Probabilistic Boundary Coverage for Unknown Target Fields with Large Perception Uncertainty and Limited Sensing Range

Binbin Li and Dezhen Song

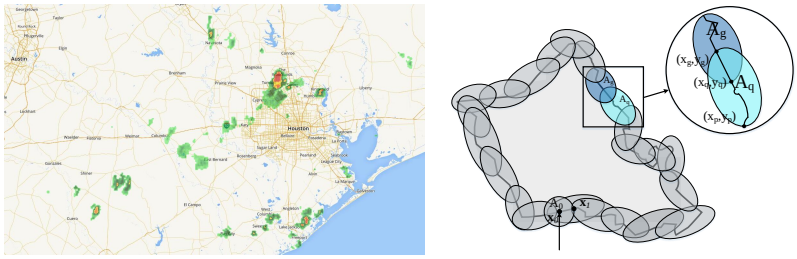
Abstract We introduce a new type of probabilistic boundary coverage problem where a robot has to enclose unknown target fields (UTFs) with large perception uncertainty and limited sensing range. When the robot gets closer to UTF and accumulates sufficient sensory readings, it employs Gaussian processes (GPs) as a local belief function to approximate field boundary distribution in an ellipse-shaped local region. The local belief function allows us to predict UTF boundary trends and establish an adjacent ellipse for further exploration. The process is governed by a depth-first search process until UTF is approximately enclosed by connected ellipses when the boundary coverage process ends. We formally prove that our boundary coverage process guarantees the enclosure above a given coverage ratio with a preset probability threshold. We have implemented our algorithm and tested it under different field types in simulation.

1 Introduction

Imagine that an unmanned aerial vehicle (UAV) is dispatched to map boundaries of excessive wind shear or low pressure regions in storm cells (see Fig. 1). The UAV has to plan its motion based on its on-board sensor readings to quickly enclose the unknown target field (UTF), which is a form of boundary coverage problem. However, UTFs often do not have a clear boundary or a priorly known dispersion function and the UAV has to get closer to the field to take multiple readings to predict

Binbin Li and Dezhen Song
Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843, United States of America
e-mail: {binbinli,dzsong}@cse.tamu.edu

This work was supported in part by National Science Foundation under IIS-1318638, NRI-1426752, NRI-1526200, and NRI-1748161.



(a) (b)

Fig. 1 Problem illustration. a) Wind shear region boundary coverage application: The green & orange clouds represent potential regions of interest that may contain UTFs. To map each UTF, we need to send an UAV to cage the UTF. b) Output of our boundary coverage algorithm is to cover the boundary using a sequence of connected ellipses.

field dispersion for boundary coverage. Moreover, the sensor readings often contain large uncertainties due to variations of the field itself or difficult sensing conditions. It is clear that regular boundary traversing techniques are not applicable. Such problems are not unusual. Another example is that an inspection robot is tasked to find thin hairline cracks on airport runway. These applications propose a new problem: how can we design a principled approach to ensure the robot can effectively cage UTFs under large perception uncertainty and limited sensing range.

We present this new boundary coverage problem and propose an algorithm to solve the problem. At each step, the robot accumulates sufficient sensory readings to instantiate Gaussian processes (GPs) as a local belief function to approximate field dispersion in an ellipse-shaped local region. The local belief function allows us to predict UTF boundary trends and establish adjacent ellipses for further exploration in next step. The process is governed by a depth-first search process until UTF is enclosed by connected ellipses (see Fig. 1(b)) with probability guarantees, as we formally prove that our boundary coverage process guarantees that the enclosed UTF is above a given coverage ratio with a preset probability threshold. We have implemented our method and tested with different types of UTFs (1D vs. 2D, smooth vs. non-smooth boundary, and convex vs. non-convex) in simulation. The results show that the algorithm always guarantees that the coverage ratio is above the given threshold for all testing cases, which is conformal to our analysis.

2 Related Work

This new UTF boundary coverage problem relates to many existing works including coverage in continuous fields, discrete space search, robotic caging in manipulation and grasping, and GPs.

Boundary coverage of UTFs is related to well-studied coverage problems because the latter also need to identify target field boundary before planning for cover-

age. However, identifying boundary might not be an issue if target field information is known whereas our problems focus on identifying boundaries. In coverage problems, the focus is to calculate optimal trajectories with respect to a given objective function for known field functions. Most existing methods depend on gradients or other information from the known field functions. Yun et al. [32] present the decentralized algorithms for the coverage with mobile robots on a graph. Miller et al. [16] use an ergodic control algorithm for the coverage with respect to the expected information density. Shnaps et al. [24] perform online tethered coverage in planar unknown environments using position and local obstacle detection sensors. Bekris et al. [2] apply cloud computing to efficiently plan the motion of new robot manipulators designed for flexible manufacturing floors. In our boundary coverage problem, we have to approximate field functions based on noisy and local/partial observations before planning for robot motion which further complicates the problem.

Searching for point/small objects without field functions can be viewed as a discrete search problem. The original search space could be either continuous or discrete but it is often discretized into grids or graphs in the searching process [5]. Acar et al. [1] introduce a hierarchical decomposition that combines the Morse decompositions and the generalized Voronoi diagram to ensure that the robot covers the searching domain. Paull et al. [17] present a sensor driven on-line approach for seabed coverage for mine countermeasure using grid-based coverage. Xu et al. [30] address the problem of effective graph coverage with environmental constraints and incomplete prior map information. Mannadiar et al. [14] guarantee the complete coverage of the free space based on the Boustrophedon cellular decomposition. Although these methods can be applied to UTF searching problem, their efficiency is problematic because the existing methods do not exploit the continuity of UTF structure.

Boundary coverage of UTFs is related to the caging problem in grasping which focuses on using geometric information of the manipulated object to generate stable grasps [6, 13, 19, 23, 29]. Vongmasa et al. [28] compute coverage parameters for 2D polygons to form a cage to transport an object. Pereira et al. [18] enable a team of robots with limited sensing range to achieve a condition of object closure, and move toward a goal position while maintaining the object closure condition. Ivan et al. [7] compute coverage without the reliance to geometrical detail but capture the topology of punctured euclidean spaces. Zarubin et al. [33] use geodesic balls to estimate object's surface in the presence of noise. These methods compute the waypoints for the object to generate a set of caging grasps. One issue that separates UTF caging from object caging is the issue of limited sensing range because objects are often small and fully covered by the sensor in the grasping process.

GPs become a powerful tool to represent unknown environments with limited observations [21]. GPs are widely used in geographical terrains [3, 27], nonlinear systems [31], complex environments [9, 12, 15], autonomous robotic vehicles [4], solar map constructions [20], and sensor networks [8] etc. Furthermore, GPs are capable to deal with the sensor information with uncertainties. In this case, we utilize GPs to approximate the distribution of the UTFs based on existing observations.

Our group has experiences in robotic searching problems. Our previous work develops algorithms and systems to detect an unknown wireless sensor network that intermittently transmits signals [10, 11, 25, 26]. The focus of our previous work is the modeling of stochastic properties of point targets where this work focuses on field targets.

3 Problem Formulation

3.1 Scenario and Assumptions

A mobile robot or UAV is dispatched to find an efficient path to enclose UTFs in an obstacle-free 2D space. This robot is equipped of sensors with a limited sensing range. The robot observation noise follows a Gaussian distribution with zero mean and variance σ^2. To formulate the UTF boundary coverage problem, we have the following assumptions:

1. We assume that the UTF is much larger than the sensing capabilities of the robot, and the moving speed of the robot is much faster than that of the UTF. This is common for large scale coverage occurring on the surface of the Earth.
2. The robot knows its current position using global positioning system and has a memory of where the robot has visited before.
3. We have no knowledge about UTF shapes.
4. GPs are capable of approximating the UTF boundary distribution.

3.2 UTF Properties and Modeling Perception

To further clarify our problem, let us define UTF and its key properties. Denote $\mathbf{z}_t$ to be the sensor readings at time t when the robot is at position $\mathbf{x}_t = [x(t)\ y(t)]^T \in \mathbb{R}^2$, and set $\mathbf{Z}_T = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_T\}$ as all observations sensed from the beginning of localization process up to time T. Denote $\mathbf{T}$ to be the UTF region and $\mathbf{x} = [x, y]^T \in \mathbb{R}^2$ to be a point in the 2d space. $\mathbf{T}$ is usually obtained by thresholding field boundary distribution function or geometric constraints which are described as follows,

$$\mathbf{T} := \left\{ \mathbf{x} \middle| \left\{ I_{\text{IN}} = \bigwedge_{\mathbf{z_t} \in \mathbf{Z_T}} \left(f(\mathbf{z}_t, \mathbf{x}_t, \sigma) \geq f_t \right) \right\} = 1 \text{ (is TRUE)} \right\}, \tag{1}$$

where indicator variable $I_{\text{IN}} \in \{0, 1\}$ is binary (boolean) variable depending on if the thresholding criteria are satisfied, $\bigwedge$ is logic AND operator, f is a nonnegative field boundary distribution function, σ is the standard deviation of observation noise, and thresholding value f_t for field value is predetermined by application.

Unfortunately, there is often no prior knowledge about shape and position of UTFs. Instead of computing from geometric constraints, indicator variable I_{IN} values are often obtained by thresholding sensory readings. We assume the robot is equipped with an omni-directional sensor with the maximum sensing distance d_s. If $d_s = 0$, the sensor becomes a point sensor such as a barometer measuring air pressure changes at its current position; if $d_s > 0$, the sensor may have measurable coverage like a camera covering the UTF. Let $\partial\mathbf{T}$ be the $\mathbf{T}$'s boundary. $\partial\mathbf{T}$ would be unmeasurable if there were no uncertainty in the sensing process. Due to the fact that $\mathbf{z}_t$ is normally distributed, we have,

$$\partial\mathbf{T} := \left\{ \mathbf{x} \middle| I_{\text{UTF}} = 1, \mathbf{x} \in \mathbf{T} \right\}, \tag{2}$$

where $I_{\text{UTF}} \in \{0,1\}$ is a binary indicator variable describing if the point is a boundary point and

$$I_{\text{UTF}} = \Big(\bigvee_{\mathbf{z_t} \in \mathbf{Z_T}} \big(f(\mathbf{z}_t, \mathbf{x}_t, \sigma) = f_t \big) \Big) \bigwedge \big(I_{\text{IN}=1} \big). \tag{3}$$

It is worth noting that (3) usually cannot be directly computed since we do not know f. With observations from multiple sensor readings, it can be predicted by a GP based on observations $(\mathbf{z}_t, \mathbf{x}_t)$. To ensure the coverage of a UTF, we just need to cover its boundary $\partial\mathbf{T}$. The coverage of UTF interior is trivial if $\partial\mathbf{T}$ is covered.

3.3 Problem Definition

To quantify the boundary coverage performance, we need to define a performance metric to determine the trade-off between quality and effort: let $\beta \in (0,1]$ be the coverage ratio threshold for UTF boundary. Denote $S_{\mathbf{T}}$ to be the length of the UTF's boundary $\partial\mathbf{T}$. We have the following boundary coverage success metric.

Condition 1 (Quality Metric) *The boundary coverage task for the UTF is considered to be accomplished if the covered boundary is no less than $\beta S_{\mathbf{T}}$ where $S_{\mathbf{T}}$ is the length of the UTF's boundary $\partial\mathbf{T}$.*

With inputs and quality metric defined, our problem is described as follows,

Problem 1 *Given the observation set $\mathbf{Z}_T$, plan robot trajectory $\mathbf{x}_{T+1}$ based on $\mathbf{x}_T$ to generate ellipses to cover $\partial\mathbf{T}$ with Condition 1 satisfied.*

4 System Modeling

The robot starts the boundary coverage process when it encounters an UTF region. The boundary coverage process generates a sequence of ellipses to enclose the UTF, which are treated as nodes for a depth-first search of a tree. Employing a sequence of

ellipses allows us to break down a long boundary traversing problem into a sequence of local problems to reduce problem scale. In each local problem, we can handle challenges associated with limited sensing range and observation uncertainty. We arrange each ellipse to have its long axis aligned with the boundary of the UTF to speed up the boundary converge process. At each ellipse, the robot accumulates observations to instantiate a Gaussian Process (GP) to predict the position of next ellipse along the boundary. The depth-first search ensures that the robot traverses the entire UTF boundary and coordinates ellipse generation as tree expansion which will be detailed later.

Let us clarify the use of index variables in the depth-first search progress. Ellipse $\mathbf{A}_q$ is where the robot is currently located. Index g refers to the total number of ellipses. Since we start the index with its root at $\mathbf{A}_0$, $\mathbf{A}_g$ is also the new ellipse during new node generation. $\mathbf{A}_p$ refers to which neighboring ellipse the robot uses to enter $\mathbf{A}_q$. The "neighboring ellipse" refers to either parent or child nodes of q on the search tree.

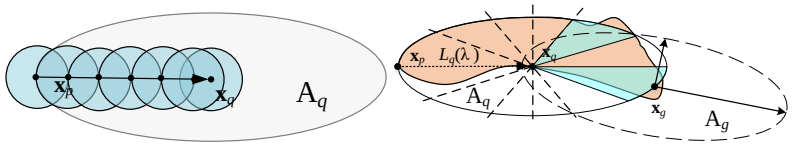
The whole process consists of two main steps: initialization and boundary traversing. We start with system initialization.

4.1 Ellipses, Robot Trajectory, Observation Set, and Initialization of the Depth-First Search

The boundary coverage process relies on a sequence of ellipses to track UTF boundaries. Define $\mathbf{A}_q$ as the q-th ellipse,

$$\mathbf{A}_q = \left\{ \mathbf{x} \middle| (\mathbf{x}-\mathbf{x}_q)^T \mathbf{C}_q (\mathbf{x}-\mathbf{x}_q) \leq 1 \right\}, \tag{4}$$

where $\mathbf{x}_q = [x_q, y_q]^T$ is its center point and $\mathbf{C}_q$ is a 2×2 positive definite matrix.



(a) Accumulate obs. in $\mathbf{A}_q$ (b) Compute $L_q(\lambda)$ and obtain $\mathbf{A}_g$

Fig. 2 The robot accumulates observations in $\mathbf{A}_q$ in blue shaded area in a), establishes belief functions in $\mathbf{A}_q$ using a GP based on observation set $\mathbf{O}_q$, which assists in determining $\mathbf{A}_g$ in b).

When the robot enters $\mathbf{A}_q$ from a neighboring ellipse center $\mathbf{x}_p$ to current ellipse center $\mathbf{x}_q$ along the shortest path, the robot trajectory set $\mathbf{u}_q$ in $\mathbf{A}_q$ can be defined as,

$$\mathbf{u}_q = \{\mathbf{x} | \mathbf{x} = \rho \mathbf{x}_p + (1-\rho)\mathbf{x}_q, \rho \in [0,1]\} \tag{5}$$

if ignoring obstacles. Denote $t_j \in \mathbb{R}$ as the exact continuous time at the moment of the discrete time j when the robot with constant velocity traverses from $\mathbf{x}_p$ to $\mathbf{x}_q$. Let $t_j - t_{j-1} = c_0$ for $j > 0$ where $c_0 \in \mathbb{R}_{++}$ is a constant variable. The index j is reset to zero every time when the robot reaches $\mathbf{x}_q$. During the travel, the robot accumulates observations from its on-board sensor to establish its observation set $\mathbf{O}_q$,

$$\mathbf{O}_q = \left\{ (\mathbf{x}, \mathbf{z}_{t_j}) \middle| \ |\mathbf{x} - \mathbf{x}_{t_j}| \leq d_s, \mathbf{x}_{t_j} \in \mathbf{u}_q \right\}, \tag{6}$$

and denote $\mathbf{x} \in \mathbf{X}_q$. Fig. 2(a) illustrates the observation set $\mathbf{O}_q$ coverage in $\mathbf{A}_q$.

The ellipse generation process initializes at the moment when the robot first encounters an UTF at point $\mathbf{x}_0$. It immediately generates $\mathbf{A}_0$ which is chosen to be a circle because it is likely that we do not have enough information to determine boundary direction yet. For $\mathbf{A}_0$, we have

$$\mathbf{C}_0 = \begin{bmatrix} 1/4d_s^2 & 0 \\ 0 & 1/4d_s^2 \end{bmatrix}. \tag{7}$$

On the other hand, $\mathbf{u}_0$ is slightly different from (5) because we do not have a neighboring ellipse. Alternatively, we substitute $\mathbf{x}_p$ with $\mathbf{x}_{\text{enter}}$ in (5) to obtain $\mathbf{u}_0$ where $\mathbf{x}_{\text{enter}}$ is the point of entry to $\mathbf{A}_0$ during the global search process. Consequently, we have a non-empty observation set $\mathbf{O}_0$.

4.2 Depth-First Search-based Boundary Traversing

We employ a depth-first search over a tree, which contains all ellipses as tree nodes. Ellipses $\mathbf{A}_0$ is the root node of the ellipse tree. Each node q stores its $\mathbf{u}_q$ and $\mathbf{O}_q$. $\mathbf{O}_q$ is updated as the robot travels inside the ellipse. As mentioned before, the robot only moves between ellipse centers $\mathbf{x}_q$ of neighboring tree nodes along a linear path because we ignore the obstacle in the process. This also yields a piecewise linear trajectory for the robot.

4.2.1 Branching Method

At each ellipse $\mathbf{A}_q$, the robot uses $\mathbf{O}_q$ to instantiate a GP [21] approximating the belief function for $\partial\mathbf{T}$ in $\mathbf{A}_q$ which provides information to determine $\mathbf{A}_g$. More specifically, for an $\mathbf{x} \in \mathbf{A}_q$, the GP provides posterior distribution $P(I_{\text{UTF}}|\mathbf{x}, \mathbf{O}_q)$ for the UTF region to stand for the field function. It is worth noting that $\mathbf{O}_q$ theoretically contains infinite number of observations. To facilitate computation, we sample $\mathbf{O}_q$ using a local lattice according to sensor spatial resolution or task needs to accelerate GP training time. Recall $\mathbf{O}_q$ is continuously updated according to robot trajectory set. This leads to a recursive Bayesian estimation process, which can be computed using a two-phase approach [22]:

1. Update Phase: For an $\mathbf{x}_* \in \mathbf{O}_q$,

$$P(I^*_{\text{UTF}}|\mathbf{x}_*, \mathbf{O}_q) = bel(I^*_{\text{UTF}}|f^*(\mathbf{x}_*), \sigma^2), \tag{8}$$

   where the latent function f^* is represented by GP, an approximation of (3).
2. Prediction Phase: The GP provides posterior distribution for the UTF boundary for a given $\mathbf{x} \notin \mathbf{O}_q$,

$$P(I_{\text{UTF}}|\mathbf{x}, \mathbf{O}_q) = \overline{bel}(I_{\text{UTF}}|\mu_I, \sigma_I^2). \tag{9}$$

   Here, μ_I and σ_I^2 are the expectation and variance of the posterior distribution related to kernel function, which characterizes the correlation between the function values at different locations, namely, $f^*(\mathbf{x}_i)$ and $f^*(\mathbf{x}_j)$. We employ a histogram intersection kernel K as

$$K(\mathbf{x}_i, \mathbf{x}_j) = \sum_{d=1}^{2} \min(\mathbf{x}_i(d), \mathbf{x}_j(d)). \tag{10}$$

   for $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^2$ with $\mu_I = \mathbf{k}_*^T(\mathbf{K} + \sigma^2\mathbf{I})^{-1} I^*_{\text{UTF}}$ and $\sigma_I^2 = \mathbf{k}_{**}\mathbf{k}_*^T(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{k}_*$ where $\mathbf{k}_* = K(\mathbf{X}_q, \mathbf{x})$, $\mathbf{K}$ is the kernel matrix of the training data $\mathbf{X}_q$, $\mathbf{k}_{**} = K(\mathbf{x}, \mathbf{x})$, and $\mathbf{I}$ is an identity matrix.

To approximate UTF boundary $\partial\mathbf{T}$, we calculate the level set $L_q(\lambda)$ where threshold $\lambda > 0$, to cover regions with high probability of containing the boundary. This is done by thresholding on $P(I_{\text{UTF}}|\mathbf{x}, \mathbf{O}_q)$.

$$L_q(\lambda) = \left\{\mathbf{x} | P(I_{\text{UTF}}|\mathbf{x}, \mathbf{O}_q) \geq \lambda, \mathbf{x} \in \mathbf{A}_q\right\}. \tag{11}$$

The value of threshold λ is determined by coverage ratio threshold β in Condition 1 and will be discussed in Section 5.2.

Now let us show how to determine $\mathbf{x}_g$, center of the new node on the ellipse tree. The boundary of $L_q(\lambda)$ intercepts the boundary of $\mathbf{A}_q$ and generates a set of points $\mathbf{X}_q^L$. As illustrated Fig. 2(b), we evenly divide $\mathbf{A}_q$ into $k_d = 12$ sectors with each sector spans $2\pi/k_d = \pi/6$. For each sector, we identify a middle angle boundary point $\mathbf{x}_q^s$ by intercepting $\mathbf{A}_q$ boundary with the ray shooting from the ellipse center along the middle angle (π/k_d from the sector side). We add $\mathbf{x}_q^s$ to the candidate solution set $\mathbf{X}_q^*$ for $\mathbf{x}_g$ if we can find a solution in set $\mathbf{X}_q^L$ located on the corresponding sector boundary. This means that we use $\mathbf{X}_q^L$ to filter out less likely candidate center locations. To avoid repeated search, we remove $\mathbf{x}_q^s$ of the sector where the robot enters $\mathbf{A}_q$ from the candidate solution set $\mathbf{X}_q^*$. Therefore, set $\mathbf{X}_q^*$ only contains branches that robot has not visited.

4.2.2 Termination Scenarios

With $\mathbf{X}_q^*$ introduced, let us explain the termination condition of the search, which has two scenarios. When $\mathbf{X}_q^*$ is empty, it means that the robot reaches the extreme

end of the UTF which is the leaf of the depth-first search tree. Now the only choice is to let the robot traverse back on the tree to the parent ellipse and check if the candidate solution set of the parent ellipse is non-empty. If a non-empty node is found, we enter the tree expansion case as described in the sub-section. Otherwise, we keep back-traversing the robot and repeat this process along the tree to upper level parent node. We update p and q in the process.

If we return to the root and find $\mathbf{X}_0^*$ is empty, it means the robot has covered the entire search tree in the boundary traversing. This concludes the first termination scenario and the depth-first search ends. This scenario occurs a lot with curves, lines or other thin UTFs.

For a compact and sizable UTF, it is likely that the robot loops around the UTF (see Fig.1(b)), which is the second termination scenario. Because it leads the robot to travel back to $\mathbf{A}_0$, we can identify this by verifying if $\mathbf{x}_q \in \mathbf{A}_0$ is true. We also need to remove the candidate solution from the sector that contains $\mathbf{x}_q$ in $\mathbf{X}_0^*$. Again, if $\mathbf{X}_0^*$ is empty, the search ends.

4.2.3 Node/Ellipse Generation

Next is about tree expansion or node generation process. It happens if the candidate solution set $\mathbf{X}_q^*$ is non-empty for the original q or the updated q from the back-traversing process, then we choose the $\mathbf{x}_g$ as follows,

$$\mathbf{x}_g = \arg\max_{\mathbf{x}\in\mathbf{X}_q^*} \left(\arccos \frac{\langle \mathbf{x}_{q^-}, \mathbf{x}_{*,q} \rangle}{\|\mathbf{x}_{q^-}\| \cdot \|\mathbf{x}_{*,q}\|} \right). \tag{12}$$

where $\mathbf{x}_{q^-} = \mathbf{x}_q - \mathbf{x}_p$ represents the vector describing how the robot enters $\mathbf{A}_q$, $\mathbf{x}_{*,q} = \mathbf{x} - \mathbf{x}_q$, $\|\cdot\|$ is vector l-2 norm, and $\langle \cdot, \cdot \rangle$ is vector inner product. This means that the candidate is the closest to the direction that the robot enters $\mathbf{A}_q$. Once $\mathbf{x}_g$ is chosen, we remove it from $\mathbf{X}_q^* = \mathbf{X}_q^* \setminus \{\mathbf{x}_g\}$. This is to avoid repeated search when we return to $\mathbf{A}_q$ in the depth-first search process. Again, set $\mathbf{X}_q^*$ keeps track of the visited branch of the search tree. New node g is added to the tree with its parent to be q and $g = g + 1$.

After obtaining new center $\mathbf{x}_g$, we place $\mathbf{A}_g$ into its position by determining $\mathbf{C}_g$. Set the long and short axes of $\mathbf{A}_g$ to be $4d_s$ and $2d_s$, respectively. To approximate the UTF boundary, we want the long axis of $\mathbf{A}_g$ to be aligned with $\mathbf{x}_g - \mathbf{x}_q$ as follows,

$$\mathbf{C}_g = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \frac{1}{16d_s^2} & 0 \\ 0 & \frac{1}{4d_s^2} \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}, \tag{13}$$

where θ is determined by

$$\theta = \arctan\left(\frac{y_g - y_q}{x_g - x_q}\right). \tag{14}$$

After $\mathbf{A}_g$ is determined, the robot motion is also obtained as $\mathbf{u}_g$. As the robot moves toward $\mathbf{x}_g$, we update $\mathbf{X}_q^*$, p, and q accordingly.

5 Boundary Coverage Performance Analysis

The remaining question is how good the boundary coverage quality is and how to guarantee Condition 1. We first analyze the coverage quality for a point $\mathbf{x}$ in $\mathbf{A}_q$ and then aggregate it into the entire boundary.

5.1 Probability Bounds for a Point $\mathbf{x}$ *in* $\mathbf{A}_q$

The UTF boundary $\partial\mathbf{T}$ is covered by the level set $L_q(\lambda)$ which is generated by the thresholding on $P(I_{\text{UTF}}|\mathbf{x},\mathbf{O}_q)$ in (11). It is important to understand $P(I_{\text{UTF}}|\mathbf{x},\mathbf{O}_q)$. $P(I_{\text{UTF}}|\mathbf{x},\mathbf{O}_q)$ is a function of its condition $(\mathbf{x},\mathbf{O}_q)$ which means it is still a random variable because $(\mathbf{x},\mathbf{O}_q)$ are random variables. Since it is a random variable, it has an expectation $E\left(P(I_{\text{UTF}}|\mathbf{x},\mathbf{O}_q)\right)$. $E\left(P(I_{\text{UTF}}|\mathbf{x},\mathbf{O}_q)\right)$ can be estimated by averaging $P(I_{\text{UTF}}|\mathbf{x},\mathbf{O}_q)$ across all points using the output from GPs. $E\left(P(I_{\text{UTF}}|\mathbf{x},\mathbf{O}_q)\right)$ can be viewed as an observation of the unconditional distribution $P(I_{\text{UTF}})$. $P(I_{\text{UTF}})$ is the mean value of $P(I_{\text{UTF}}|\mathbf{x},\mathbf{O}_q)$. $P(I_{\text{UTF}})$ is important because it can help us to determine if we miss any UTF boundary points when performing thresholding in (11). We do not know $P(I_{\text{UTF}})$ but we know its low bound in probability as follows.

Lemma 1 *with* $1-\tau$ *probability,* $P(I_{UTF})$ *which is the unconditional probability that a point in* $\mathbf{A}_q$ *belongs to UTF boundary has the following lower bound* B_q^-*,*

$$P(I_{UTF}) \geq B_q^- = \inf_{\eta>0}\left\{E(P(I_{UTF}|\mathbf{x},\mathbf{O}_q)) - \frac{\eta}{2} - \frac{1}{l_{\max}\eta}\log\frac{1}{\tau}\right\}, \tag{15}$$

where $\tau \in (0,1)$ *is a chosen small number,* $l_{\max}$ *is the set cardinality of* $L_q(\lambda)$*, and nonnegative variable* η *is determined by the* inf *computation.*

Proof. The lower bound of $P(I_{\text{UTF}})$ can be proved by relating it to its estimator $E(P(I_{\text{UTF}}|\mathbf{x},\mathbf{O}_q))$. We construct the following probability event E_0 for $t>-1$,

$$P(E_0) = P\left(P\left(I_{\text{UTF}}\right) - \frac{\eta}{2} - E\left(P(I_{\text{UTF}}|\mathbf{x},\mathbf{O}_q)\right) \leq -(t+1)\right), \tag{16}$$

This is equivalent to,

$$P(E_0) = P(e^{-\eta l_{\max}\left(P(I_{\text{UTF}})-\frac{\eta}{2}-E\left(P(I_{\text{UTF}}|\mathbf{x},\mathbf{O}_q)\right)\right)} \geq e^{\eta l_{\max}(t+1)}), \tag{17}$$

and apply Markov's inequality, we have,

$$P(E_0) \leq e^{-\eta l_{\max}(t+1)}E\left(e^{-\eta l_{\max}\left(P(I_{\text{UTF}})-\frac{\eta}{2}-E\left(P(I_{\text{UTF}}|\mathbf{x},\mathbf{O}_q)\right)\right)}\right). \tag{18}$$

Since $e^{\eta l_{\max}\frac{\eta}{2}} \geq 1$, we multiply it to the right hand side of (18) and move the constant terms outside the expectation $E(\cdot)$, we have

$$P(E_0) \leq e^{-\eta l_{\max}\left(t-\frac{\eta}{2}\right)-\eta l_{\max}\left(P(I_{\text{UTF}})-\frac{\eta}{2}\right)} \times E\left(e^{\eta l_{\max}\left(E\left(P(I_{\text{UTF}}|\mathbf{x},\mathbf{O}_q)\right)-1\right)}\right) \tag{19}$$

Using the fact that $E(P(I_{\text{UTF}}|\mathbf{x},\mathbf{O}_q)) \leq 1$, we know

$$P(E_0) \leq e^{-\eta l_{\max}\left(t-\frac{\eta}{2}\right)-\eta l_{\max}P(I_{\text{UTF}})+\frac{\eta^2 l_{\max}}{2}} E\left(e^0\right), \tag{20}$$

which leads to

$$P(E_0) \leq e^{-\eta l_{\max}\left(t-\frac{\eta}{2}\right)+\frac{\eta^2 l_{\max}}{2}} = \tau. \tag{21}$$

We can solve t from (21) and plug it back to (16) and the lemma is proved.

5.2 Probability of Covering an UTF Boundary Point in Level Set Construction

Lower bound B_q^- can help us determine the probability that the UTF boundary is captured by the GP in level set construction (11). It also helps determine how to choose λ. It is clear that a reasonable λ should be smaller than B_q^-. Defining $F_{(\mu,\sigma^2)}(x)$ as the cumulative probability function of the Gaussian distribution $N(\mu,\sigma^2)$, we have the following lemma,

Lemma 2 *For a given lower bound B_q^- and $\lambda \leq B_q^-$ at point $\mathbf{x} \in \mathbf{A}_q$ when computing the level set $L_q(\lambda)$, the probability that an UTF boundary point satisfies* (11) *is no less than $(1-F_{(B_q^-,\sigma_I^2)}(\lambda))(1-\tau)$ where σ_I^2 is the variance of $P(I_{UTF}|\mathbf{x},\mathbf{O}_q)$.*

Proof. From GP model, we know $P(I_{\text{UTF}}|\mathbf{x},\mathbf{O}_q)$ is a Gaussian distribution with mean $\mu_I = P(I_{\text{UTF}})$ and variance σ_I^2. For a given λ and B_q^-, the fact that an UTF boundary point is captured means the following conditional event $P(I_{\text{UTF}}|\mathbf{x},\mathbf{O}_q) \geq \lambda \,|\, \mu_I \geq B_q^-$ occurs. We now compute its probability by further conditioning on E_0 and applying the fact that B_q^- is not a deterministic bound,

$$\begin{aligned}
&P(P(I_{\text{UTF}}|\mathbf{x},\mathbf{O}_q) \geq \lambda \,|\, \mu_I \geq B_q^-) \\
&\quad = P(P(I_{\text{UTF}}|\mathbf{x},\mathbf{O}_q) \geq \lambda \,|\, \mu_I \geq B_q^-, E_0)(1-\tau) \\
&\qquad + P(P(I_{\text{UTF}}|\mathbf{x},\mathbf{O}_q) \geq \lambda \,|\, \mu_I \geq B_q^-, \overline{E_0})\tau \\
&\quad \geq P(P(I_{\text{UTF}}|\mathbf{x},\mathbf{O}_q) \geq \lambda \,|\, \mu_I \geq B_q^-, E_0)(1-\tau) \\
&\quad \geq P(P(I_{\text{UTF}}|\mathbf{x},\mathbf{O}_q) \geq \lambda \,|\, \mu_I = B_q^-, E_0)(1-\tau).
\end{aligned} \tag{22}$$

Therefore, we have

$$P(P(I_{\text{UTF}}|\mathbf{x},\mathbf{O}_q) \geq \lambda \,|\, \mu_I \geq B_q^-)) \geq (1-F_{(B_q^-,\sigma_I^2)}(\lambda))(1-\tau). \tag{23}$$

Note that σ_I^2 can be obtained using GP outputs.

5.3 Ensure Boundary Coverage Quality

Recall that we have coverage ratio threshold β in boundary coverage quality metric defined in Condition 1 in our problem definition in Section 3.3. To satisfy the condition, we choose λ and τ accordingly. This means that we can set

$$(1-F_{(B_q^-,\sigma_I^2)}(\lambda))(1-\tau)=\beta, \tag{24}$$

to help obtain correct threshold λ. We do have some freedom in choosing τ to assist the selection of λ. It is not difficult to perform a binary search to obtain it.

5.4 Algorithm and Complexity Analysis

Algorithm 1: Boundary Coverage of an UTF

Input : robot observations $\mathbf{O}_q$
Output: robot trajectory in set $\mathbf{G}$

Generate $\mathbf{A}_0$ at $\mathbf{x}_0$; ▷$O(1)$
Stack $S=\{\}$; ▷$O(1)$
Push$(S,\mathbf{x}_0)$; ▷$O(1)$
while *S is not empty* **do** ▷$O(\nu)$
  $\mathbf{x}_q=\text{Pop}(S)$; ▷$O(1)$
  if *Visited*$(\mathbf{x}_q):=$ *FALSE* **then**
    Visited$(\mathbf{x}_q):=$ TRUE; ▷$O(1)$
    Update p and q according to trajectory; ▷$O(1)$
    $P(I_{\text{UTF}}|\mathbf{x},\mathbf{O}_q)$ using GPs; ▷$O(n_q \log n_q)$
    Obtain B_q^- according to (15); ▷$O(\log n_s)$
    Calculate λ through (24); ▷$O(1)$
    $\mathbf{X}_q^*=\bigcup \mathbf{x}_q^s \backslash \{\mathbf{x}_p\}$; ▷$O(1)$
    if $\mathbf{X}_q^*=\emptyset$ *or* $\mathbf{x}_q \in \mathbf{A}_0$ *for* $q\neq 0$ **then**
      if *Robot at* $\mathbf{A}_0$ **then**
        Break; ▷$O(1)$
      else
        Travese back to parent node; ▷$O(1)$
    else
      Obtain $\mathbf{A}_g$ and move to $\mathbf{x}_g$; ▷$O(1)$
    Push$(S,\mathbf{x}_g)$; ▷$O(1)$
    $\mathbf{G}=\mathbf{G}\bigcup\{\mathbf{x}_g\}$; ▷$O(1)$

Algorithm 1 summarizes the proposed method. To overcome the computational limitations of naive GPs with time complexity $O(n_q^3)$, we employ a Gaussian process with generalized histogram intersection kernels to speed up the naive GPs to

$O(n_q \log n_q)$ where n_q is the set cardinality of $\mathbf{O}_q$ [22]. Recall that ν refers the number of ellipse centers, and we initially set $\mathbf{O}_q = \emptyset$, $\mathbf{X}_q^* = \emptyset$, and $\mathbf{G} = \emptyset$. For the coverage process, Algorithm 1 details the pseudocode, which leads to the following complexity result.

Lemma 3 *Boundary coverage algorithm for an UTF runs in $O(\nu n_q \log n_q)$ time, where n_q is the set cardinality of $\mathbf{O}_q$, and ν is the number of ellipses to enclose the UTF.*

6 Experimental Result

We have implemented the proposed method in Matlab on a Laptop PC with an Intel(R) Core$^{\text{TM}}$2 i7-3517U CPU@1.90GHz and 8 GB memory. To verify the proposed local coverage method, we simulate different field shapes to test our approach (see Table 1). It includes both simple geometric shapes such as lines, circles, squares, and two complex shapes including storm cells and an island. Fig. 3 shows the images of the two complex shapes. Each image has a resolution of 720×480 pixels.

To measure our algorithm's boundary coverage capability, we define $\mathbf{A}_L$ as the boundary area covered using our method, and $\mathbf{A}$ as the actual area the UTF boundary occupies. We evaluate the coverage ratio β_r by using:

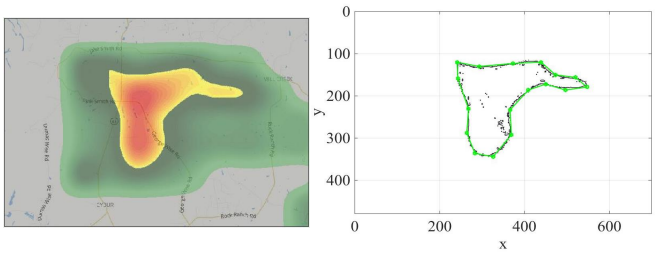
$$\beta_r = \frac{|\mathbf{A}_L \cap \mathbf{A}|}{|\mathbf{A}|}. \tag{25}$$

We set the value $d_s = 1$ for the simple geometric shapes and d_s equal to 20 pixels for the two image-based case.

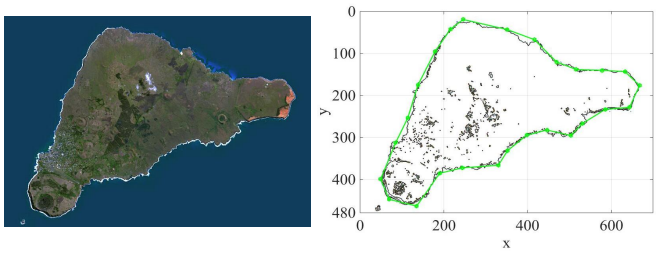
Table 1 local coverage experiment settings and results.

UTF Type	$\mathbf{f}_j(\mathbf{x})$	Dimension	Boundary	Shape	β_r	β
Line	$2x-y+5=0$ with $3 \le x \le 12$	1D	smooth	convex	98.12%	95.00%
Circle	$25-(x-6)^2-(y-6)^2 \ge 0$	2D	smooth	convex	97.56%	95.00%
Square	$\lvert x-5\rvert \le 6, \lvert y-5\rvert \le 6$	2D	non-smooth	convex	93.34%	90.00%
Storm cell	see Fig. 3(a)	2D	non-smooth	non-convex	87.32%	85.00%
Island	see Fig. 3(b)	2D	non-smooth	non-convex	88.59%	85.00%

The experimental settings and results are shown in Table 1. The last two rows show that for a given different threshold β, our algorithm has guaranteed that the actual coverage is no less than β for all testing cases, which is conformal to our analysis. Also, the sample robot trajectories for boundary traversing are illustrated as green piecewise linear curves in the right side of Fig. 3. It is clear that the robot successfully covers the both testing cases.



(a)



(b)

Fig. 3 Local coverage testing with real image data and robot coverage path. a) A weather radar map showing a storm cell. b) An island map.

7 Conclusion and Future Work

We introduced a new UTF boundary coverage problem with many applications. We reported a probabilistic boundary coverage method for addressing UTF problems. We generated a sequence of ellipses to cover UTF boundary. The introduction of ellipse sequence also allowed us to decompose the long trajectory traversing problem to multiple local problems with each ellipse represented a local problem. In each local problem, we employed Gaussian processes (GPs) as a local belief function to approximate field distribution. The local belief function allows us to predict UTF boundary trends and establish adjacent ellipses for further exploration. The process was governed by a depth-first search process until UTF is approximately enclosed by connected ellipses. We formally proved that our boundary coverage process guarantees the enclosure above a given coverage ratio with a preset probability threshold. We implemented our algorithm and successfully tested it in experiments with different field types.

In the future, we will perform physical experiments using bridge deck scanning and storm cell mapping applications. We will provide overall trajectory length prediction for the algorithm. We will consider a multiple robot team and moving targets.

We will also consider robots/UAVs with kinodynamic constraints in the trajectory generation.

Acknowledgment

Thanks for C. Chou, H. Cheng, S. Yeh, A. Kingery, A. Angert, H. Li, and T. Sun for their inputs and Y. Sun, M. Jin, D. Wang, and Y. Yu for their contributions to the NetBot Laboratory, Texas A&M University.

References

[1] Acar EU, Choset H, Lee JY (2006) Sensor-based coverage with extended range detectors. IEEE Transactions on Robotics 22(1):189–198
[2] Bekris K, Shome R, Krontiris A, Dobson A (2016) Reducing roadmap size for network transmission in support of cloud automation. IEEE Robotics and Automation Magazine
[3] Bryan B, Nichol RC, Genovese CR, Schneider J, Miller CJ, Wasserman L (2006) Active learning for identifying function threshold boundaries. In: Advances in neural information processing systems, pp 163–170
[4] Chen J, Low KH, Yao Y, Jaillet P (2015) Gaussian process decentralized data fusion and active sensing for spatiotemporal traffic modeling and prediction in mobility-on-demand systems. IEEE Transactions on Automation Science and Engineering 12(3):901–921
[5] Chung TH, Hollinger GA, Isler V (2011) Search and pursuit-evasion in mobile robotics. Autonomous robots 31(4):299
[6] Fink J, Hsieh MA, Kumar V (2008) Multi-robot manipulation via caging in environments with obstacles. In: IEEE International Conference on Robotics and Automation, pp 1471–1476
[7] Ivan V, Vijayakumar S (2015) Space-time area coverage control for robot motion synthesis. In: International Conference on Advanced Robotics (ICAR), IEEE, pp 207–212
[8] Jadaliha M, Xu Y, Choi J, Johnson N, Li W (2013) Gaussian process regression for sensor networks under localization uncertainty. IEEE Transactions on Signal Processing 61(2):223–237, DOI 10.1109/TSP.2012.2223695
[9] Jing W, Newman W (2002) Improving robotic assembly performance through autonomous exploration. IEEE International Conference on Robotics and Automation p 3303
[10] Kim CY, Song D, Xu Y, Yi J, Wu X (2014) Cooperative search of multiple unknown transient radio sources using multiple paired mobile robots. IEEE Transactions on Robotics 30(5):1161–1173
[11] Kim CY, Song D, Yi J, Wu X (2015) Decentralized searching of multiple unknown and transient radio sources with paired robots. Engineering 1(1):058–065
[12] Low KH, Chen J, Dolan JM, Chien S, Thompson DR (2012) Decentralized active robotic exploration and mapping for probabilistic field classification in environmental sensing. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1, International Foundation for Autonomous Agents and Multiagent Systems, pp 105–112
[13] Maeda Y, Kodera N, Egawa T (2012) Caging-based grasping by a robot hand with rigid and soft parts. In: IEEE International Conference on Robotics and Automation (ICRA), pp 5150–5155
[14] Mannadiar R, Rekleitis I (2010) Optimal coverage of a known arbitrary environment. In: IEEE International Conference on Robotics and Automation, pp 5525–5530

[15] Marchant R, Ramos F (2012) Bayesian optimisation for intelligent environmental monitoring. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 2242–2249
[16] Miller LM, Silverman Y, MacIver MA, Murphey TD (2016) Ergodic exploration of distributed information. IEEE Transactions on Robotics 32(1):36–52
[17] Paull L, Saeedi S, Seto M, Li H (2013) Sensor-driven online coverage planning for autonomous underwater vehicles. IEEE/ASME Transactions on Mechatronics 18(6):1827–1838
[18] Pereira GA, Campos MF, Kumar V (2004) Decentralized algorithms for multi-robot manipulation via caging. The International Journal of Robotics Research 23(7-8):783–795
[19] Pipattanasomporn P, Makapunyo T, Sudsang A (2016) Multifinger caging using dispersion constraints. IEEE Transactions on Robotics 32(4):1033–1041
[20] Plonski PA, Vander Hook J, Isler V (2016) Environment and solar map construction for solar-powered mobile systems. IEEE Transactions on Robotics 32(1):70–82
[21] Rasmussen CE (2006) Gaussian Processes for Machine Learning. The MIT Press
[22] Rodner E, Freytag A, Bodesheim P, Denzler J (2012) Large-scale gaussian process classification with flexible adaptive histogram kernels. In: European Conference on Computer Vision, Springer, pp 85–98
[23] Rodriguez A, Mason MT, Ferry S (2012) From caging to grasping. The International Journal of Robotics Research 31(7):886–900
[24] Shnaps I, Rimon E (2014) Online coverage by a tethered autonomous mobile robot in planar unknown environments. IEEE Transactions on Robotics 30(4):966–974
[25] Song D, Kim CY, Yi J (2011) On the time to search for an intermittent signal source under a limited sensing range. IEEE Transactions on Robotics 27(2):313–323
[26] Song D, Kim CY, Yi J (2012) Simultaneous localization of multiple unknown and transient radio sources using a mobile robot. IEEE Transactions on Robotics 28(3):668–680, DOI 10.1109/TRO.2012.2183069
[27] Vasudevan S, Ramos F, Nettleton E, Durrant-Whyte H (2009) Gaussian process modeling of large-scale terrain. Journal of Field Robotics 26(10):812–840, DOI 10.1002/rob.20309
[28] Vongmasa P, Sudsang A (2006) Coverage diameters of polygons. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, pp 4036–4041
[29] Wan W, Fukui R (2016) Efficient planar caging test using space mapping. IEEE Transactions on Automation Science and Engineering
[30] Xu L, Stentz A (2011) An efficient algorithm for environmental coverage with multiple robots. In: IEEE International Conference on Robotics and Automation (ICRA), pp 4950–4955
[31] Yang K, Keat Gan S, Sukkarieh S (2013) A gaussian process-based RRT planner for the exploration of an unknown and cluttered environment with a UAV. Advanced Robotics 27(6):431–443, DOI 10.1080/01691864.2013.756386
[32] Yun Sk, Rus D (2014) Distributed coverage with mobile robots on a graph: locational optimization and equal-mass partitioning. Robotica 32(02):257–277
[33] Zarubin D, Pokorny FT, Toussaint M, Kragic D (2013) Caging complex objects with geodesic balls. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp 2999–3006