of parameters that satisfy the boundary conditions and constraints in Eq. (2.7).

2.2.2 Cross-Entropy Method

In this thesis, we follow the work of Kobilarov [37] and employ the cross entropy (CE) method to optimize the parameters of the trajectory. There are other sampling-based global optimization methods such as Bayesian optimization [39], simulated annealing [40], and other variants of stochastic optimization [41] that can also be used to optimize parameterized trajectories. We use the CE method because it utilizes importance sampling to efficiently compute trajectories that have lower costs after few iterations of the algorithm, and it has been shown to perform well for trajectory optimization of nonlinear dynamic systems [37].

The CE method treats the optimization in Eq. (2.10) as an estimation problem of rare-event probabilities. The rare event of interest in this work is to find a parameter z whose cost $J(z)$ is very close to the cost of an optimal parameter z^*. It is assumed that the parameter $z \in Z$ is sampled from a Gaussian mixture model defined as

$$p(z; v) = \sum_{k=1}^{K} \frac{w_k}{\sqrt{(2\pi)^{n_z}|\Sigma_k|}} e^{-\frac{1}{2}(z-\mu_k)^T \sigma_k^{-1}(z-\mu_k)} \tag{2.11}$$

where $v = (\mu_1, \Sigma_1, ..., \mu_K, \Sigma_K, w_1, ..., w_K)$ corresponds to K mixture components with means μ_k, covariance matrices Σ_k, and weights w_k, where $\sum_{k=1}^{K} w_k = 1$.

The CE method involves an iterative procedure where each iteration has two steps: *(i)* select a set of parameterized trajectories from $p(z;v)$ using importance sampling [42] and evaluate the cost function $J(z)$, (ii) use a subset of *elite* trajectories[1] and update v using expectation maximization [43]. After a finite number of iterations $p(z;v)$ approaches to a delta distribution, thus the sampled trajectories remains unchanged. For implementation details, the reader is referred to [44].

2.3 Gaussian Process Regression

Gaussian process (GP) is a popular tool used to perform nonparameteric regression. Intuitively a GP can be viewed as a distribution over functions. By using GP, we assume a smooth change in the latent function we are estimating (in this thesis the stiffness distribution across the organ).

A GP is defined by its mean and covariance functions f_{GP} and k respectively. Given a d-dimensional search domain $X \subset \mathbb{R}^d$, the distribution of function values at a point $\boldsymbol{x} \in X$ is represented by a random variable, y, and has a Gaussian distribution, $N(f_{GP}(\boldsymbol{x}), \sigma^2(\boldsymbol{x}))$ where we abbreviate $\sigma^2(\boldsymbol{x}) = k(\boldsymbol{x},\boldsymbol{x})$. Given a set of n observations $\bar{\boldsymbol{y}} = [y_1, y_2, \ldots, y_n]^T$ at $\bar{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]^T$, GP regression can be used to make predictions on the distribution of function values at a new point $\boldsymbol{x}_* \in X$

$$p(y_*|\bar{\boldsymbol{y}}) \sim N(\boldsymbol{K}_*\boldsymbol{K}^{-1}\bar{\boldsymbol{y}}, k_{**} - \boldsymbol{K}_*\boldsymbol{K}^{-1}\boldsymbol{K}_*^{\boldsymbol{T}}),$$

where $k_{**} = k(\boldsymbol{x}_*, \boldsymbol{x}_*)$, and $\boldsymbol{K}$ is the $n \times n$ covariance matrix whose elements $\boldsymbol{K}_{ij}$ are calculated using any positive definite covariance function $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ (in this thesis, we use the squared exponential covariance function). Similarly, $\boldsymbol{K}_*$ is a $1 \times n$ vector defined as,

$$\boldsymbol{K}_* = [k(\boldsymbol{x}_*, \boldsymbol{x}_1), \ldots, k(\boldsymbol{x}_*, \boldsymbol{x}_n)].$$

2.4 Active Learning and Bayesian Optimization

In many learning scenarios, unlabeled data are plentiful and manually labeling them is expensive. The role of active learning algorithms is to efficiently find which data to label. In this work, we leverage active learning algorithm to direct a robot where to search for tumor. The search space is the surface of the organ and labeling data corresponds to assigning a binary value to every point on the organ's surface: normal tissue vs. tissue abnormality depending on the measured stiffness value at that point. Thus, the active learning algorithms guides the robot of where to go, probe, and collect stiffness next. More details are presented in Section 4.2. We consider in this work various active learning algorithms: active area search (AAS), active level sets estimation (LSE), and uncertainty sampling (UNC), and compare them with Bayesian optimization algorithm (BOA) which gained interest in recent works [12, 30, 31].

2.4.1 Active Area Search

AAS discretizes the search domain into a set of regions G = $\{g_i | i = 1 \ldots N, g_i \subset X\}$ and classifies each as region-of-interest (tissue abnormalities corresponding to regions that have

[1] A fraction of the sampled trajectories with the best costs form an elite set. See [42] for details.

high stiffness) if the average estimated latent function (stiffness function in our case) is above some threshold τ with high probability θ. AAS sequentially queries at a point $\boldsymbol{x}_* \in X$ that maximizes the expected sum of binary rewards r_g defined over each region $g \in G$ as,

$$r_g = \begin{cases} 1, & \text{if } T_g > \theta \\ 0, & \text{otherwise} \end{cases}$$

where y_* is the observation at $\boldsymbol{x}_*$, and $T_g = p\left(f_g > \tau | \bar{X}, \bar{\boldsymbol{y}}, (\boldsymbol{x}_*, y_*)\right)$. f_g is the average area integral of f_{GP} over the region g and is defined as,

$$f_g = \frac{1}{A_g} \int_g f_{GP}(\boldsymbol{x}) d\boldsymbol{x}$$

where A_g is the area of g. Thus, AAS sequentially samples the point $\boldsymbol{x}_*$ that maximizes the expected total reward, i.e.,

$$\begin{aligned} \boldsymbol{x}_* &= \arg\max_{\boldsymbol{x} \in X} \sum_g \mathbb{E}[r_g | \bar{X}, \bar{\boldsymbol{y}}, (\boldsymbol{x}, y)] \\ &= \arg\max_{\boldsymbol{x} \in X} \sum_g p\big(T_g > \theta\big) \end{aligned} \tag{2.12}$$

For more details, we refer the reader to the work of Ma *et al.* [13].

2.4.2 Level Set Estimation

The LSE algorithm determines the set of points for which an unknown function (stiffness map in our case) takes a value above or below some given threshold level h. LSE guides both sampling and classification based on GP-derived confidence bounds. The mean and covariance of the GP can be used to define a confidence interval,

$$Q_t(\boldsymbol{x}) = \left[f_{GP_t}(\boldsymbol{x}) \pm \beta^{1/2} \sigma_t(\boldsymbol{x}) \right]$$

for each point $\boldsymbol{x} \in \bar{X}$, where the subscript t refers to time. Furthermore, a confidence region C_t which results from intersecting successive confidence intervals can be defined as,

$$C_t(\boldsymbol{x}) = \bigcap_{i=1}^{t} Q_i(\boldsymbol{x})$$

LSE then defines a measure of classification ambiguity $a_t(\boldsymbol{x})$ defined as,

$$a_t(\boldsymbol{x}) = \min\left\{ \max(C_t(\boldsymbol{x})) - h, h - \min(C_t(\boldsymbol{x})) \right\} \tag{2.13}$$

LSE chooses sequentially queries (probes) at $\boldsymbol{x}_*$ such that,

$$\boldsymbol{x}_* = \arg\max_{\boldsymbol{x} \in X} a_t(\boldsymbol{x}).$$

For details and how to select the parameter h, we refer the reader to the work of Gotovos *et al.* [35].

2.4.3 Uncertainty Sampling

The Uncertainty Sampling (UNC) algorithm explores locations that have high variance in the GP posterior distribution [45]. The algorithm is very simple; it sequentially queries (probes) at $\boldsymbol{x}_*$ such that,

$$\boldsymbol{x}_* = \arg\max_{\boldsymbol{x} \in X} \sigma(\boldsymbol{x}).$$

where $\sigma(\boldsymbol{x})$ is the standart deviation of th GP estimate of the latent function at location $\boldsymbol{x}$.

2.4.4 Bayesian Optimization

Bayesian Optimization algorithm (BOA) is a sequential sampling strategy for finding the global maxima of black-box functions [46]. A GP is used as a surrogate for the function to be optimized. BOA uses the posterior mean, $f_{GP}(\boldsymbol{x})$, and variance, $\sigma^2(\boldsymbol{x})$, of the GP for all $\boldsymbol{x} \in X$, to sequentially select the next best sample as the point that maximizes an objective function such as expected improvement (EI) [46] given by,

$$\begin{aligned} \boldsymbol{x}_* &= \arg\max_{\boldsymbol{x} \in X} EI(\boldsymbol{x}) \\ EI(\boldsymbol{x}) &= \begin{cases} (f_{GP}(\boldsymbol{x}) - y^+)\Phi(z) + \sigma(\boldsymbol{x})\phi(z) & \text{if } \sigma(\boldsymbol{x}) > 0 \\ 0 & \text{if } \sigma(\boldsymbol{x}) = 0 \end{cases} \end{aligned} \tag{2.14}$$

where $z = \left(\frac{f_{GP}(\boldsymbol{x}) - y^+}{\sigma(\boldsymbol{x})}\right)$, y^+ is the current maximum. $\phi(\cdot)$ and $\Phi(\cdot)$ are the probability density function and cumulative distribution function of the standard normal distribution, respectively.

Chapter 3

Ergodic Coverage in Constrained Environments

In this chapter, we introduce a formulation that optimizes an ergodic coverage objective using the cross entropy trajectory planning algorithm presented in Section 2.2. In this formulation, which we will refer to as *stochastic trajectory optimization for ergodic coverage* (STOEC), we incorporate kinematic constraints of the robot and constraints arising from obstacles in the enivornement, and we pose ergodic coverage as a sampling-based trajectory planning problem. We choose a sampling-based technique because it is easy to implement, it can be used as a trajecory optimization method, it does not assume any differentiable properties of the objective functions for which it optimizes, and it can return a solution as fast as required by reducing the number of samples, thus giving away some accuray for speed [37]. We start by redefining the time average statistics of a robot's trajectory to capture the robot's sensor footprint. We then demonstrate that obstacle avoidance can be easily incorporated into this framework by penalizing sampled trajectories that collide with arbitrarily-shaped obstacles or that intersect restricted regions.

3.1 Redefinition of the Metric of Ergodicity

Let us start with two key observations. First, we note that in Eq. (2.1), it is assumed that a robot's trajectory is composed of Dirac delta functions. In other words, it is assumed that the sensor footprint, which is basically the region of the domain covered by the sensors of the robot at a given location, is negligible compared to the size of the domain. This assumption is adopted to make the calculation of the Fourier coefficients of the distributions given by Eq. (2.3) simple [47]. In many practical applications, the robots can be equipped with sensors of various types (camera, lidar, ultrasonic, etc.), which cannot really be modelled using the Dirac delta function as a footprint of the robot. Having control over the sensor footprint opens up interesting and useful strategies for coverage: a robot equipped with a wide sensor footprint should be able to perform a coarse ergodic coverage of the domain compared to a robot equipped with a narrow sensor footprint. Similarly, the optimal ergodic-coverage trajectory of a ground vehicle equipped with a forward looking ultrasonic sensor would be different than an aerial vehicle equipped with a downward-looking sensor.

Second, as also stated in [5], the motivation in using the norm of the Fourier coefficients is to obtain an objective function that is differentiable with respect to the trajectory. When

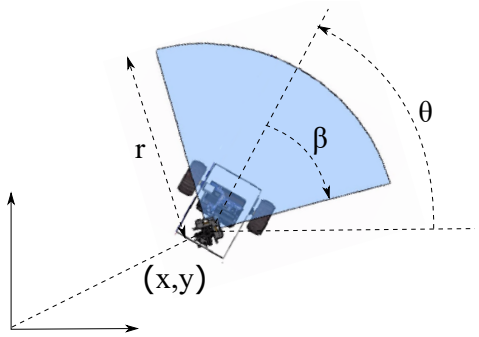

Figure 3.1: Forward looking beam sensor model (e.g. ultrasonic sensor): an example of a sensor footprint of the robot. The variables (x, y, θ) represent the state of the robot, whereas r and β represent the bearing measurement limits of the sensor.

a sampling-based approach is used for trajectory optimization, there is no need for a differentiable objective function, but this comes at the expense of sub-optimal solution of the optimization problem due to the sampling nature of sampling-based approaches. There are many other measures used in information theory and statistics to compare two distributions. Here, we consider Kullback-Leibler (KL) divergence, which measures the relative entropy between two distributions defined over a domain X,

$$D_{KL}(\Gamma||\xi) = \int_X \Gamma_t(\boldsymbol{x}) \log\left(\frac{\Gamma_t(\boldsymbol{x})}{\xi(\boldsymbol{x})}\right) d\boldsymbol{x}. \tag{3.1}$$

KL divergence can encode an ergodic coverage objective without resorting to the spectral decomposition of the desired coverage distribution, whose accuracy is limited by the number of basis functions used. This makes KL divergence faster to compute that the traditional metric for ergodicity given by Eq. (2.2). Additionally, we can explicitly measure the ergodicity of a trajectory without resorting to simplifications in the representation of a robot's sensor footprint.

In our formulation, the optimization problem posed in Eq. (2.10) can be solved by defining the cost function $J(\mathbf{z})$ to be

$$J(\mathbf{z}) = D_{KL}(\Gamma||\xi) \tag{3.2}$$

where z is sampled from a Gaussian mixture model defined in Eq. (2.11).

3.2 Sensor Footprints

In this thesis, we consider two sensor footprints: (1) Gaussian sensor footprint, and (2) forward-looking beam sensor footprint. Assuming there are N robots, let $f_j(.)$ be the sensor footprint of the j-th robot. The time-average statistic, $\Gamma_t(\boldsymbol{x})$, can be defined as,

$$\Gamma_t(\boldsymbol{x}) = \frac{1}{t}\sum_{j=1}^{N}\int_0^t f_j(\boldsymbol{x} - \gamma_j(\tau))d\tau, \tag{3.3}$$

where γ_j is the trajectory of the j-th robot. A Gaussian sensor footprint is described as

$$f_j(\boldsymbol{y}) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \exp\left(-\frac{1}{2}\boldsymbol{y}^T\Sigma^{-1}\boldsymbol{y}\right)$$

where Σ is the covariance of the Gaussian distribution (controls how wide the sensor footprint is). On the other hand, the sensor footprint of a beam model is defined by a radius r, and angle of view β, similar to an ultrasonic sensor as shown in Fig. 3.1. The sensor footprint of the beam model is parameterized as

$$f_j(\boldsymbol{y}) = \begin{cases} 1, & \text{if } |\boldsymbol{y}| \leq r \quad \text{and} \quad \theta_\tau - \frac{\beta}{2} \leq \arctan(\frac{y_2}{y_1}) \leq \theta_\tau + \frac{\beta}{2} \\ 0, & \text{otherwise.} \end{cases}$$

where θ_τ is the heading of the robot at time τ, and y_i is the i-th element of the vector $\boldsymbol{y}$.

3.3 Obstacle Avoidance

Suppose that the domain X contains p obstacles denoted by $O_1, ..., O_p \subset X$. We assume that the robot at state $\boldsymbol{q}$ is occupying a region $A(\boldsymbol{q}) \subset X$. Borrowing the notation in [37], let the function $\textbf{prox}(A_1, A_2)$ return the closest Euclidean distance between two sets $A_{1,2} \subset X$. This function returns a negative value if the two sets intersect. Therefore, for an agent to avoid the obstacles $O_1, ..., O_p$, we impose a constraint of the form of Eq. (2.6) expressed as,

$$F(\boldsymbol{q}(t)) = \min_i \Big(\textbf{prox}\big(A(\boldsymbol{q}(t)), O_i\big) \Big), \ \forall t \in [0, \infty). \tag{3.4}$$

The sampled trajectories that do not satisfy the constraints of Eq. (2.6) are penalized.

3.4 Example: Dubins Car

The sampling-based trajectory planning framework that we use can be applied to any robot whose dynamics are given as Eq. (2.5) given that we can parameterize the space of all trajectories satisfying the dynamics of the robot using motion primitives. We will consider the Dubins car model whose motion is restricted to a plane. The state space for such a model is $Q = SE(2)$ with state $\boldsymbol{q} = (x, y, \theta)$, where (x, y) are the Cartesian coordinates of the robot, and θ is the orientation of the robot in the plane. The dynamics of a dubins car is defined by,

$$\dot{x} = v\cos\theta, \quad \dot{y} = v\sin\theta, \quad \dot{\theta} = w \tag{3.5}$$

where $v \in [v_{min}, v_{max}]$ is a controlled forward velocity, and $w \in [w_{min}, w_{max}]$ is a controlled turning rate. We can represent a trajectory satisfying (3.5) as a set of connected motion primitives consisting of either straight lines with constant velocity v or arcs of radius v/w. We define a primitive by a constant controls (v, w). The duration τ of each primitive is constant and $\tau > 0$. We parameterize the trajectory of the robot using m primitives, and this finite dimensional parameterization is represented by a vector $\boldsymbol{z} \in \mathbb{R}^{2m}$ such that,

$$\boldsymbol{z} = (v_1, w_1, ..., v_m, w_m) \tag{3.6}$$

The j^{th} primitive ends at time $t_j = j\tau$ for $j \in 1, .., m$. For any time $t \in [t_j, t_{j+1}]$, the

parameterization $\varphi_z = (v, w, x, y, \theta)$ is given by,

$$
\begin{aligned}
v(t) &= v_{j+1} \\
w(t) &= w_{j+1} \\
\theta(t) &= \theta_j + w_{j+1}\Delta t_j \\
x(t) &= \begin{cases} x(t_j) + \frac{v_{j+1}}{w_{j+1}}(\sin(\theta(t)) - \sin\theta_j), & \text{if } w_{j+1} \neq 0 \\ x(t_j) + v_{j+1}\Delta t_j \cos\theta_j, & otherwise \end{cases} \\
y(t) &= \begin{cases} y(t_j) + \frac{v_{j+1}}{w_{j+1}}(\cos\theta_j - \cos(\theta(t))), & \text{if } w_{j+1} \neq 0 \\ y(t_j) + v_{j+1}\Delta t_j \sin\theta_j, & otherwise \end{cases}
\end{aligned} \tag{3.7}
$$

where $\theta_j = \theta(t_j)$ and $\Delta t_j = t - t_j$. The paramterization φ_z allows us to calculate the cost function in Eq.(2.9) and then solve tehe optimization problem shown in Eq. (2.10).

3.5 Simulation Results

In this section, we first compare our algorithm with the implementation of Mathew *et al.* [33] in an unconstrained environment. In Section 3.5.2, we demonstrate that our formulation can capture coordination of multiple robots equipped with different sensing capabilities. It is also straightforward to encode additional tasks besides ergodic coverage. In Section 3.5.3, we present an example in which a robot is directed to reach a destination in a desired time while performing ergodic coverage.

3.5.1 Comparisons

We first compare the coverage performance of three different implementations of the ergodic coverage algorithm: (i) We generate a trajectory using the method presented in Mathew *et al.* [33], which will be referred to as spectral-multiscale coverage (SMC) algorithm. (ii) We use STOEC to optimize the trajectory such that Eq. (2.2) is minimized. This implementation will be referred as Ergodic-STOEC. (iii) We use STOEC to optimize the trajectory such that Eq. (3.1) is minimized. This will be referred as KL-STOEC.

We perform numerical simulations for each of the aforementioned implementations. We define the desired coverage distribution $\xi(\boldsymbol{x})$, as a Gaussian mixture model[1] as shown in Fig. 3.2. For visualization, $\xi(\boldsymbol{x})$ is normalized such that the maximum value of the desired coverage distribution takes the value of 1. The initial position of the robot is randomly selected. The robot is assumed to have bounded inputs $v \in [0.1, 5]$ m/s and $w \in [-0.2, 0.2]$ rad/s. For SMC, the total simulation time is $T = 1000$ sec, and the time step is 0.1 sec. For STOEC, we run 20 stages of the algorithm. A single stage of the algorithm corresponds to generating a trajectory using 5 primitives where the duration of each primitive is τ = 10 sec by optimizing Eq. (2.9) . Therefore, the full trajectory is 1000 sec. STOEC is seeded with 40 sampled-trajectories.

The results are shown in Fig. 3.2. Since the metric for ergodicity and KL divergence both measure similarity between two distributions, we need to choose another metric to compare between $\xi(\boldsymbol{x})$ and $\Gamma_t(\boldsymbol{x})$ of the trajectory optimized by KL-STOEC, and to compare

[1] There are no restrictions on the choice of representation for $\xi(\boldsymbol{x})$.

Table 3.1: Comparison of the computation time

Method	Computing 1 stage (50 sec)	Computing the full trajectory (1000 sec)
KL-STOEC	**0.5**	**10.0**
Ergodic-STOEC	1.2	24.0
SMC	1.7	35.0

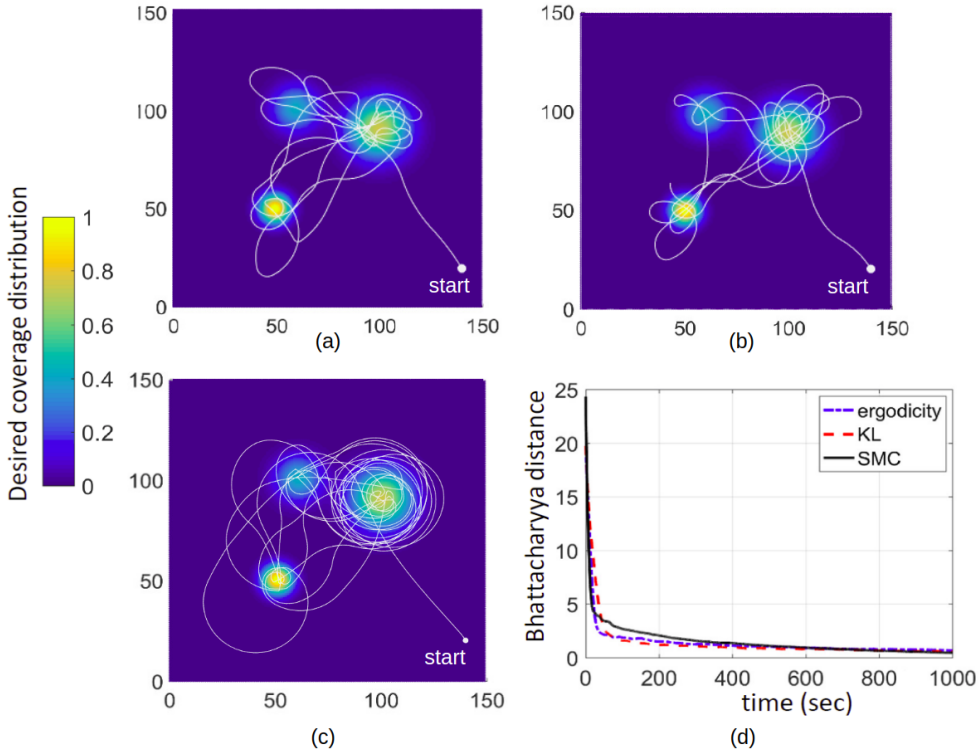

Figure 3.2: Ergodic coverage results for three different implementations: (a) Ergodic-STOEC: minimizing Eq.(2.2), (b) KL-STOEC: minimizing Eq.(3.1), (c) SMC implementation in Mathew *et al.* [33]. Ergodic coverage performance is assessed through plotting the Bhattacharyya distance between the coverage distribution $\xi(\boldsymbol{x})$ and the time-average statistics distribution $\Gamma_t(\boldsymbol{x})$ in (d) for the different implementations. Notice that (a) and (b) have similar decay rates, while the decay rate of (c) is smaller.

betweebetween $\xi(\boldsymbol{x})$ and $\Gamma_t(\boldsymbol{x})$ of the trajectory optimized by Ergodic-STOEC. To assess the coverage performance of the algorithms, we use Bhattacharyya distance which is a well known metric that is used to measures the similarity of two probability distributions. The Bhattacharyya distance is defined as,

$$D_B(\Gamma,\xi) = -\ln\left(\int_X \sqrt{\Gamma_t(\boldsymbol{x})\xi(\boldsymbol{x})}d\boldsymbol{x}\right). \tag{3.8}$$

Table 1 shows the computation time[2] for each algorithm for the example in Fig.3.2. We

[2]The code runs on MATLAB 2016a in Windows 10 on a laptop with i7 CPU and 8 GB RAM.

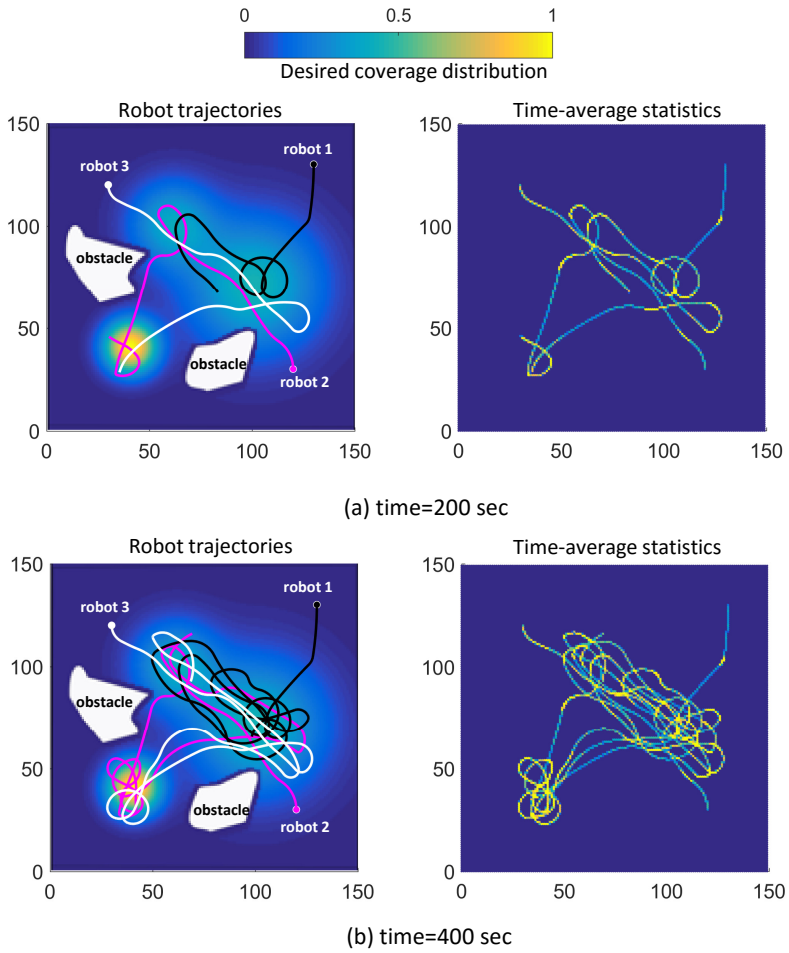

Figure 3.3: Ergodic coverage results for three robots with Dirac delta sensor footprint. The left figures show the robots trajectories, and the right figures show the corresponding time-average statistics. The results are shown for two different instants of the KL-STOEC.

should mention that the computational time of STOEC varies depending on the number of trajectory samples that are seeded to the algorithm and the number of primitives. With around 40 sampled trajectories, the algorithm converges to a solution more quickly than SMC. It is also important to note that STOEC plans over a finite time horizon as opposed to the greedy approach in SMC. The metric for ergodicity, given by Eq. (2.2), gives higher weights to the low frequency components. Therefore, the algorithm favors large scale exploration of the coverage distribution.

3.5.2 Sensor Footprints

In this section, we demonstrate that our framework can capture multi-agent systems with different sensing capabilities. It is important to note that the formulation of the ergodic coverage algorithm assumes that robots have perfect communication, thus have access to the trajectory plans of other robots. The robot trajectories are sequentially optimized by taking into account the time-average statistics of every robot's trajectory. Trajectory optimization also assumes

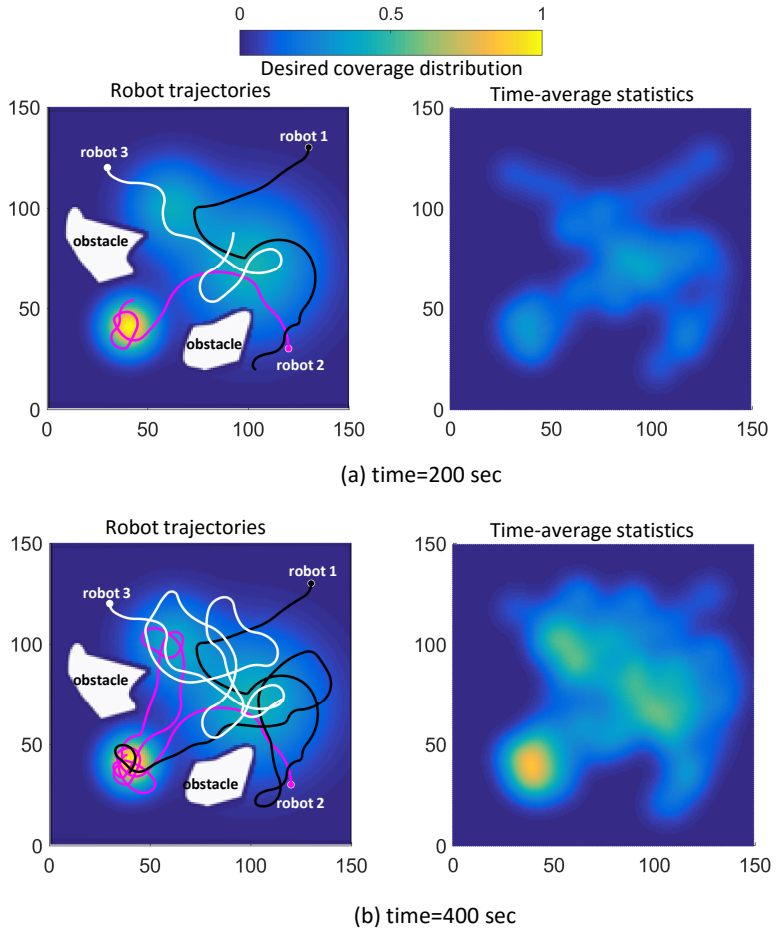

Figure 3.4: Ergodic coverage results for three robots with wide Gaussian sensor footprint. The left figures show the robots trajectories, and the right figures show the corresponding time-average statistics. The results are shown for two different instants of the KL-STOEC.

deterministic motion models.

Fig. 3.3 shows the ergodic coverage of a desired distribution using KL-STOEC assuming a Dirac delta sensor footprint. We run 10 stages of the algorithm using 5 primitives where the duration of each primitive is $\tau = 8$ sec. The robot is assumed to have bounded inputs $v \in [0.1, 5]$ m/s and $w \in [-0.2, 0.2]$ rad/s. Although the coverage focuses on the regions where the measure of the desired coverage distribution, ξ, is greater than 0, the algorithm needs a much longer time for time-average statistics to converge to the desired distribution. Fig. 3.4 demonstrates results with a wide Gaussian sensor footprint using the same parameters. Fig.3.5 shows the errors measured using the Bhattacharyya distance. The computational time for these two scenarios in Fig. 3.3 and Fig. 3.4 are 0.7 sec/stage and 1.7 sec/stage, respectively. Fig. 3.6 shows another example where there are three robots equipped with different sensing capabilities. The sensor footprint of each robot is shown in Fig. 3.6 (c).

Note that, in these examples, the robots can continue to perform ergodic coverage of the domain indefinitely. That is, once the time-average statistics converge to the desired distribution, any new motion will result in a difference between the two distributions, $\Gamma_t(\boldsymbol{x})$ and

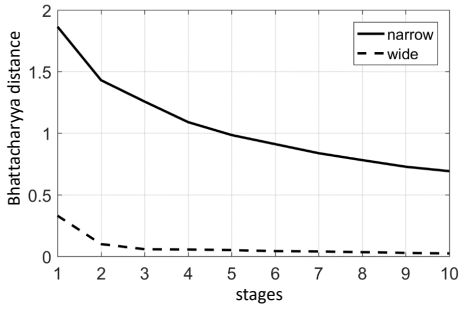

Figure 3.5: Ergodic coverage performance comparison for the robots with narrow vs. wide Gaussian footprints.

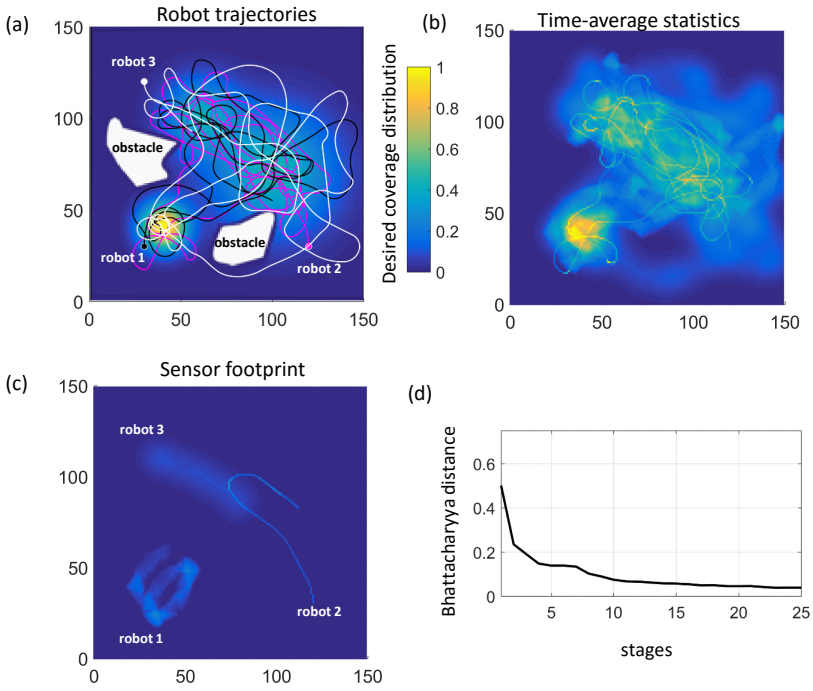

Figure 3.6: Ergodic coverage results for three robots equipped with a forward-looking beam sensor footprint, a Dirac delta sensor footprint, and a wide Gaussian sensor footprint, respectively. The resulting trajectories of the robots using KL-STOEC implementation are shown in (a) for 25 concatenated segments. The time-average statistics is shown in (b). The different footprints of the three robots are shown in (c). Finally, (d) shows the Bhattacharyya distance between the coverage distribution $\xi(\boldsymbol{x})$ and the time-average statistics distribution $\Gamma_t(\boldsymbol{x})$ as a function of the number of stages.

$\xi(\boldsymbol{x})$. The trajectory planner then drives robots to continue covering the domain such that $\Gamma_t(\boldsymbol{x})$ again converges to the desired distribution $\xi(\boldsymbol{x})$. This is a very useful trait for surveillance applications, where the robots need to continuously monitor a region while spending more time in regions that have high values marked by the desired coverage distributions.

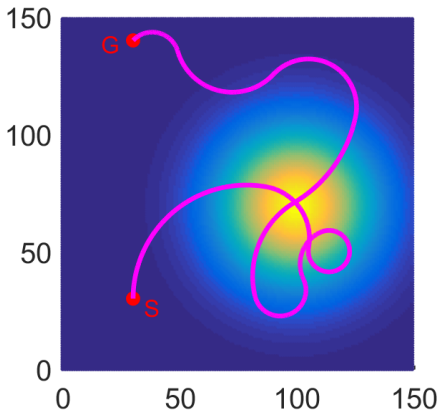

Figure 3.7: The robot starts at **S**:start and is supposed to reach point **G**:goal in 500 seconds using a maximum of 10 primitives.

3.5.3 Point-To-Point Planning

Our framework can be extended to encode additional tasks beside ergodic coverage. We present an example in which the robot reaches a desired state in a given time while performing ergodic coverage along the way. This task is easily performed by adding a term to the objective function, Eq. (3.1), that penalizes the error between the state of the robot at the end of the trajectory (assuming a trajectory of duration T), $\boldsymbol{q}(T)$, and the desired state of the robot $\boldsymbol{q}_d$. Thus, the cost function can be defined as,

$$C(\boldsymbol{z},\boldsymbol{q}) = D_{KL}(\Gamma||\xi) + \alpha\|\boldsymbol{q}(T) - \boldsymbol{q}_d\|_2. \tag{3.9}$$

where $\alpha \geq 0$ is a weighting parameter, chosen heuristically, which specifies how *hard* the constraint of reaching a desired state at time T is. As α increases, the robot is guaranteed to reach its final desired state. We simulate an example of such a scenario in Fig. 3.7. The robot is assumed to have bounded inputs $v \in [0.1, 5]$ m/s and $w \in [-0.2, 0.2]$ rad/s. How to systematically choose the parameter α is an open question and is left as a future work.

3.6 Contribution

The contributions from this chapter include

1. The construction of a new ergodic coverage objective that can take into account the robots' sensor footprint.

2. The formulation of the ergodic coverage within a stochastic trajectory optimization framework that lends itself to motion constraints of the robot, and constraints arising from arbitrary-shaped obstacles.

3.7 Published Work

Material from this chapter has appeared in the following publications

1. Hadi Salman, Elif Ayvali, Howie Choset,"Multi-agent Ergodic Coverage with Obstacle Avoidance", In the proceedings of the Twenty-Seventh International Conference on Automated Planning and Scheduling (ICAPS), Pittsburgh, USA, June 2017.

2. Elif Ayvali, Hadi Salman, and Howie Choset, "Ergodic Coverage In Constrained Environments Using Stochastic Trajectory Optimization", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, Canada, September 2017.

Chapter 4

Trajectory Optimized Active Search for Tumor Localization

Ergodic coverage is one strategy to search for tumors in anatomies [31]. Given a prior distribution of the location of tumors, using a CT scan for example, a robot can be directed to ergodically cover the organ by focusing on areas where the distribution is denser. However, Ayvali *et al.* [31] showed that while ergodic coverage succeeds in locating tumors, other methods, such as Bayesian optimization, perform better in terms of how quickly it locates tumors. Furthermore, the work of Chalasani *et al.* [30] demonstrates how using continuous plapation, such as the sweeping palpation motion used by surgeons, can offer useful information that can be used to simultaneously search for tumors and estimate the geometry of the organ. Yet, this work does not consider any robot constraints, and it assumes that the robot is capable of execting those continuous plapation trajectories.

This directed us to explore GP-based active search methods and reformulate them in a framework of continuous palpation that takes into account the motion constraints of the robot, in addition to any undesired regions which should not be traversed during palpation.

In this chapter, we build on top of the work of Ayvali *et al.* [31], and we explore more recent active search methods. We then integrate these active search methods with a sampling-based trajectory planner that takes into account constraints associated with the motion model of the robot. Finally, we demonstrate that obstacle avoidance can be easily incorporated into this framework by penalizing the sampled trajectories that collide with arbitrarily-shaped obstacles or that pass through restricted regions.

4.1 Modelling the Stiffness Map

In our work, we utilize Gaussian processes (GPs) to model the distribution of stiffness of the organ. By using GP, we assume a smooth change in the latent function we are estimating which is the stiffness distribution across the organ. We also assume that every point on the organs's surface can be mapped in a 2D grid, thus, the domain of search used is $X \subset \mathbb{R}^2$. The measured force and deflection after probing the organ by the robot at position $\boldsymbol{x}$ provides the stiffness estimation represented by y. The set $\{(\boldsymbol{x_1}, y_1),\ldots,(\boldsymbol{x}_N, y_N)\}$ contains the data points that are used to train the GP as discussed in Section 2.3, where N is the number of probed points.

The motivation behind choosing a GP to model the stiffness map of the organ is that a

GP estimates the mean and covariance of the stiffness function at each point in the search space. This covariance represents the uncertainty of the value estimated by the GP at each point. Therefore, we obtain an uncertainty bound for the stiffness estimates across the organ which can be used to direct the search as we will discuss in Section 4.2.

Before moving on, we present a comparison between GP and a variety of regression methods. This comparison shows that, in addition to offering a confidence bound for the stiffness map, a GP leads to lower RMSE of estimation compared to other regression methods when estimating a latent function in a low dimensional space $\left(\mathbb{R}^2\right)$ and with few training data ($(\boldsymbol{x}, y)$ tuples of probed points on the organ). We consider the following state-of-the-art regression methods,

- Gaussian Process regression (GPR)
- Kernel ridge regression
- Support vector regression (SVR)
- k-nearest neighbors (k-NN)
- Decision tree
- Ensemble method (Adaboost)
- Neural network
- Random forest

In addition to these methods, we consider cubic interpolation as a method to estimate the stiffness map given samples from the true function. We randomly sample 50 datapoints $\{(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x}_{50}, y_{50})\}$ from a ground truth stiffness map. We use the same 50 datapoints to train each of the aforementioned regression methods. The results are shown in Fig. 4.1 and 4.2. The GP outperforms other regression methods in estimating the ground truth stiffness map.

Note that we choose only 50 points to train each method because in applications such as tumor localization, it is crucial to be able to detect tumors in the least number of probes possible. Also, we note that as the number of data points increases, all the regression methods tend to give similar estimates. Furthermore, for other applications that are characterized with high dimensional space or huge amounts of data, GP might not be the best option, and neural networks, for example, can be more suitable.

4.2 GP-based Active Search

In Section 4.1, we presented how we model and estimate the stiffness map assuming we are given a set labelled data points $\{(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x}_N, y_N)\}$. In this section, we discuss how we can collect these data points (where to probe and collect stiffness values) in an efficient sequential fashion.

We assume that a robot is equipped with a force sensor attached to its end-effector, and the robot is supposed to move, probe, and measure stiffness at a certain location of the organ. In this work, we leverage state-of-the-art active search methods to guide the robot where to go and probe next includeing active area search (AAS), level sets estimation (LSE), and uncertainty sampling (UNC), in addition to Bayesian optimization algorithm. These algorithms are described in Section 2.4.

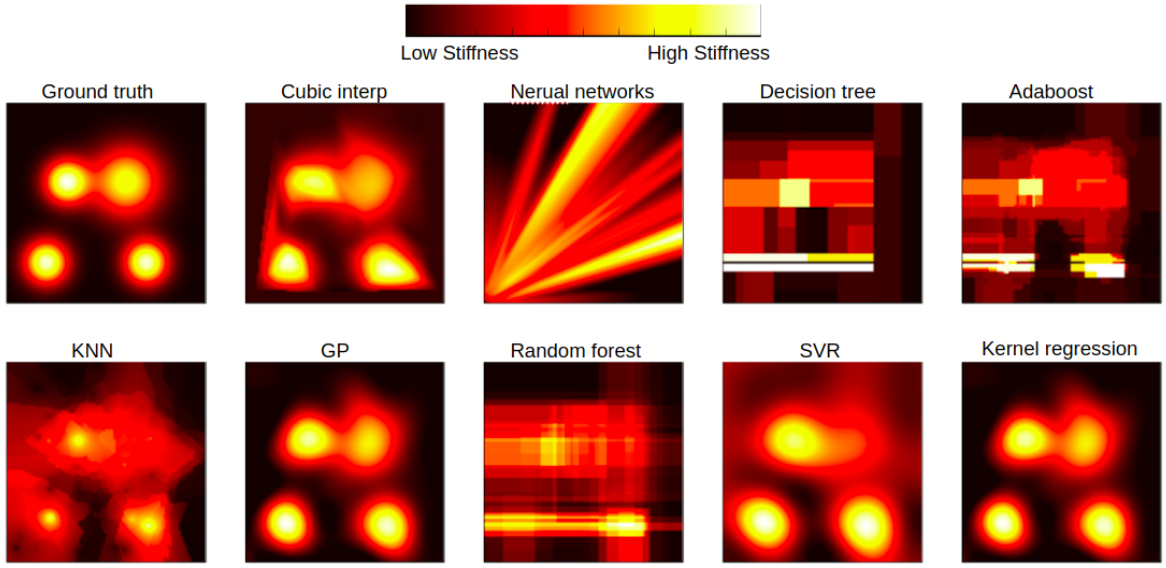

Figure 4.1: The estimated stiffness map using various regression methods. Each method is trained using 50 randomly sampled points from the ground truth stiffness map shown in the upper left corner. Given the few data points that are used to fit a function, GP qualitatively outperforms the other methods. For quantitative analysis check Fig. 4.2.

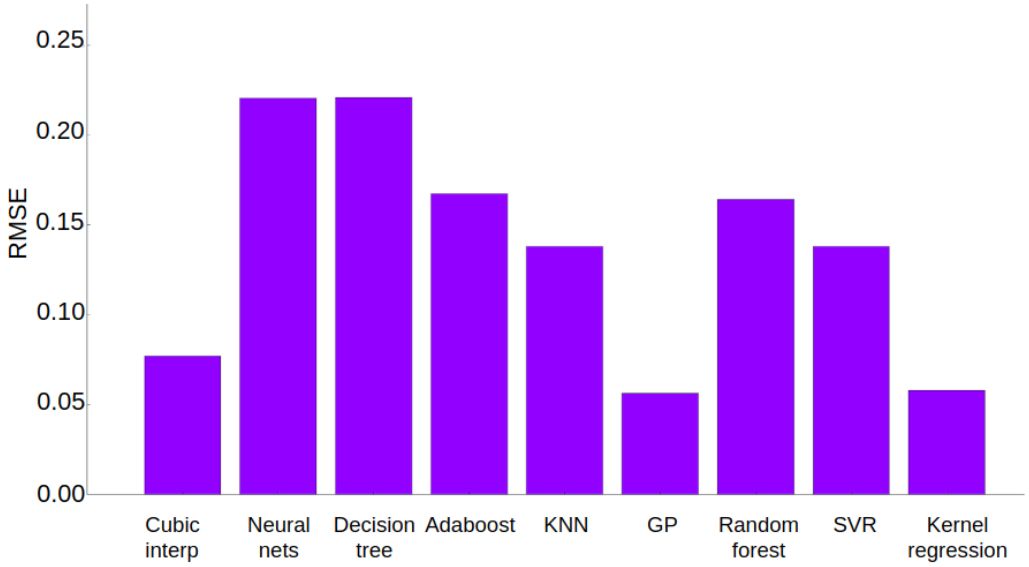

Figure 4.2: The RMSE error between the ground truth stiffness map and the estimated stiffness map using various regression methods. We use 50 randomly sampled data from the ground truth stiffness map to train each of the methods. GP results in the closest estimated stiffness map to the ground truth.

4.3 Objective Function for Active Search

In this section, we introduce the objective function for which we optimize. We use the sampling-based trajectory planning framework presented in Section 2.2.

Algorithm 1 Discrete Palpation

Initialize the GP with zero mean and squared exponential covariance function
$\boldsymbol{x}^* \leftarrow random$ ▷ random initialization of probed point
while TRUE **do**
    Probe at $\boldsymbol{x}^*$
    Calculate stiffness at probed points
    Update GP using the stiffness estimate
    Update acquisition function ξ_{acq} using GP
    $\boldsymbol{x}^* \leftarrow \arg\max \xi_{acq}$

Algorithm 2 Trajectory-Optimized Continuous Palpation

Initialize the GP with zero mean and squared exponential covariance function
$\boldsymbol{z}^* \leftarrow random$ ▷ random initial trajectory in the space of motion primitives
while TRUE **do**
    Execute trajectory $\boldsymbol{z}^*$
    Collect stiffness measurements along the trajectory
    Update GP using the stiffness estimate
    Update acquisition function ξ_{total} using GP ▷ (4.2)
    $\boldsymbol{z}^* \leftarrow \arg\min_{\boldsymbol{z}\in Z_{con}} J(\boldsymbol{z})$ ▷ (4.1)

The problem of finding the location and shape of the stiff inclusions (objects of interest in an organ, such as tumors) while considering various inherent constraints can be modeled as an optimization problem. However, an exact functional form for such an optimization is not available in reality. Hence, we maintain a probabilistic belief about the stiffness distribution and define an "acquisition function" to determine where to sample next.

We define the cost function $J(\boldsymbol{z})$ that we use to optimize the plapation trajectory of the robot as,

$$J(\boldsymbol{z}) = -\int_{\phi(\boldsymbol{z})} \xi_{total}(\boldsymbol{q}) d\boldsymbol{q} \tag{4.1}$$

where $\boldsymbol{z}$ is sampled from a Gaussian mixture model defined in Eq. (2.11), $\phi(\boldsymbol{z})$ is the sampled trajectory that is parameterized by the motion primitive $\boldsymbol{z}$, and ξ_{total} is total acquisition that is to be maximized by each sample trajectory, and is defined by,

$$\xi_{total}(\boldsymbol{q}) = \eta\left(\xi_{acq}(\boldsymbol{q}) + \alpha(t)\xi_{prior}(\boldsymbol{q})\right) \tag{4.2}$$

where ξ_{acq} is a normalized acquisition function defined by any one of the active learning algorithms described in Section 2.4. ξ_{acq} is defined as the expected total reward from Eq. 2.12 when using AAS, the ambiguity a_t from Eq. 2.13 in the case of LSE, the uncertainty in the GP posterior distribution when using UNC[1] , and the EI as shown in Eq. 2.14 in the case of BOA. $\xi_{prior}(\boldsymbol{q})$ is a normalized distribution capturing the prior on the locations of the tumors, and it decays as search progresses by means of a decay function $\alpha(t)$. Note that the effect of this term has been studied in detail in our previous work [48]. In this work, we only focus on the effect of ξ_{acq}. Finally, η is a normalizing constant.

[1]This uncertainty is associated with the estimated stiffness map produced by the GP and should not to be confused by the uncertainty in the robot's position or force measurement.

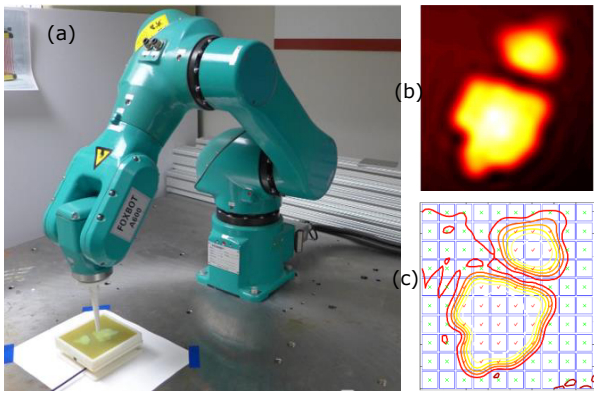

Figure 4.3: (a) 6 DoF industrial robot arm with a force sensor attached to its end effector. (b) Ground truth stiffness map generated by densely probing a silicone phantom organ. (c) A contour map showing various stiffness levels.

After defining the objective function for which we would like to optimize, we summarize the active search for tumor process by the pseudocode shown in Algorithm 2.

4.4 Obstacle Avoidance

In some surgical scenarios, one may want to avoid palpating certain regions of the organ's surface such as a bony region or regions occupied by other instruments etc. In order to handle such scenarios, our framework can also account for obstacles while searching following the formulation presented in Section 3.3.

4.5 Simulation Results

In this section, we present and compare simulation results for discrete probing and continuous probing scenarios using the various active search algorithms presented in Section 2.4.

4.5.1 Discrete Probing

We start by comparing four different sequential probing algorithms which we adopt in this thesis as efficient methods to guide our tumor search. This section considers discrete probing scenarios, that are described in Algorithm 1, where it is assumed that the robot can reach any point in the search domain.

The robot has no prior knowledge of the locations of the stiff regions. The starting point of the robot is chosen randomly in the 2D seach domain. The robot then sequentially decides where to go and probe next such that the acquisition function ξ_{total} associated with each algorithm is maximized. For example, for the LSE algorithm, the robot chooses the point with the highest ambiguity in its classification at each step and goes and probes there. We test the four different algorithms listed in Section 2.4 in a simulated experiment. We use a ground truth of a silicone phantom organ obtained by doing a raster scan using a 6 DoF

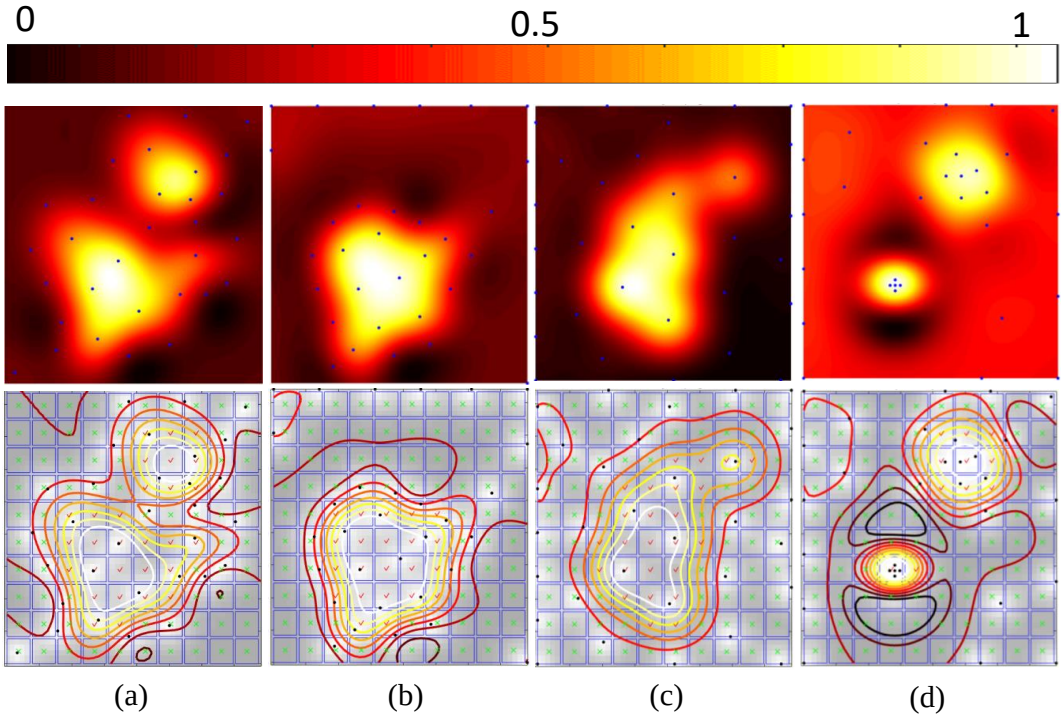

Figure 4.4: Discrete probing using (a) AAS, (b) LSE, (c) UNC, and (d) EI algorithms with only 30 probed points. We discretize the search space into regions (squares) as shown in the figures at the bottom: regions whose average estimated stiffness is above a certain threshold are marked as with tumor and are marked with a red tick. Otherwise, the regions are marked with green cross signifying normal tissue regions.

industrial robot as shown in Fig. 4.3. The results of this simulation are shown in Fig. 4.4. We will discuss the results of this experiment in Section 4.5.3[2].

To assess the performance of the different algorithms, we report the *recall* since it is a suitable performance measure for regions-of-interest detection problems. The recall is widely used in the machine learning community as a performance measure for similar problems [13]. In order to calculate the recall, we discretize the search space into regions (squares) as shown in Fig. 4.4; regions whose average estimated stiffness is above a certain threshold are considered as containing a tumor and are marked with a red tick. Otherwise, the regions are marked with green cross signifying normal tissue regions. The recall is then calculated as the ratio of the number of regions that are marked with red ticks (contain tumor) over the total number of regions that truely contain tumor.

We perform 100 simulations with same parameters of the GP but with randomly generated ground truths. Then we perform another 100 simulations on a fixed ground truth but with random initial probed points. The average of the recall across each 100 simulations as a function of the number of probed points for the different algorithms is reported in Fig. 4.5.

[2]Note that through out this thesis, we normalize the estimated stiffness map. This is because we are more interested in revealing the position and boundaries of a tumor rather than its actual stiffness.

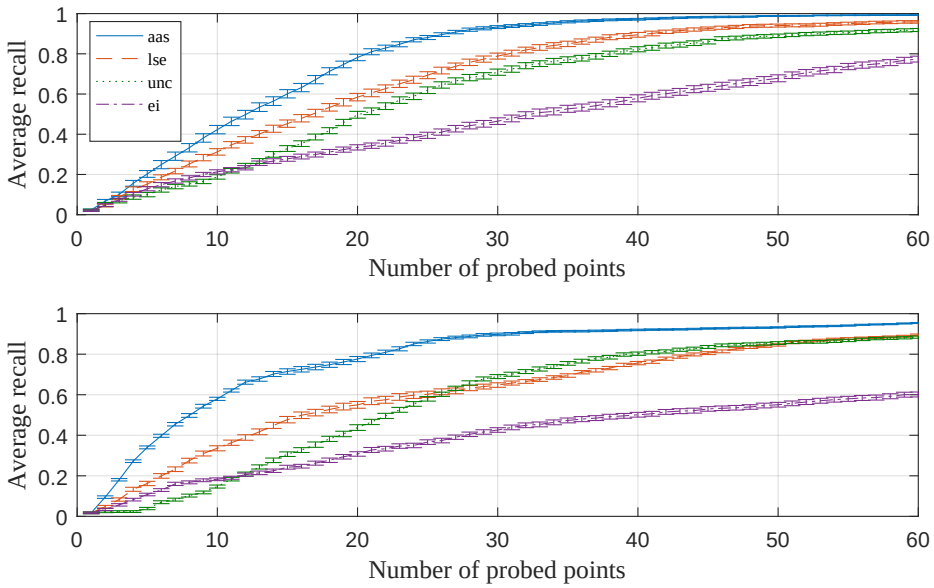

Figure 4.5: The top plot shows the average of the recall as a function of the number of the palpated points across 100 simulations of discrete palpation (Algorithm 1) with randomly generated ground truths for the stiffness map. The mean and the covariance of each of the four algorithms considered in this thesis is reported. The bottom plot shows the average of the recall for 100 repeated simulations for tumor localization over a fixed ground truth but with random starting probing point locations.

4.5.2 Continuous Probing

Discrete probing does not impose a constraint on the next location to be probed. A robot may not be able to reach the next desired point due to motion constraints. Further, the robot can benefit from collecting information along an optimized path to improve the predictions of tumor location and boundaries.

We perform continuous palpation experiments in simulation on the same dataset used in the previous section and shown in Fig. 4.3. The results are shown in Fig. 4.6 and discussed in Section 4.5.3. Similar to Section 4.5.1, we repeat this simulation 100 times with same parameters of the GP but with randomly generated ground truths. Then we repeat 100 simulations on a fixed ground truth but with random initial starting positions. The average of the recall across each 100 simulations as a function of the number of fixed-frequency-sampled measurements along the palpation path for the different algorithms is reported in Fig. 4.7.

4.5.3 Discussion

The Bayesian optimization algorithm is designed to focus on finding the global maxima of a function. Therefore, once a point with high stiffness is detected, the algorithm collects more samples around it instead of moving out from that region and discerning the boundary. This is clearly observed in both Fig 4.4-d as well as Fig. 4.6-d.

LSE algorithm is designed to direct search around an predefined level set[3] corresponding to tumor boundaries and, as a result, can find the boundaries of the tumors fairly well. How-

[3]This level set can defined as a percentage of the maximum estimated stiffness value so far.

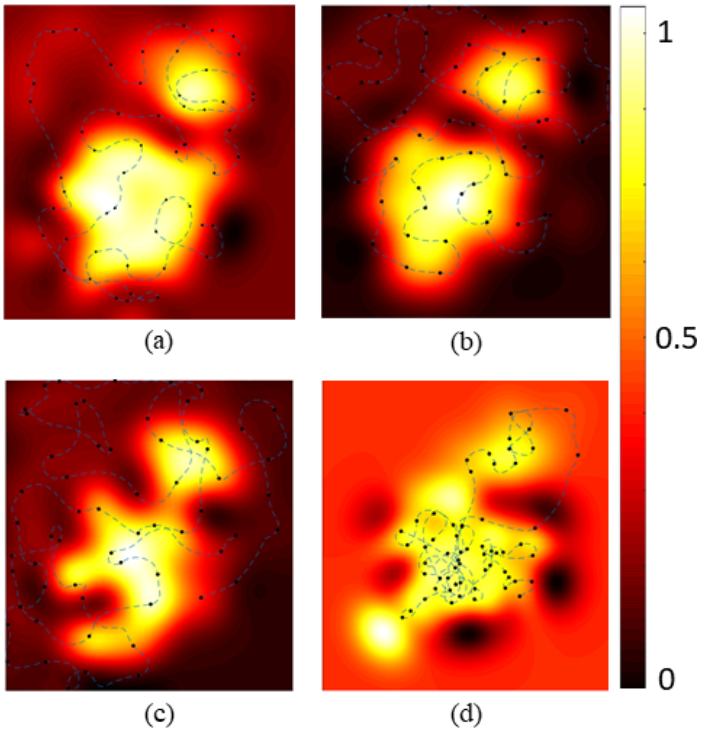

Figure 4.6: Continuous palpation for tumor localization using (a) AAS, (b) LSE, (c) UNC, and (d) EI acquisition functions in a trajectory optimized search framework represented by Algorithm 2.

ever, when each tumor boundary lies on a different level set, the algorithm may spend too much time finding one boundary instead of exploring for other tumors. This is evident from Fig. 4.4(b), where the shape of one tumor is estimated correctly, but in the given number of probings, the second tumor was not detected.

AAS provides a good balance between finding the boundaries of the tumors and finding the location of multiple tumors as the algorithm searches all areas where the average of the unknown function (stiffness distribution) over the region exceeds the implicit threshold. Both in the case of discrete as well as continuous palpation, the AAS outperforms all the other approaches. The UNC approach has the worst performance since the algorithm is blind to the value of the predicted stiffness distribution.

Table 1 shows the computation time[4] for each algorithm required to generate a 30-sec robot-trajectory. LSE, UNC, and EI have comparable average computation times of around 0.5 sec, whereas AAS has an average computation time of around 0.85 sec. Notice that AAS sacrifies some computation speed for more accurate results. Yet the computation time of the four algorithms is sufficient for real-time applicability.

Fig. 6 shows simulation results for a case where there are restricted regions in the domain that should be avoided. The trajectory planner, using the AAS algorithm, succeeds in avoiding the obstacle while still localizing both tumors.

Note that the recent work of Chalasani *et al.* [32] uses continuous palpation to estimate the stiffness and the geometry of an organ. The palpatation is done along predefined trajectories or under telemanipulation. Thus, we expect their method to perform poorly compared to ours

[4]The code runs on MATLAB 2016a in Windows 10 on a laptop with i7 CPU and 8 GB RAM.

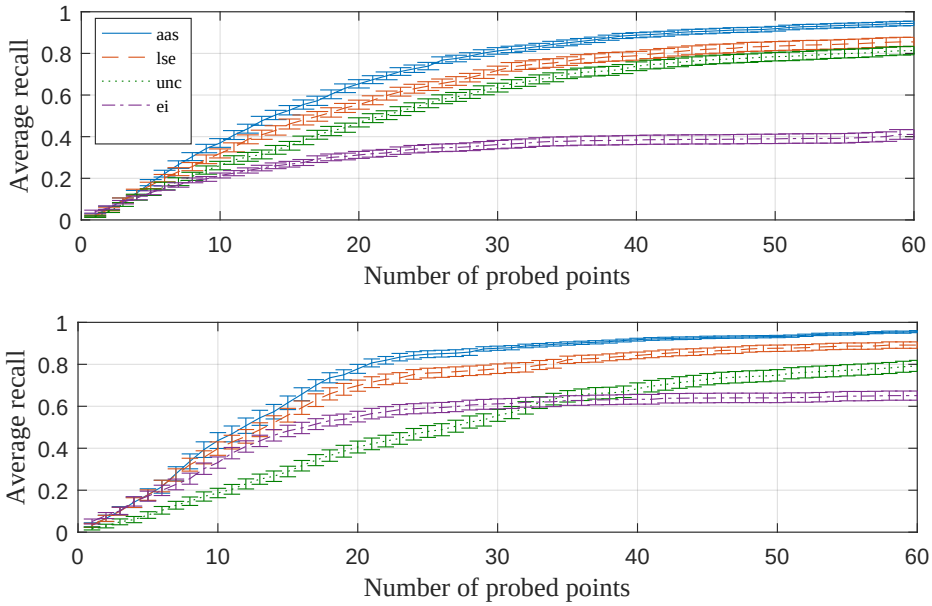

Figure 4.7: The top plot shows the average recall as a function of the number of the palpated points repeated over 100 simulations of continuous palpation (Algorithm 2) with randomly generated ground truths for the stiffness map. The mean and the covariance of each of the four algorithms considered in this thesis is reported. The bottom plot shows the average recall for 100 repeated simulations for tumor localization over a fixed ground truth but with random starting probing point for the algorithms.

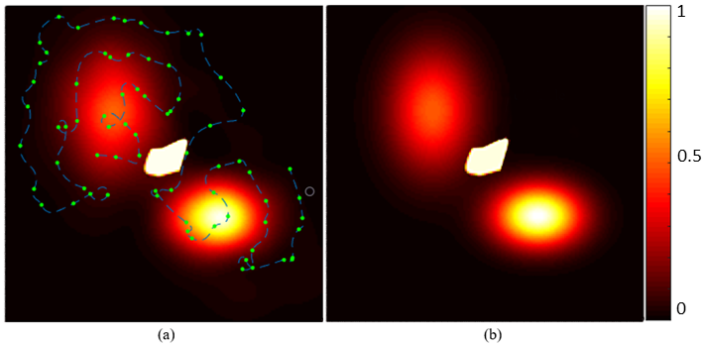

Figure 4.8: (a) The trajectory of the robot overlaid on the predicted stiffness function of the search domain. (b) The ground truth of the stiffness function. The white region is an obstacle. The green points along the trajectory are the points which we used to update the GP (probed points). The AAS algorithm is used.

as their method do not encode an active search objective as we do to reduce palpation time. Furthermore, their method do not take into account constraints resulting from obstacles in the search domain, or from the kinematic model of the robot.

Table 4.1: Comparison between the algorithms for computing a 30 sec robot trajectory

Method	AAS	LSE	UNC	EI
Mean computation time (sec)	0.85	0.55	0.49	0.52

4.6 Robot Experiments

We validate our results by performing experiments on three different robots (6 DoF industrial robot, dVRK, and IREP) to do autonomous palpation and search for tumors. The experimental studies are all performed on phantom silicone organs with embedded stiff inclusions. We observe that continuous palpation using AAS produces best results as previously observed in simulation. Therefore, we only use the AAS algorithm in all the robot experiments.

4.6.1 6-DoF Industrial Arm

We use a 6-DoF industrial arm as a platform to verify our simulation results and show that our framework runs real-time (See Fig. 4.3). A commercial force sensor, ATI Nano25 F/T, was attached at the end effector of the robot. As the robot is commanded to move along a trajectory, we continuously collect force measurements from the sensor and position measurements from the kinematics. We employ a linear stiffness model and use the slope of the line that best fits the force-displacement profile similar to [27] to find a scalar stiffness value at every location on the organ.

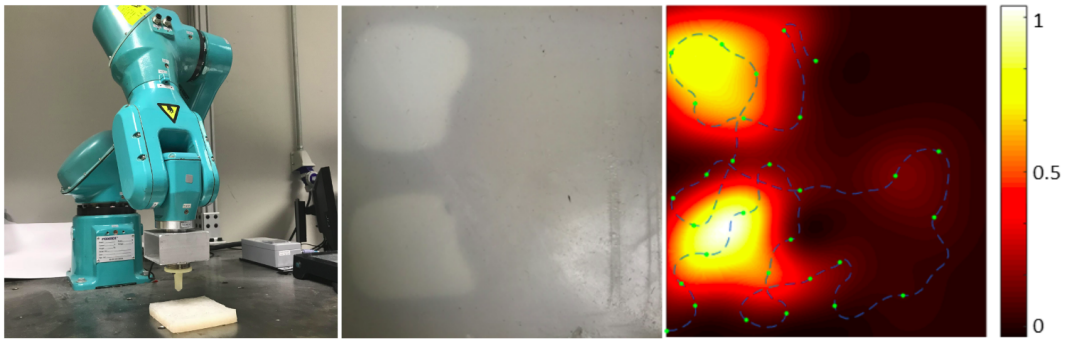

Figure 4.9: Result of the experiments performed using the 6-DoF Industrial Arm. Left: Top view of the silicon phantom organ showing two stiff inclusions. Right: Stiffness map as estimated by our approach. The palpation trajectory is superimposed on the stiffness map.

Fig. 4.9 shows the stiffness map as estimated by using our framework to palpate the organ along a continuous trajectory. The estimated stiffness map clearly reveals the location and the shape of the stiff inclusions without wasting time exploring the softer regions of the organ.

4.6.2 da Vinci Research Kit

We use the open source da Vinci Research Kit (dVRK) [49] for evaluating our approach on silicone tissue samples. The dVRK serves as a realistic surgical platform for evaluating the efficacy of tumor search algorithms. In order to perform palpation, we attach a custom 3D printed spherical-head tip to the 8mm needle driver tool of the robot. The silicone tissue

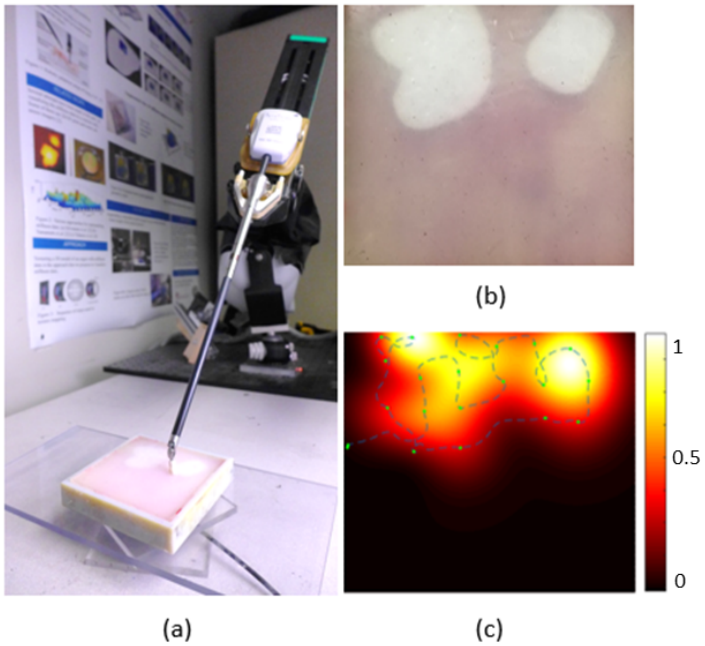

Figure 4.10: (a) Experimental setup showing da Vinci Research Kit (dVRK), equipped with a spherical tool tip. (b) Silicone phantom organ with embedded stiff inclusions. (c) Stiffness map as estimated by our approach using active area search and continuous palpation. The estimated stiffness map accurately reveals the location and shape of the two embedded stiff inclusions.

sample with embedded stiff inclusions (see Fig. 4.10) is placed on top of an ATI Nano25 F/T sensor. Fig. 4.10 shows the stiffness map as estimated by our approach as well as the superimposed palpation trajectory. The stiffness map accurately reveals the stiff inclusions without wasting time exploring the softer regions in the bottom half of the tissue sample.

4.6.3 Insertable Robotic Effector Platform (IREP)

The IREP is a two-segment, four-backbone continuum robot actuated with push-pull nitinol wires designed for single port access surgery [50]. The IREP has an architecture which is very different from conventional rigid link robots. Hence, IREP provides a challenging platform to demonstrate our approach which can take into account the kinematic constraints of the robot. The experimental set up is similar to the one used with dVRK and is shown in Fig. 4.11. While this type of robot architecture is compatible with intrinsic force sensing as in [7], the integration of trajectory optimization with intrinsic force sensing on the IREP and the accompanying challenges of uncertainty estimation of pose and force are part of ongoing research.

Fig. 4.11(b) shows the ground truth stiffness map as generated by densely probing the organ surface using the IREP over a grid of 330 points. Fig. 4.11(c) shows the stiffness map as estimated by our approach as well as the superimposed palpation trajectory. In this experiment, we do not perform continuous palpation with the robot, but instead use the data obtained by densely probing with IREP to simulate continuous palpation. The estimated stiffness map qualitatively confirms with the ground truth stiffness map.

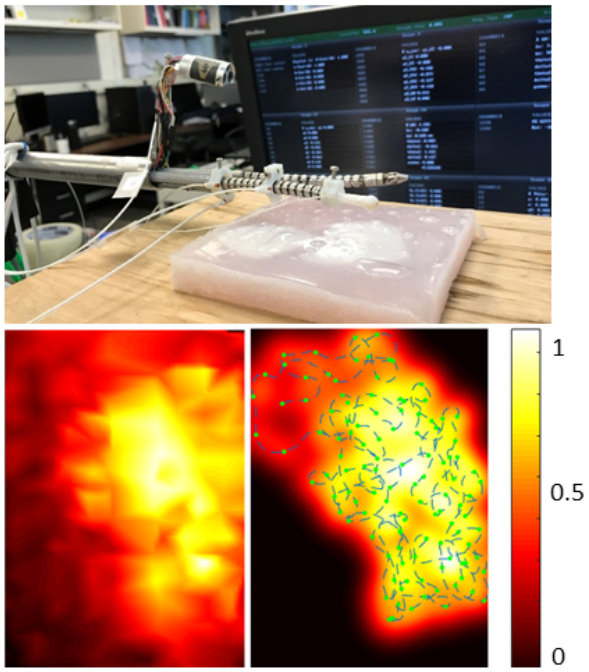

Figure 4.11: (a) Experimental setup consisting of an Insertable Robotic Effector Platform (IREP), probing a silicone phantom organ which is placed on top of a force sensor. (b) Ground truth stiffness map generated by densely probing the organ. (c) Stiffness map as estimated by continuous palpation using AAS. The estimated stiffness map confirms well with the ground truth.

4.7 Contribution

The contribution from this chapter is the formulation of a tumor search framework that leverages state-of-the-art active search methods as the objective to optimize robot's trajectories and explicitly encodes search of stiff regions and their boundaries.

4.8 Published Work

Material from this chapter has appeared in the following publications

1. Nicolas Zevallos, Rangaprasad Arun Srivatsan, Hadi Salman, Lu Li, Jianing Qian, Saumya Saxena, Mengyun Xu, Kartik Patath and Howie Choset, "A Surgical System for Automatic Registration, Stiffness Mapping and Dynamic Image Overlay", Robotics: Science and Systems (RSS), Pittsburgh, PA, USA, June 2018.

2. Hadi Salman, Elif Ayvali, Rangaprasad Arun Srivatsan, Yifei Ma, Nico Zevallos, Rashid Yasin, Long Wang, Nabil Simaan, and Howie Choset, "Trajectory-Optimized Sensing for Active Search of Tissue Abnormalities in Robotic Surgery", IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, May 2018.

3. Nicolas Zevallos, Rangaprasad Arun Srivatsan, Hadi Salman, Lu Li, Jianing Qian, Saumya Saxena, Mengyun Xu, Kartik Patath and Howie Choset, "A Surgical System for Automatic Registration, Stiffness Mapping and Dynamic Image Overlay", International Symposium on Medical Robotics (ISMR), Atlanta, USA, March 2018.

Chapter 5

Conclusion and Future Work

This thesis extended the ergodic coverage algorithm presented in [4] to robots operating in constrained environments. The sampling-based trajectory planner presented in this work allows obstacle avoidance by penalizing sampled trajectories that collide with arbitrarily-shaped obstacles or that cross restricted regions. We demonstrated that Kullback-Leibler divergence can also be used to encode an ergodic coverage objective without resorting to spectral decomposition of the desired coverage distribution, whose accuracy is limited by the number of basis functions used. We believe, our formulation will be of interest to the wider robotics community because it captures sensor footprints, avoids obstacles, and can be applied to nonlinear dynamic systems. Our framework has some limitations related to its centralized nature. Therefore, any communication failure or limited range communication between the agents can affect the performance of our framework. Future work will focus on decentralization of the ergodic coverage algorithm to address this issue.

This thesis also introduced an approach for active search of stiff inclusions, such as tumors and arteries, in tissues by means of forceful palpation. We incorporated three different active learning objectives, namely active area search, level sets estimation and uncertainty sampling, into a sampling-based trajectory planning framework that respects the robot's kinematic and workspace constraints. One limitation of our framework is that it assumes that the user has a prior knowledge of which kernel to use for the GP based on the shape of the stiff inclusions that the robot is searching for. Automatic choice of the kernel is left for future work. Accurate modelling of the interaction between the tissue and and continuum robots, and accurate force sensing remains to be significant challenges in this domain and will be a focus of future work as well.

Bibliography

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[2] E. Acar, H. Choset, Y. Zhang, and M. Schervish, "Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods," *The Int. Journal of Robotics Research*, vol. 22, no. 7 - 8, pp. 441 – 466, July 2003.

[3] F. Bourgault, T. Furukawa, and H. F. Durrant-Whyte, "Optimal search for a lost target in a Bayesian world," in *Field and Service Robotics*. Springer, 2003, pp. 209–222.

[4] G. Mathew and I. Mezić, "Metrics for ergodicity and design of ergodic dynamics for multi-agent systems," *Physica D: Nonlinear Phenomena*, vol. 240, no. 4, pp. 432–442, 2011.

[5] L. M. Miller and T. D. Murphey, "Trajectory optimization for continuous ergodic exploration," in *American Control Conf. (ACC), 2013*. IEEE, 2013, pp. 4196–4201.

[6] P. Puangmali, K. Althoefer, L. D. Seneviratne, D. Murphy, and P. Dasgupta, "State-of-the-art in force and tactile sensing for minimally invasive surgery," *IEEE Sensors Journal*, vol. 8, pp. 371–381, 2008.

[7] K. Xu and N. Simaan, "An investigation of the intrinsic force sensing capabilities of continuum robots," *IEEE Transactions on Robotics*, vol. 24, no. 3, pp. 576–587, 2008.

[8] A. L. Trejos, J. Jayender, M. Perri, M. D. Naish, R. V. Patel, and R. Malthaner, "Robot-assisted tactile sensing for minimally invasive tumor localization," *The International Journal of Robotics Research*, vol. 28, no. 9, pp. 1118–1133, 2009.

[9] J. B. Gafford, S. B. Kesner, A. Degirmenci, R. J. Wood, R. D. Howe, and C. J. Walsh, "A monolithic approach to fabricating low-cost, millimeter-scale multi-axis force sensors for minimally-invasive surgery," in *ICRA*. IEEE, 2014, pp. 1419–1425.

[10] S. McKinley, A. Garg, S. Sen, R. Kapadia, A. Murali, K. Nichols, S. Lim, S. Patil, P. Abbeel, A. M. Okamura, *et al.*, "A single-use haptic palpation probe for locating subcutaneous blood vessels in robot-assisted minimally invasive surgery," in *IEEE International Conference on Automation Science and Engineering*, 2015, pp. 1151–1158.

[11] L. Li, B. Yu, C. Yang, P. Vagdargi, R. A. Srivatsan, and H. Choset, "Development of an inexpensive triaxial force sensor for minimally invasive surgery," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017.

[12] A. Garg, S. Sen, R. Kapadia, Y. Jen, S. McKinley, L. Miller, and K. Goldberg, "Tumor localization using automated palpation with gaussian process adaptive sampling," in *Automation Science and Engineering (CASE), 2016 IEEE International Conference on*. IEEE, 2016, pp. 194–200.

[13] Y. Ma, R. Garnett, and J. Schneider, "Active area search via Bayesian quadrature," in *Artificial Intelligence and Statistics*, 2014, pp. 595–603.

[14] R. Garnett, Y. Krishnamurthy, X. Xiong, J. Schneider, and R. Mann, "Bayesian optimal active search and surveying," *arXiv preprint arXiv:1206.6406*, 2012.

[15] K. E. Petersen, *Ergodic theory*. Cambridge University Press, 1989, vol. 2.

[16] H. Choset, "Coverage for robotics–a survey of recent results," *Annals of mathematics and artificial intelligence*, vol. 31, no. 1-4, pp. 113–126, 2001.

[17] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull, "Morse decompositions for coverage tasks," *The International Journal of Robotics Research*, vol. 21, no. 4, pp. 331–344, 2002.

[18] P. N. Atkar, A. Greenfield, D. C. Conner, H. Choset, and A. A. Rizzi, "Uniform coverage of automotive surface patches," *The International Journal of Robotics Research*, vol. 24, no. 11, pp. 883–898, 2005.

[19] A. Hubenko, V. A. Fonoberov, G. Mathew, and I. Mezic, "Multiscale adaptive search," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 4, pp. 1076–1087, 2011.

[20] H. Salman, E. Ayvali, and H. Choset, "Multi-agent ergodic coverage with obstacle avoidancech," in *International Conference on Automated Planning and Scheduling (ICAPS), Accepted*, 2017.

[21] F. H. Clarke, Y. S. Ledyaev, R. J. Stern, and P. R. Wolenski, "Nonsmooth analysis and control theory". Springer Science & Business Media, 2008, vol. 178.

[22] T. Yamamoto, B. Vagvolgyi, K. Balaji, L. L. Whitcomb, and A. M. Okamura, "Tissue property estimation and graphical display for teleoperated robot-assisted surgery," in *ICRA*, 2009, pp. 4239–4245.

[23] M. Beccani, C. Di Natali, L. J. Sliker, J. A. Schoen, M. E. Rentschler, and P. Valdastri, "Wireless tissue palpation for intraoperative detection of lumps in the soft tissue," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 2, pp. 353–361, 2014.

[24] H. Liu, D. P. Noonan, B. J. Challacombe, P. Dasgupta, L. D. Seneviratne, and K. Althoefer, "Rolling mechanical imaging for tissue abnormality localization during minimally invasive surgery," *IEEE Transactions on Biomedical Engineering*, vol. 57, pp. 404–414, 2010.

[25] R. E. Goldman, A. Bajo, and N. Simaan, "Algorithms for autonomous exploration and estimation in compliant environments," *Robotica*, vol. 31, no. 1, pp. 71–87, 2013.

[26] R. D. Howe, W. J. Peine, D. Kantarinis, and J. S. Son, "Remote palpation technology," *IEEE Engineering in Medicine and Biology Magazine*, vol. 14, no. 3, pp. 318–323, 1995.

[27] R. A. Srivatsan, E. Ayvali, L. Wang, R. Roy, N. Simaan, and H. Choset, "Complementary model update: A method for simultaneous registration and stiffness mapping in flexible environments," in *ICRA*. IEEE, 2016, pp. 924–930.

[28] K. A. Nichols and A. M. Okamura, "Methods to segment hard inclusions in soft tissue during autonomous robotic palpation," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 344–354, 2015.

[29] E. Ayvali, R. A. Srivatsan, L. Wang, R. Roy, N. Simaan, and H. Choset, "Using Bayesian optimization to guide probing of a flexible environment for simultaneous registration and stiffness mapping," in *ICRA*, no. 10.1109/ICRA.2016.7487225, 2016, pp. 931–936.

[30] P. Chalasani, L. Wang, R. Roy, N. Simaan, R. H. Taylor, and M. Kobilarov, "Concurrent nonparametric estimation of organ geometry and tissue stiffness using continuous adaptive palpation," in *ICRA*. IEEE, 2016, pp. 4164–4171.

[31] E. Ayvali, A. Ansari, L. Wang, N. Simaan, and H. Choset, "Utility-guided palpation for locating tissue abnormalities," *IEEE Robotics and Automation Letters*, vol. PP, no. 99, pp. 1–1, 2017.

[32] P. Chalasani, L. Wang, R. Yasin, N. Simaan, and H. Taylor, Russel, "Online estimation of organ geometry and tissue stiffness using continuous palpation," *in proceedings of Robotics and Automation Letters*, 2018.

[33] G. Mathew, S. Kannan, A. Surana, S. Bajekal, and K. R. Chevva, "Experimental implementation of spectral multiscale coverage and search algorithms for autonomous UAVs," in *AIAA Guidance, Navigation, and Control (GNC) Conf*, 2013, pp. 1–13.

[34] L. M. Miller and T. D. Murphey, "Trajectory optimization for continuous ergodic exploration on the motion group SE (2)," in *52nd IEEE Conference on Decision and Control*. IEEE, 2013, pp. 4517–4522.

[35] A. Gotovos, N. Casati, G. Hitz, and A. Krause, "Active learning for level set estimation," in *IJCAI*, 2013, pp. 1344–1350.

[36] J. Hauser, "A projection operator approach to the optimization of trajectory functionals," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 377–382, 2002.

[37] M. Kobilarov, "Cross-entropy motion planning," *The International Journal of Robotics Research*, vol. 31, no. 7, pp. 855–871, 2012.

[38] R. M. Neal, "Annealed importance sampling," *Statistics and computing*, vol. 11, no. 2, pp. 125–139, 2001.

[39] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Advances in neural information processing systems*, 2012, pp. 2951–2959.

[40] P. J. Van Laarhoven and E. H. Aarts, "Simulated annealing," in *Simulated Annealing: Theory and Applications*. Springer, 1987, pp. 7–15.

[41] A. Zhigljavsky and A. Žilinskas, "Stochastic global optimization". Springer Science & Business Media, 2007, vol. 9.

[42] R. Y. Rubinstein and D. P. Kroese, "The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning". Springer Science & Business Media, 2013.

[43] G. McLachlan and D. Peel, "Finite mixture models". John Wiley & Sons, 2004.

[44] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals of operations research*, vol. 134, no. 1, pp. 19–67, 2005.

[45] S. Seo, M. Wallat, T. Graepel, and K. Obermayer, "Gaussian process regression: Active data selection and test point rejection," in *Mustererkennung 2000*. Springer, 2000, pp. 27–34.

[46] E. Brochu, V. M. Cora, and N. De Freitas, "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," *arXiv preprint arXiv:1012.2599*, 2010.

[47] G. Mathew and I. Mezic, "Spectral multiscale coverage: A uniform coverage algorithm for mobile sensor networks." in *CDC*. Citeseer, 2009, pp. 7872–7877.

[48] E. Ayvali, A. Ansari, L. Wang, N. Simaan, and H. Choset, "Utility-guided palpation for locating tissue abnormalities," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 864–871, 2017.

[49] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, "An open-source research kit for the da Vinci® surgical system," in *ICRA*. IEEE, 2014, pp. 6434–6439.

[50] A. Bajo, R. E. Goldman, L. Wang, D. Fowler, and N. Simaan, "Integration and preliminary evaluation of an insertable robotic effectors platform for single port access surgery," in *ICRA*. IEEE, 2012, pp. 3381–3387.