Hindawi Publishing Corporation
Journal of Robotics
Volume 2012, Article ID 590479, 14 pages
doi:10.1155/2012/590479

Research Article

Partition Learning for Multiagent Planning

Jared Wood[1] **and J. Karl Hedrick**[2]

[1] *Vehicle Dynamics Lab and Center for Collaborative Control of Unmanned Vehicles, Department of Mechanical Engineering, University of California, Berkeley, 6141 Etcheverry Hall, Berkeley, CA 94720-1740, USA*
[2] *Vehicle Dynamics Lab, Department of Mechanical Engineering, University of California, Berkeley, 6141 Etcheverry Hall, Berkeley, CA 94720-1740, USA*

Correspondence should be addressed to Jared Wood, jared.jwood@gmail.com

Received 30 March 2012; Revised 27 July 2012; Accepted 10 August 2012

Academic Editor: Duško Katić

Automated surveillance of large geographic areas and target tracking by a team of autonomous agents is a topic that has received significant research and development effort. The standard approach is to decompose this problem into two steps. The first step is target track estimation and the second step is path planning by optimizing directly over target track estimation. This standard approach works well in many scenarios. However, an improved approach is needed for the scenario when general, nonparametric estimation is required, and the number of targets is unknown. The focus of this paper is to present a new approach that inherently handles the task to search for and track an *unknown* number of targets within a *large* geographic area. This approach is designed for the case when the search is performed by a team of autonomous agents and target estimation requires general, nonparametric methods. There are consequently very few assumptions made. The only assumption made is that a time-changing target track estimation is available and shared between the agents. This estimation is allowed to be general and nonparametric. Results are provided that compare the performance of this new approach with the standard approach. From these results it is concluded that this new approach improves search and tracking when the number of targets is unknown and target track estimation is general and nonparametric.

1. Introduction

The advancement of computing technology has enabled the practical development of intelligent autonomous systems. Intelligent autonomous systems can be used to perform difficult sensing tasks. One such sensing task is to search for and track targets over large geographic areas. Much research has gone into this task resulting in a standard approach. This standard approach decomposes the problem into two steps.

(1) Target track estimation.
(2) Agent path optimization based on target track estimation.

Significant research has been accomplished for each of these steps. Target track estimation has largely been solved [1–5] and this paper proposes no new methods for target track estimation. Agent path optimization based on target track estimation has been solved for many scenarios. However, the general scenario of when the number of targets is unknown still requires more development.

The standard approach in general works particularly well when it can be assumed that there is a single target [6–14]. And in many scenarios the standard approach works well even when there are multiple targets [15, 16]. However, when there are multiple targets, methods following the standard approach start requiring limiting assumptions on the problem. For example, many methods require that the geographic area be easily scanned so that there are frequent target detections throughout the geographic area. When this can be assumed it allows for simpler estimation, such as Gaussian distributions, and consequently agent paths can be more readily optimized. However, these methods do not extend well when the geographic area is too large to scan quickly. When the geographic area is large, agents frequently do *not* detect targets. These no-detection events must be utilized to estimate the target tracks [6, 10, 12]. General recursive Bayesian estimation methods are required to accomplish estimation for this case. This results in a general, nonparametric estimate of the target tracks. Yet, because of this generality, as the number of targets increases it becomes very difficult to optimize sensor paths over the target track estimates.

The focus of this paper is then on the problem of search and tracking an *unknown* number of possibly multiple targets in a *large* geographic area utilizing a team of autonomous, sensing agents. As such, very few assumptions are placed on the problem. All that is assumed is that time-changing target track estimates are provided and shared by the team of autonomous agents [10, 11, 13, 15, 16]. The target track estimates are completely general and nonparametric and the geographic area is too large to scan quickly.

A solution to this general problem is provided by proposing a new approach to perform autonomous search and tracking that inherently handles the case of an unknown number of targets in a general, nonparametric estimation setting. The performance of this new approach is then compared with the standard approach. The specific method that will be used for comparison is direct optimization over target estimation distribution [6, 10, 14].

2. Problem Structure: New Approach

In this paper a new approach is presented to perform autonomous search and tracking over large geographic areas. The union of such large geographic areas will be referred to as the surveillance area S in this paper. This new approach is designed to inherently handle the case of the number of targets being unknown and target track estimation being general and non-parametric. Figure 1 provides an overview of this approach. Notice, instead of the standard two-step approach there are three steps. These three steps are

(1) target density estimation [17],

(2) partition learning based on target density estimation,

(3) agent path planning based on partitions.

With this decomposition of the problem, separate subproblems are defined for each step. Each step will be described in the following sections. Two of these steps leverage existing work by extending existing methods to fit the structure of the problem presented in this paper. These steps are target density estimation and agent path planning based on partitions. The required extensions will be presented in this paper. Partition learning requires further development. Consequently most discussion provided in this paper will focus on the partition learning step.

Also note, these steps are repeated at each time instance. As the target estimation changes with time, so do the partitions and the agent paths.

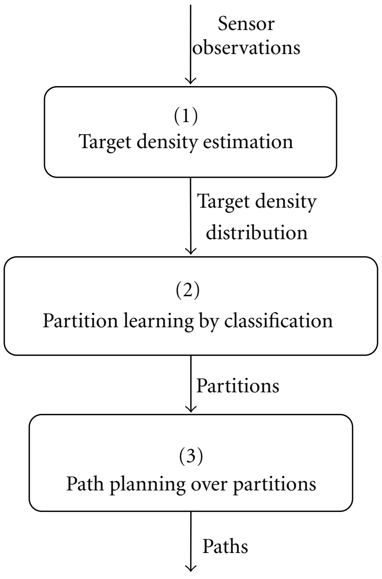
Sensor observations
(1) Target density estimation
Target density distribution
(2) Partition learning by classification
Partitions
(3) Path planning over partitions
Paths

Figure 1: Problem structure of new approach for autonomous search and tracking. The problem is decomposed into three steps. Each of these steps is performed at each instance in time. As the target estimation changes with time, so do the partitions and the agent paths.

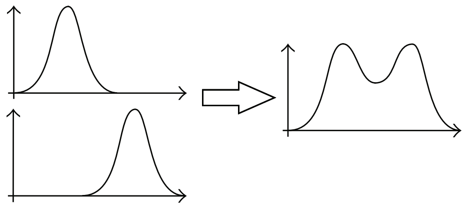

Figure 2: Depiction of capturing the complexity of the high-dimensional target track estimation space and mapping it to the surveillance area. This forms the target density distribution.

3. Step 1: Target Density Estimation

The first step in the problem decomposition is target density estimation [18]. Recall that the first step of the standard approach is target track estimation. Target track estimation is not identical to target density estimation. In order to understand the difference, consider a search and tracking application that estimates the position of N targets utilizing the standard approach. The first step of the standard approach is target track estimation. Assume that target positions are within some region S of the plane (R^2). Call this region the surveillance area. The estimation space is then $N \times S$. The dimension of this space is potentially very high. In order to estimate the position of the targets a probability distribution is determined which is defined over this high-dimensional space. The second step of the standard approach is then agent path planning by optimizating over this high-dimensional space. However, from the perspective of agent path planning it would be beneficial to optimize paths over the significantly lower dimensional space S, instead of $N \times S$. This is the purpose of the target density distribution.

The target density distribution captures the complexity of the high-dimensional target track estimation space and maps it to the single planar space of the surveillance area. This process is depicted in Figure 2. This figure shows multiple single-target distributions combining to form a single target density distribution. A sample target density distribution, defined over a planar surveillance area, is provided in Figure 3. In this figure the target density distribution is represented by contour lines. Red lines have high density and blue lines have low density. The details of this distribution's shape are not important. What is important is to note that a target density distribution captures the complexity of combining the high-dimensional target track estimation.

The space of the target density distribution is the surveillance area S. Now consider some subspace $A \subset S$ of the surveillance area. The target density can be defined as a distribution $f(x)$ that is defined over the surveillance area as

$$\mathrm{E}N_A = \int_{x \in A \subset S} f(x) d\mu(x), \tag{1}$$

where N_A is the number of targets in the subspace A and $\mu(x)$ denotes that measure on which the integral is performed. For example, if the space is discretized then the integral represents a summation.

Notice that the target density distribution provides the estimated number of targets within regions of the surveillance area. The expected total number of targets in the surveillance area is then

$$\mathrm{E}N_S = \int_{x \in S} f(x) d\mu(x). \tag{2}$$

The target density distribution can be computed from target track estimation. For example, consider the case when the positions of N targets are estimated independently. Target track estimation then provides a set of target distributions $\{P(X^1), \ldots, P(X^N)\}$. The target density distribution can be computed as $f(x) = \sum_i P(x^i)$. Consequently there is no need to develop new methods for estimating the target density distribution. Instead, the vast body of target track estimation work can be leveraged.

However, it is not necessary to obtain the target density distribution from target track estimation. Instead, it can be estimated directly from sensor observations. One approach [5] that accomplishes this utilizes random set theory [19] to obtain an approximate target density distribution.

4. Step 2: Partition Learning

Recall that the information contained in a target density distribution can be very complex. This is because it combines the information of all target track estimates and maps them onto the surveillance area. To aid in distributing agents across the surveillance area, the complexity of the target density distribution can be used to partition the surveillance area into disjoint regions. For example, recall the sample target density distribution depicted in Figure 3. One possible set of partitioned regions is depicted in Figure 4. Note that in this figure the target density distribution is represented by contour lines and the boundaries of the partitions are represented by thick, straight lines. The particular choice of partitions displayed is of little significance. What is significant is to note that the partitions are determined based on the information content provided by the target density distribution.

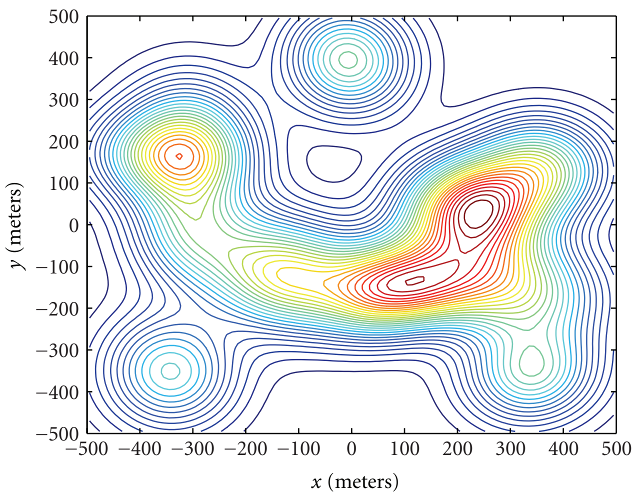

FIGURE 3: Sample target density distribution, represented by contour lines, defined over a surveillance area. The details of this distribution's shape are not important. What is important is to note that a target density distribution captures the complexity of the high-dimensional target track estimation space.

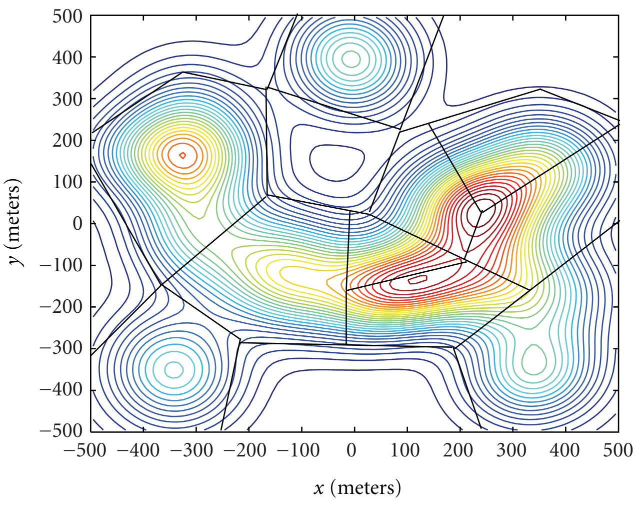

FIGURE 4: Example partitioning of the surveillance area based on a target density distribution. The target density distribution is represented by contour lines. The boundaries of the partitions are represented by thick, straight lines. The space of this plot represents the surveillance area over which the target density distribution is defined. The particular choice of partitions displayed is of little significance. What is significant is to note that the partitions are determined based on the information content provided by the target density distribution.

Instead of partitioning the surveillance area into arbitrary regions the approach taken in this paper is to partition the surveillance area into regions that correspond to

(1) a null target partition,

(2) an exploration partition,

(3) a set of search and tracking partitions.

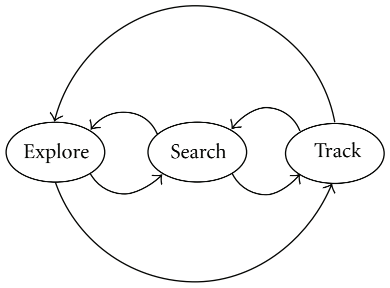

FIGURE 5: The path planning modes that may exist when surveillance is performed by region-based planning.

By partitioning the surveillance area into these types of regions it gives agent path planning algorithms the flexibility to switch between modes of surveillance exploration, target search, and target tracking. These modes are depicted in Figure 5. In order to compute each of these partitions a different classifier is designed. These partition classifiers along with corresponding modes will be described further below.

The combination of these partition classifiers constructs the overall partition learning classifier. The structure of this classifier is presented in Figure 6. The general steps are to classify (1) the null target partition, (2) the exploration partition, and (3) a set of search and tracking partitions.

Before describing the details of these steps, the computational flow of the overall classifier can be understood by considering a set Γ and how it changes as it moves through the classifier. Let Γ be the set of all points in the surveillance area. As such, the target density distribution is defined over Γ. The flow of the classifier can then be understood as

(1) the null target partition S_{null} is classified and removed from Γ ($\Gamma \leftarrow \Gamma \setminus S_{\text{null}}$) in block (1) of Figure 6,

(2) Γ is now a subset of the surveillance area. Within Γ, the exploration partition S_{explore} is classified and then removed from Γ ($\Gamma \leftarrow \Gamma \setminus S_{\text{explore}}$) in block (2) of Figure 6,

(3) Γ now consists of only the subset of the surveillance area that will be partitioned into search and tracking partitions. In block (3) of Figure 6 an ordered set of search and tracking partitions are classified within Γ.

Each of these partition learning classifier steps will now be discussed in more detail.

4.1. Partitioning Step 1: Null Target Partition. Over time some regions will be repeatedly observed. Much of the observed regions will never have targets detected. Dependent on anticipated possible target mobility, it may be concluded that no targets exist in these areas. It is necessary to maintain a partition that classifies regions in which no targets exist. These regions form the null target partition.

The first step of partition learning is to classify this null target partition. The importance of this partition is seen by considering two scenarios. The first is when there is no

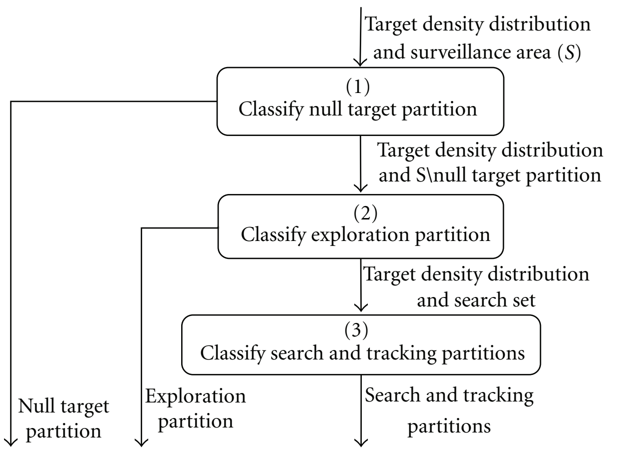
Target density distribution and surveillance area (S)
(1) Classify null target partition
Target density distribution and S\null target partition
(2) Classify exploration partition
Target density distribution and search set
(3) Classify search and tracking partitions
Null target partition
Exploration partition
Search and tracking partitions

FIGURE 6: Cascade of classifiers that partition the surveillance area into (1) a null target partition, (2) an exploration partition, and (3) a set of search and tracking partitions. Note that at each instance of time that the target density distribution changes, this classifier is processed on the new target density distribution. In this sense partitions change with time.

target in the surveillance area. At some point in time the conclusion should be reached that there is no target. The second scenario is when there is a vast exploration partition. As regions become fully observed, but no targets have been detected, these observed regions should cease to be explored.

The method of classification for this step of partition learning will now be described. To do this the features used for classification will be described first. Then the classifier will be described.

4.1.1. Features. Low values for target density are what define the null target partition. The only feature required to classify this partition is then simply values of the target density distribution. The target density distribution was defined previously in (1).

4.1.2. Classifier. Recall that Γ represents the set over which the classifier operates. As such, Γ is initially the entire surveillance area ($\Gamma \leftarrow S$). The first step (block (1) of Figure 6) of partition learning is to determine the null target partition S_{null} and remove it from Γ ($\Gamma \leftarrow \Gamma \setminus S_{\text{null}}$). This step of the classifier is visualized in Figure 7 for a simple one-dimensional target density distribution.

To determine which points in Γ belong to the null target partition, a target nullity threshold ϵ_{null} is required. This threshold specifies the value of target density below which it is assumed no targets exist. With this target nullity threshold given, all points in the surveillance area that correspond to regions of essentially no targets can then be defined by

$$S_{\text{null}} := \{x \in S : f(x) < \epsilon_{\text{null}}\}. \quad (3)$$

The set of points in S_{null} then form the null target partition. This set of points is removed from Γ and the classifier continues by classifying the exploration partition.

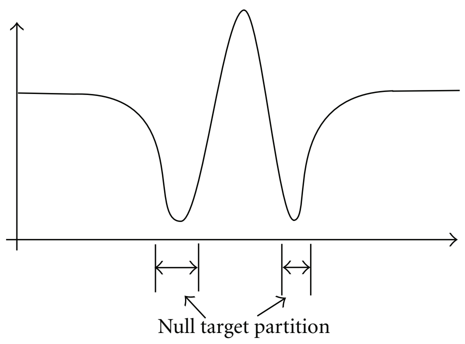

Figure 7: Visualization of a simple target density distribution with corresponding null target partition noted.

4.2. Partitioning Step 2: Exploration Partition. The second step of partition learning is to classify the exploration partition. The exploration partition consists of areas within the surveillance area that have low-to-no information bias. For example, a region over which there is defined a uniform probability distribution would be included in the exploration partition. Because there is no bias in information, an exploration-oriented mode of path planning may be preferred for these regions [20, 21]. To allow this exploration-oriented mode of path planning, these types of regions are classified into a separate partition.

The method of classification for this step of partition learning will now be described. To do this the features used for classification will be described first. Then the classifier will be described.

4.2.1. Features. In order to classify points in Γ into the exploration partition, two features are required. These features are

(1) local uncertainty,

(2) target density.

Local uncertainty is used to determine regions of locally uniform value. Initially it may appear that only local uncertainty is required to define the exploration partition completely. However, there is a subtle aspect that requires the addition of target density in order to completely capture the entire exploration partition.

This subtle aspect can be understood by considering the case when the entire surveillance area is initially uniformly distributed. An agent makes imperfect no-detection observations. As such, some regions will have low-target density (regions that have been observed well), completely unobserved regions will have an unchanged uniform value, and others will have value somewhere in between the low value and the unchanged uniform value (due to poor observation in these areas). These in-between-valued areas will not have locally uniform value, yet will still belong to the exploration partition. This case suggests that target density, in addition to local uncertainty, is required to catch the complete exploration partition.

Before defining local uncertainty a definition for local area is required. The local area $S_r(x_0) \subseteq S$ of some point $x_0 \in S$ in the surveillance area (where S is the space of the surveillance area) is defined as

$$S_r(x_0) := \{x \in S : d(x, x_0) < r\}, \tag{4}$$

where $d(x, x_0)$ is some measure of distance.

Local uncertainty is defined by first selecting a measure for uncertainty. In this paper local uncertainty is based on entropy. As such, local uncertainty is computed by evaluating the local entropy defined as

$$\mathrm{H}_r(x_0) := \int_{x \in S_r(x_0)} f_r(x, x_0) \log\left(\frac{1}{f_r(x, x_0)}\right) d\mu(x), \tag{5}$$

where $f_r(x, x_0)$ is the locally normalized target density function defined by

$$f_r(x, x_0) := \frac{f(x)}{\int_{y \in S_r(x_0)} f(y)\, d\mu(y)}. \tag{6}$$

Note, if local uncertainty were not defined on a locally normalized target density distribution there would not be a well-established maximum value for local uncertainty. A locally normalized density is then required so that the maximum value for local uncertainty can be referenced during classification. This maximum value allows the classifier to determine if a particular region contains significantly biased information of target density. To aid in understanding local uncertainty Figure 8 presents the computation of local entropy over a surveillance area when the underlying target density distribution is a simple Gaussian probability distribution centered in the middle of the surveillance area. In this figure, local uncertainty is represented by shade value where white is high value and black is low value.

4.2.2. Classifier. At this point of the classifier Γ consists of a subset of the surveillance area defined by $\Gamma := S \setminus S_{\text{null}}$. In this step (block (2) of Figure 6) of partition learning the exploration partition S_{explore} is classified and removed from Γ ($\Gamma \leftarrow \Gamma \setminus S_{\text{explore}}$). Define this resulting state of Γ to be the search partition (or search set) S_{search}. The process of this stage of the classifier is visualized in Figure 9 for a simple one-dimensional target density distribution.

To determine which regions in Γ are approximately locally uniform, a new set S_{HE}, called the high-entropy set, is computed. The high-entropy set is defined as

$$S_{\text{HE}} := \{x \in \Gamma : |\mathrm{H}_r(x) - \mathrm{H}_{\max}| < \epsilon\}, \tag{7}$$

where $\mathrm{H}_r(x)$ is the local entropy feature as defined in (5) and $\mathrm{H}_{\max}$ is the maximum local entropy possible defined as

$$\mathrm{H}_{\max} := \log|S_r|, \tag{8}$$

where $|S_r| := \int_{x \in S_r} d\mu(x)$. Take, for example, the case when the target density distribution is discretized on a fixed grid defined over the surveillance area. Recall the definition of S_r in (4). In (4), note that in order to define S_r it is required to define a measure of distance $d(x, x_0)$ between two points x

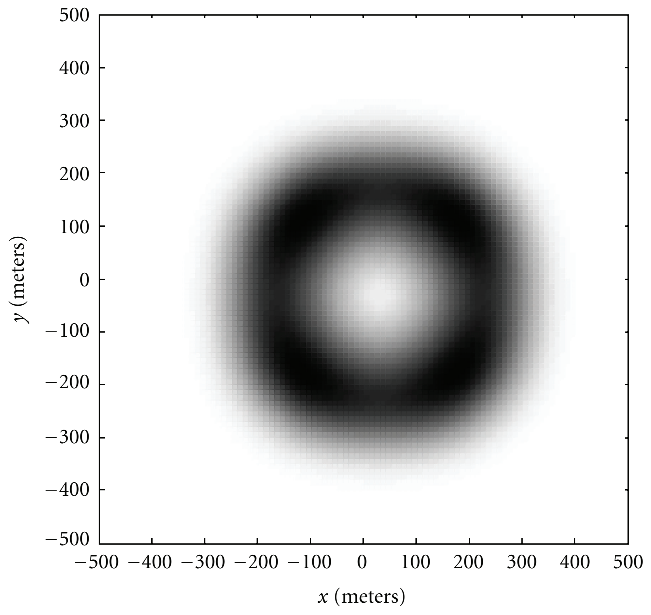

Figure 8: A sample computation of local uncertainty over a surveillance area when the underlying target density distribution is a simple Gaussian probability distribution. Local uncertainty is represented by shade value where white is high-local uncertainty and black is low-local uncertainty. In this sample, notice that the peak and the tails of the Gaussian have high-local uncertainty whereas the regions in between the peak and tails have low uncertainty.

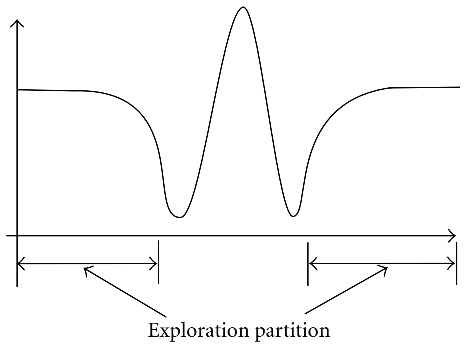

Figure 9: Visualization of a simple target density distribution with corresponding exploration partition noted.

and x_0 in the surveillance area. For a fixed grid, this distance is defined as

$$d(x, x_0) := \max(\{|x(1) - x_0(1)|, |x(2) - x_0(2)|\}), \quad (9)$$

where the numbers 1 and 2 specify the indices of the points. Then, according to the definition of S_r, the maximum local entropy is $H_{\max} = \log N^2$, where N is the number of rows/columns in the square local area S_r.

To catch regions for which the subtlety mentioned above applies another set is computed. This set is called the low-density set S_{NI}. The definition of this set is simple, however, it requires explanation. At the beginning of a surveillance task an initial target density (or prior distribution) is constructed that expresses prior belief in possible target locations. At a minimum this prior consists of two pieces. These pieces are

(1) a prior distribution $f_{\text{prior}}(x)$ of previously known target positions,

(2) an estimated number of additional targets $EN_{\text{additional}}$ that may exist in the surveillance area.

The prior target density distribution $f_{\text{prior}}(x)$ provides a target density bias based on where targets have most recently been observed and where they might be now. For example, f_{prior} may consist of a summation of Gaussian distributions with each Gaussian representing the possible location of a particular target whose position was once known or whose position is simply guessed. The estimated number of additional targets $EN_{\text{additional}}$ affects the initial target density distribution by defining a uniform target density distribution $U(x)$ such that

$$\int_{x \in S} U(x) d\mu(x) = \frac{EN_{\text{additional}}}{\int_{x \in S} d\mu(x)}. \quad (10)$$

The resultant prior target density distribution is then

$$f(x) = f_{\text{prior}}(x) + U(x). \quad (11)$$

Note that some regions of the prior target density distribution will have a characteristic uniform value U. This characteristic low-information value can then be used to define the low-density set. The regions that must be captured in the low-density set are those regions that have target density in between the locally uniform density and the null target threshold. Yet, because $\Gamma = S \setminus S_{\text{null}}$ at this point of the classifier, the low-density set can be defined simply as

$$S_{NI} := \{x \in \Gamma : f(x) < U + \epsilon\}. \quad (12)$$

Then, combining the low-density set with the high-entropy set, the exploration partition is defined as

$$S_{\text{explore}} := S_{\text{HE}} \cap S_{NI}. \quad (13)$$

S_{explore} can then be removed from Γ as $\Gamma \leftarrow \Gamma \setminus S_{\text{explore}}$. Let this state of Γ be called the search partition (or search set) S_{search}. S_{search} is then defined as

$$S_{\text{search}} := S \setminus \left\{S_{\text{null}} \cup S_{\text{explore}}\right\}. \quad (14)$$

After removing the exploration partition from Γ the classifier then continues on to classifying the search and tracking partitions.

4.3. Partitioning Step 3: Search and Tracking Partitions.

The third step of partition learning is to classify a set of search and tracking partitions. Both search and tracking partitions are classified by the same classifier.

In terms of information content, the opposite type of region to the exploration partition is a tracking partition. Tracking partitions are small spatially and are partitions in which there is strong bias of target density and high certainty.

These are partitions in which targets have been detected consistently. Consequently, tracking partitions define locations of known targets. These partitions must continue to be tracked according to the mobility of the tracked targets. The search strategy then becomes that of keeping observance of the known position of the targets. For example, the points of maximum density within tracking partitions are kept in observance. The search strategy for tracking partitions is then the most constraining on agent motion. For example, if an agent is a fixed-wing aircraft it will have to fly orbit-like paths encircling the known position of the target [14, 22–25].

Similar to tracking partitions are search partitions. The similarity that the search partitions have with the tracking partitions is that both consist of an information set that provides a bias to aid in optimizing search plans. However, search partitions are different from tracking partitions in that the information content is not very certain. Consequently, not much can be said about exactly where a target may be located. However, there is bias over which parts of the regions have high possibility of target existence. It then becomes the duty of a search plan to optimize paths based on the information content in order to yield a new distribution with higher certainty. The search strategy then becomes to maximize some type of information gain, and a searcher's paths are guided to improve the information content in order to ultimately observe a target [6–8, 12, 26, 27].

The method of classification for this step of partition learning will now be described. To do this the features used for classification will be described first. Then the classifier will be described.

4.3.1. Features. In order to classify points in Γ into search and tracking partitions, two features are required. These features are

(1) normalized position within the surveillance area,

(2) local expected number of targets.

Normalized position is computed for some point $x \in S$ by dividing by the size of the surveillance area S. Local expected number of targets comes readily from the target density distribution. To understand this, recall the definition of the target density distribution. From this definition it is apparent that the expected number of targets within some region A is just the integration of the target density distribution over that region. Recalling the definition of local area, local expected number of targets is then computed by integrating the target density distribution over the local area as

$$\begin{aligned} \mathrm{I}_r(x_0) &:= \mathrm{E}N_{S_r(x_0)} \\ &= \int_{x\in S_r(x_0)} f(x)d\mu(x). \end{aligned} \tag{15}$$

To aid understanding of local expected number of targets, Figure 10 presents a sample computation over a surveillance area when the target density distribution is a simple Gaussian probability distribution. In this figure the value of local expected number of targets is represented by shade value, white being high and black being low. From this figure it can be observed that local expected number of targets acts as a smoothing filter over the target density distribution.

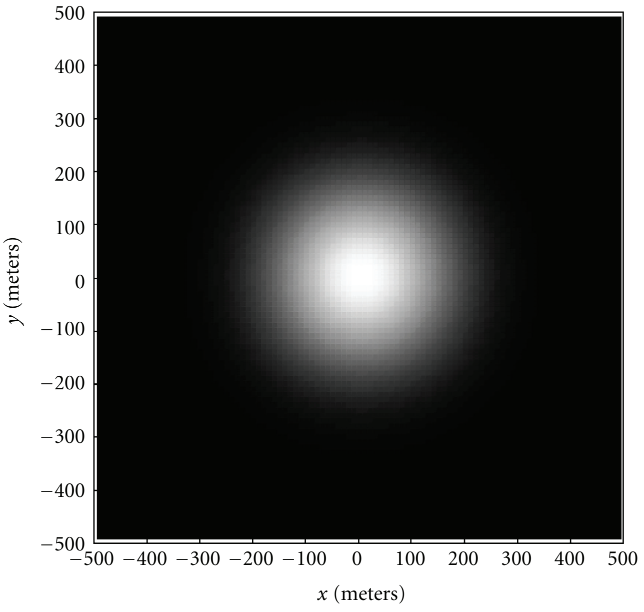

Figure 10: A sample computation of local expected number of targets I_r over a surveillance area when the underlying target density distribution is a simple Gaussian probability distribution. Notice that local expected number of targets acts as a smoothing filter over the target density distribution.

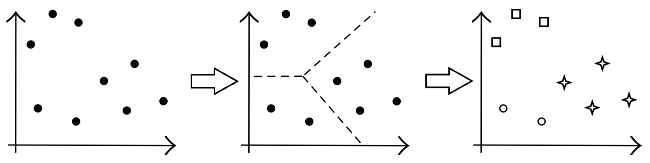

Figure 11: Visualization of how points in the target density distribution feature space are convexly partitioned. In this example three partitions are classified. The resulting three partitions are represented by circles, stars, and rectangles.

4.3.2. Classifier. At this point in partition learning (block 3 in Figure 6) the set on which classification operates is $\Gamma = S_{\text{search}} = S \setminus \{S_{\text{null}} \cup S_{\text{explore}}\}$. Γ then consists of regions in the surveillance area with some level of biased information of possible target locations. In this step of partition learning Γ is partitioned into a set of search and tracking partitions. This step cannot be performed by a simple set computation as was done for the previous two steps of partition learning. Instead, points in Γ are clustered according to the target density feature space. In order to help visualize the action that occurs at this stage of the classifier, observe Figure 11. This figure shows how a set of points in the target density distribution feature space are convexly partitioned.

There are many convex clustering methods [28] that would work for this level of the classifier. The approach taken in this paper is to perform classification by utilizing both K-means and Gaussian Mixture Model EM [28]. K-means is used to initialize search and tracking partitioning at the beginning of the surveillance task. Gaussian Mixture Model EM is then used at subsequent steps in time, where the Gaussian Mixture Model EM is seeded with previous partition means [17]. Both K-means and Gaussian Mixture Model EM operate in the target density distribution feature space consisting of normalized position within the surveillance area and local expected number of targets.

In order to perform this classification, the number of partitions to classify must be initialized. The initial number of partitions is determined from the total expected number of targets in the surveillance area

$$\begin{aligned} N_{\text{initial}} &:= \text{ceil}(\mathrm{E}N_S) \\ &= \text{ceil}\left(\int_{x \in S} f(x) d\mu(x)\right). \end{aligned} \tag{16}$$

After performing K-means or Gaussian Mixture Model EM to convexly partition $\Gamma = S_{\text{search}}$ into a set of search and tracking partitions, these partitions are then ordered. The partitions are ordered so that tracking partitions appear first and uncertain searching partitions appear last. This enables path planning algorithms to prioritize the various search and tracking partitions. To accomplish this ordering the partition density ρ_{P_i} of each partition P_i is computed. Partition density is defined as

$$\rho_{P_i} := \frac{\int_{x \in P_i} f(x) d\mu(x)}{\int_{x \in P_i} d\mu(x)}. \tag{17}$$

At this point in this step of partition learning an ordered set of search and tracking partitions have been classified. Some of these partitions may correspond to regions with many densely located targets. It may be beneficial to allocate more searching resource to these types of partitions. Accordingly, these partitions are further subpartitioned.

In order to determine if some partition P_i should be subpartitioned its expected number of targets $\mathrm{E}N_{P_i}$ is computed. $\mathrm{E}N_{P_i}$ is easily computed from the target density distribution $f(x)$ as

$$\mathrm{E}N_{P_i} := \int_{x \in P_i} f(x) d\mu(x). \tag{18}$$

If $\mathrm{E}N_{P_i} > 1$ then it is expected that there is more than one target within P_i. In order to track all of these possible targets multiple agents may be required. To account for the possible need of multiple agents, any P_i with $\mathrm{E}N_{P_i} > 1$ is subpartitioned into $\text{ceil}(\mathrm{E}N_{P_i})$ new partitions. And this is where the classifier ends. The end result is one null target partition, one exploration partition, and a set of ordered search and tracking partitions.

5. Step 3: Path Planning over Partitions

The final step of the approach presented in this paper for autonomous search and tracking is path planning. This path planning is performed over the set of partitions. In order to plan paths over partitions, path planning is decomposed into two steps as depicted in Figure 12. These steps are

(1) partition task allocation,

(2) target density distribution based path optimization.

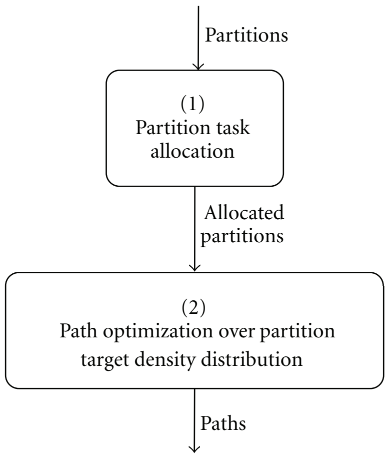
Partitions
(1) Partition task allocation
Allocated partitions
(2) Path optimization over partition target density distribution
Paths

Figure 12: Structure of path planning decomposed into task allocation over partitions and path planning by optimizing directly over the target density distribution.

In the first step partitions are allocated to the team of agents [29, 30]. In the second step agent paths are determined within allocated partitions by optimizing directly over partition level target density distributions [6–8, 10, 14].

By decomposing path planning in this manner the vast amount of work that has been developed for vehicle routing (for partition task allocation) and receding horizon path optimization (for target density-distribution-based path optimization) can be leveraged. Now, all that is required is extensions of existing methods where necessary. As such, this section refers the reader to the body of work that is leveraged and then presents any required extensions.

5.1. Path Planning Step 1: Partition Task Allocation. The partitions generated by the classifier define areas over which subsets of the target density distribution can be extracted. This suggests the application of some kind of task allocation algorithm that takes each of the partitioned search areas as tasks with varying level of certainty or priority. The exact method of task allocation is beyond the scope of this paper since it has been well developed by researchers already. Refer specifically to [29, 30] for methods that directly apply. For task allocation algorithms that have been developed for projects at the Center for Collaborative Control of Unmanned Vehicles refer to [29].

5.2. Path Planning Step 2: Distribution-Based Optimization. Optimizing a path over a target density distribution is almost identical to optimizing a path over a probability distribution. Fortunately, much work has been done to develop path optimization over probability distributions [6–8, 10, 14]. These existing methods are leveraged in this paper. The general approach of these methods is to define a function for measuring the utility of a path based on an underlying probability distribution. Then, this utility function is used to optimize paths through the surveillance area. These methods are extended by defining a utility function based on target density distributions. Defining this utility function is the focus of this section.

In order to define a path's utility the utility of a point in the surveillance area must be defined. However, before defining the utility of a point, an agent's sensor observation coverage $f_C(x, x_0)$, about a point x_0 in the surveillance area, must be determined. $f_C(x, x_0)$ essentially specifies how applicable some point x in the surveillance area is to a particular agent when the agent is located at x_0. For example, consider the case of a fixed sensor. This sensor is free to rotate in order to observe its surroundings. However, it cannot see beyond r meters. Consequently, any point farther away than r meters is of little significance to this sensor.

An agent's observation coverage is determined by the properties of the agent's sensor. For example, if an agent can make observations perfectly within a radius r, then the observation coverage is an indicator function defined by

$$f_C(x, x_0) = \begin{cases} 1, & \text{if } \|x - x_0\| < r, \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

However, in general, the observation coverage is determined by the sensor's capable field of view as well as the resolution of observable points within the field of view and the probability of missed detection [31]. In order to further visualize possible sensor observation coverages, consider two cases.

(1) An agent can view it's surroundings perfectly within 25 meters. Beyond that, the agent's view linearly degrades until it cannot make any observation at 70 meters.

(2) An agent cannot view anything near it until a distance of 45 meters away. After that, observations quickly become perfect but then start to fade around 65 meters. By 90 meters, observations are no longer possible.

The first of these cases is similar to what is true for many sensing agents [32]. They are designed such that their observations improve with proximity. Figure 13 depicts this sensor coverage. In this figure the quality of a point's coverage by the agent's sensor is represented by a shaded value where white is high utility and black is low utility.

The second of these cases may seem odd, but is actually similar to what was used in an experiment performed with autonomous aircraft equipped with visual spectrum cameras [33]. In this experiment, a camera on-board an aircraft was zoomed in to detect features of a pedestrian. The zoom was designed so that good resolution would be provided when the aircraft orbited the pedestrian. Consequently, it was designed to make good observations at an orbit's radius away from the aircraft. Figure 14 depicts this sensor coverage. In this figure the quality of a point's coverage by the agent's sensor is represented by a shaded value where white is high utility and black is low utility.

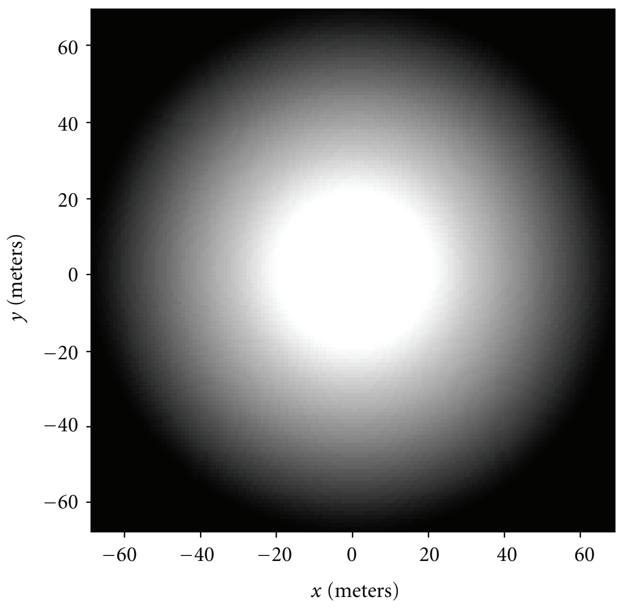

Figure 13: A sample sensor observation coverage where the quality of coverage is represented by shaded value, white being high quality and black being low quality. This type of coverage is applicable when a sensor's observations are improved with close proximity.

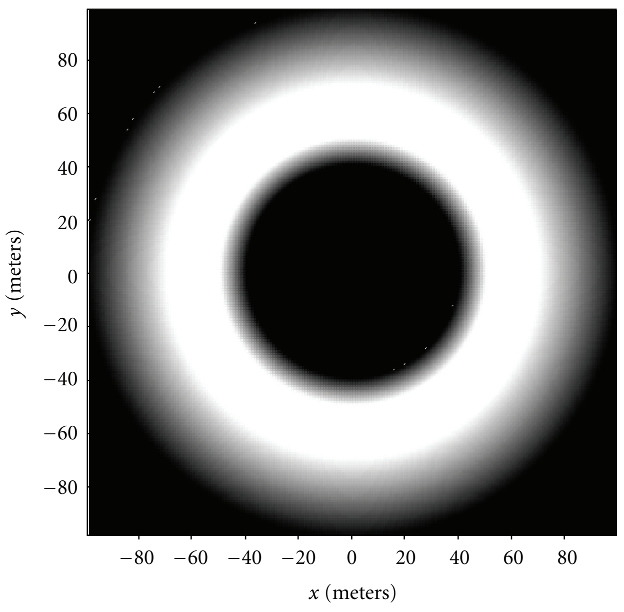

Figure 14: A sample sensor observation coverage where the quality of coverage is represented by shaded value, white being high quality and black being low quality. This type of coverage is applicable when a sensor makes good observations at some specified distance away.

The utility of an agent's point x_0 in the surveillance area can then be defined utilizing sensor coverage. First, consider a zero horizon path. The utility of a point x_0 is

$$V_0(x_0) := \int_{x \in S} f(x) f_C(x, x_0) d\mu(x), \tag{20}$$

where $f(x)$ is the target density distribution. Extending this to finite horizon planning, define the H-step horizon observation coverage over the path $x_{0:H} = (x_0, \ldots, x_H)$ as

$$f_C(x, x_{0:H}) := 1 - \prod_{t=0:H} (1 - f_C(x, x_t)). \tag{21}$$

The utility of a point $x_0 \in S$, and consequently the path $x_{0:H}$, is then defined as

$$V_H(x_0) := \max_{\substack{x_i \in \mathrm{R}(x_{i-1}) \\ i=1,\ldots,H}} \int_{x \in S} f(x) f_C(x, x_{0:H}) d\mu(x), \tag{22}$$

where $\mathrm{R}(x)$ is the set of all points within the reach set of x [14]. Intuitively, (22) represents the expected number of targets within the sensor coverage over a H-step path originating from the point x_0. Maximizing (22) then corresponds to choosing the point $x_0^\star$ that yields the maximum expected number of targets within the observation coverage of a path originating from $x_0^\star$.

This definition of path utility was based on the entire target density distribution. In order to optimize paths through specific partitions the utility must be defined for partition level target density distributions. Fortunately, this extension is easily accomplished. First, let the set of partitions defined over the surveillance area be $P = \{P_1, \ldots, P_n\}$. The partition level target density distribution $f_{P_i}(x)$ for partition $P_i \in P$ is then defined as

$$f_{P_i}(x) := \begin{cases} f(x) & \text{if } x \in P_i, \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

And then replacing $f(x)$ with $f_{P_i}(x)$, the partition level utility of a path starting at x_0 is then defined as

$$V_H^{P_i}(x_0) := \max_{\substack{x_i \in \mathrm{R}(x_{i-1}) \\ i=1,\ldots,H}} \int_{x \in S} f_{P_i}(x) f_C(x, x_{0:H}) d\mu(x). \tag{24}$$

This equation fully specifies partition level target density-distribution-based path optimization. Further development and application-specific details can be found in [14, 17].

6. Results

In this paper a new approach for autonomous search and tracking was presented. This new approach was designed for the case when the surveillance area is large, the number of targets is unknown, and target estimation is general and nonparametric. In this section, results of the performance of this approach are presented.

The performance of partition learning to aid in agent path planning was tested by constructing a simulation environment. In this environment the team of agents consisted of autonomous aircraft equipped with visual spectrum gimballed camera sensors. The capabilities of these agents were designed to closely represent behaviors observed in flight experiments [32, 33]. The camera characteristics were designed to represent a field of view resulting from a 0.9273 rad view angle. Effects of resolution were included by limiting the distance of observations to 250 m. The agents were designed to fly at 25 m/s and 100 m altitude with a

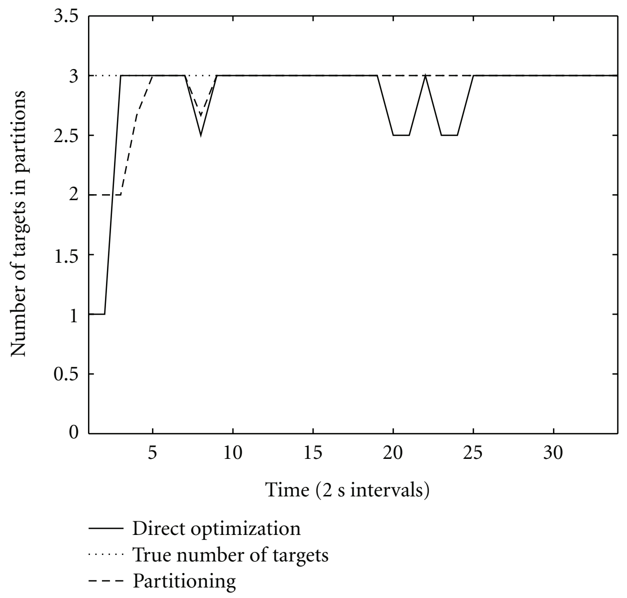

Figure 15: Comparison of sample mean number of targets within search or tracking partitions over simulation time between state-of-the-art direct distribution optimization and partition learning classification path planning approaches.

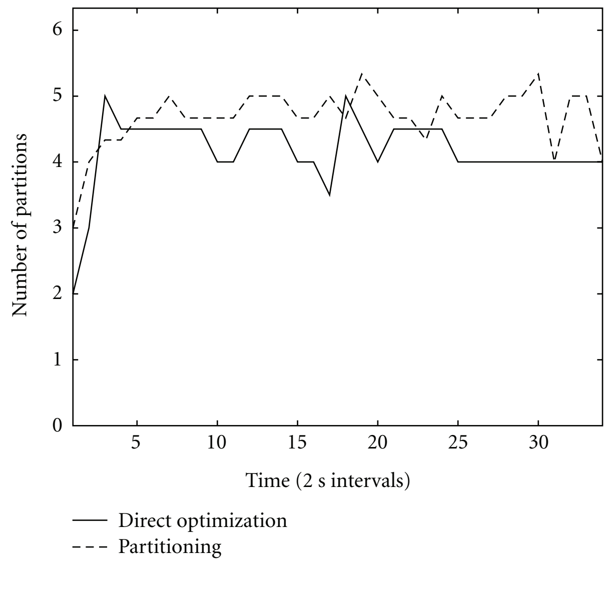

Figure 16: Comparison of sample mean number of search and tracking partitions over simulation time between state-of-the-art direct distribution optimization and partition learning classification path planning approaches.

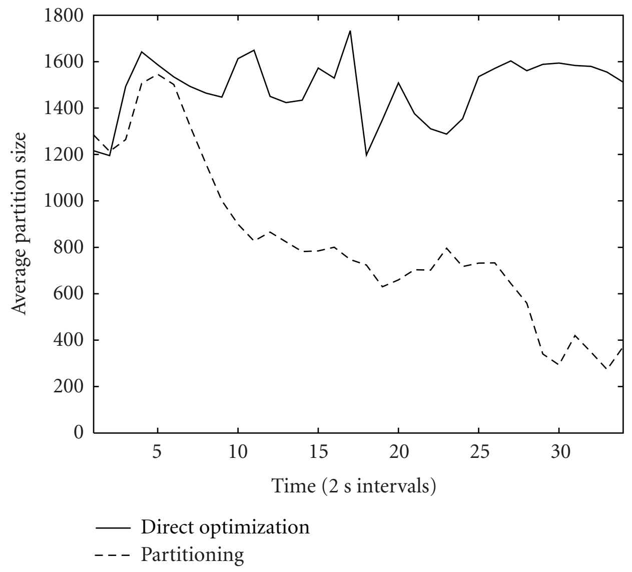

Figure 17: Comparison of sample mean average search and tracking partition size over simulation time between state-of-the-art direct distribution optimization and partition learning classification path planning approaches.

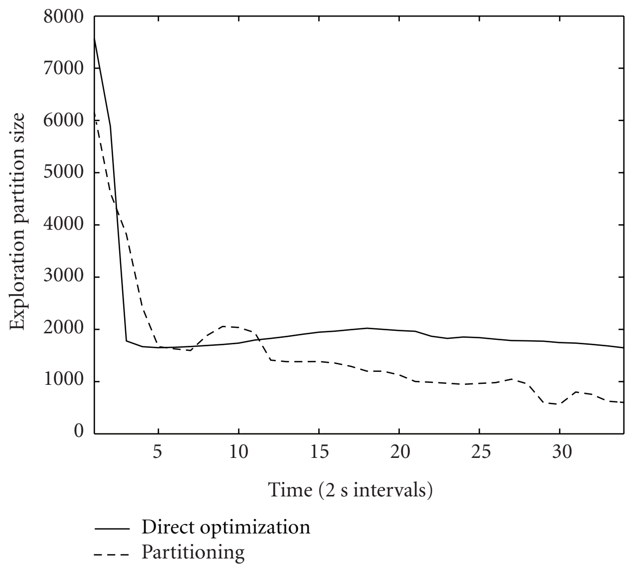

Figure 19: Comparison of sample mean exploration partition size over simulation time between state-of-the-art direct distribution optimization and partition learning classification path planning approaches.

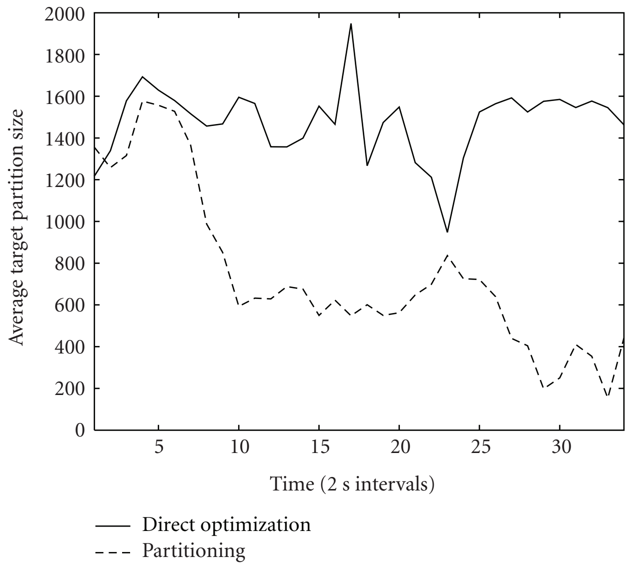

Figure 18: Comparison of sample mean average partition size of partitions containing targets over simulation time between state-of-the-art direct distribution optimization and partition learning classification path planning approaches.

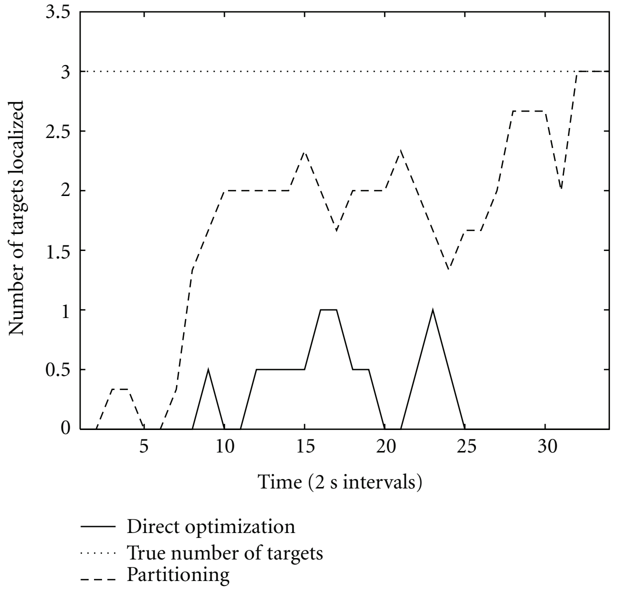

Figure 20: Comparison of sample mean number of targets localized over simulation time between state-of-the-art direct distribution optimization and partition learning classification path planning approaches.

maximum turn rate of 0.2 rad/s. The targets were allowed to move according to a transition model defined by

$$x_{T,t} = x_{T,t-1} + r\begin{bmatrix}\cos(\theta)\\ \sin(\theta)\end{bmatrix}, \tag{25}$$

where r and θ were distributed as

$$\begin{aligned} r &\sim \text{Gaussian } (\mu, \sigma^2), \\ \theta &\sim \text{Uniform } ([0, 2\pi)), \end{aligned} \tag{26}$$

with $\mu = 2\,\text{m}$ in one second and $\sigma^2 = 10\,\text{m}^2$. The time interval of each simulation iteration was 4 seconds. Several simulation samples were performed with various initial target and agent positions as well as various prior density distributions.

The performance of partition classification is affected by the quality and diversity of observations made over the surveillance area. Partition classification should perform well according to the observations it receives. To measure this performance several metrics were used. These metrics are

(1) number of targets in search and tracking partitions,

(2) number of search and tracking partitions,

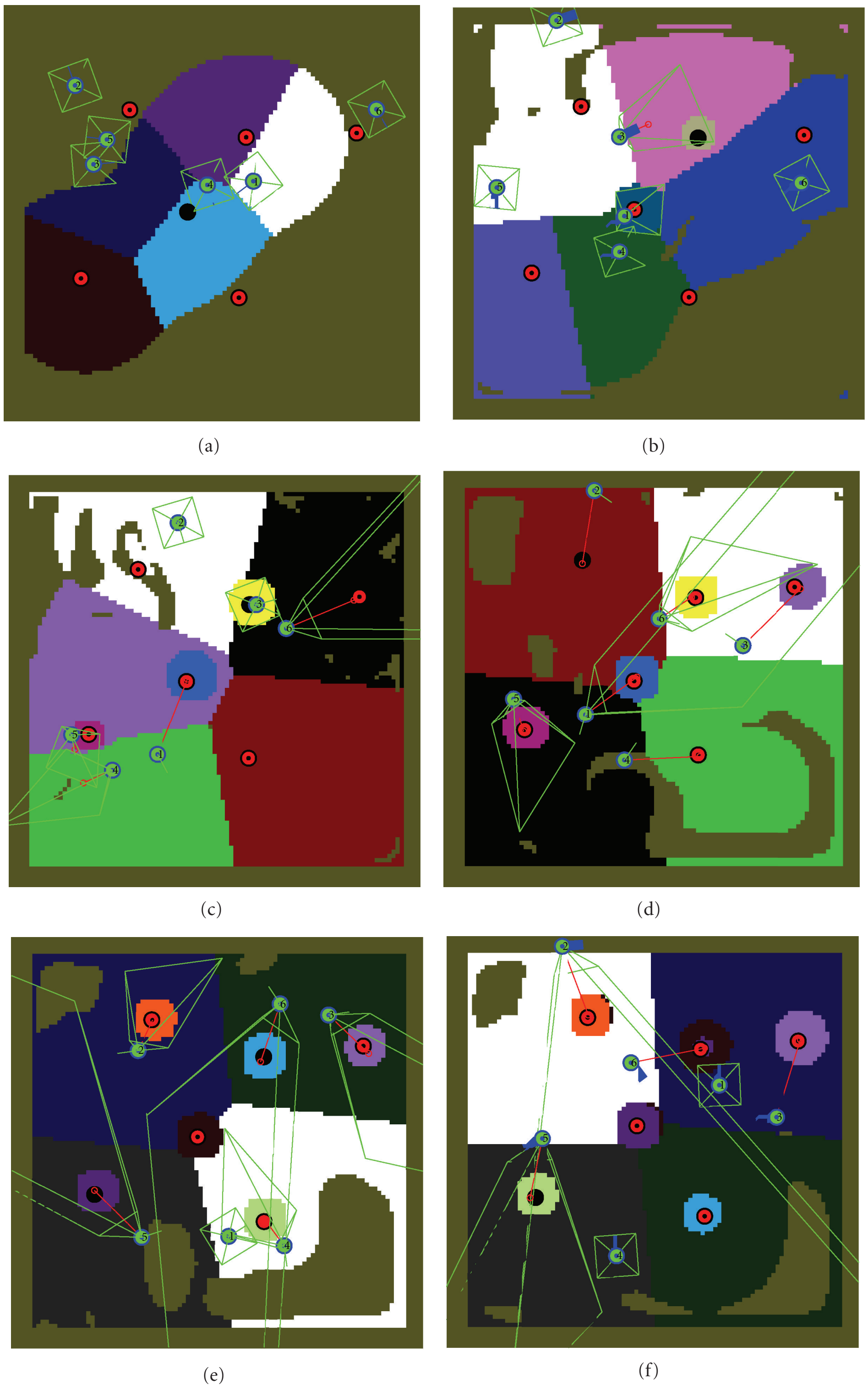

Figure 21: Sample sequence of partition learning classification path planning for a scenario involving six agents and six targets (the number of targets was unknown to the agents). Agents are represented by blue/green circles with protruding lines that represent the direction of the agents. The approximate field of view of each agent is represented by green connected lines. Targets are represented by black/red circles. The entire surveillance area is partitioned. These partitions are represented by different colors. Follow the sequence from left to right and from top to bottom. In (a) initial partitions are formed. In (b) tracking partitions appear. By (f) all targets are within tracking partitions.

(3) average search and tracking partition size,

(4) average size of search and tracking partitions containing targets,

(5) exploration partition size,

(6) number of targets localized.

It is additionally necessary to provide a comparison in order to see how well the presented methods perform. Recall the standard approach for autonomous search and tracking. The standard approach optimizes agent paths directly over the distribution. The comparison provided in this section is then between using partition learning to aid path planning versus optimizing paths directly over the target distribution. This standard approach will be referred to as the state-of-the-art. Note, however, in order to compute some of the metrics above, it is necessary to run the partition classification algorithm for both cases. The partition learning classifier was run for both path planning approaches (that presented in this paper and the state-of-the-art). Yet, only the path planning approach presented in this paper utilized the partitions for path planning purposes.

In this paper, results for the scenario in which there are six agents and three targets is provided. Additionally, a sample sequence of partition learning is provided as a visual aid to understand how these partitions may look for the case when there are six agents and six targets. This sample sequence is found in Figure 21. This figure spans an entire page so it is provided after all other figures. Additional scenarios are provided in [17].

The results provided in this section demonstrate the scenario when there are sufficient resources to perform surveillance. From these results it is concluded that the the approach presented in this paper performed well. This conclusion is determined by observing Figures 15 and 16. Figure 15 presents the number of targets in search or tracking partitions over time. Figure 16 presents the number of search or tracking partitions over time.

From Figures 15 and 16 it is apparent that all targets are quickly captured within search or tracking partitions and the number of partitions is bounded. However, the performance of the two path planning approaches is very different. From Figure 17, it can be seen that the average partition size does not decrease for state-of-the-art path planning. However, the average partition size decreases substantially for partition learning classification path planning.

A similar result is also true for the average size of partitions containing targets, plotted in Figure 18. Additionally, according to Figure 19, the exploration size continually decreases for partition learning classification, but tends to level off for state-of-the-art path planning. Furthermore, the state-of-the-art path planning did not perform well to localize targets in this scenario. In contrast to this, partition learning classification path planning performed well to eventually localize all targets. This can be seen in Figure 20. From the results presented here, it is then apparent that partition learning classification path planning performs well to find and localize targets for this scenario, as compared to state-of-the-art path planning.

7. Conclusions

In this paper a new approach for autonomous search and tracking was presented. This new approach was designed for the case when the geographic area is large, the number of targets is unknown, and target track estimation is general and nonparametric. This is a challenging problem because very little is assumed. All that was assumed is that some form of target track estimation is available and shared among the team of autonomous agents performing the search. This new approach decomposes the search and tracking problem into three steps. The first step is target density distribution estimation. The second step is partition learning classification based on the target density distribution. The third step is path planning based on the partitions.

The vast body of work available for target track estimation and path planning over probability distributions was leveraged to provide solutions for the first and third steps. As such, the main focus of this paper was on partition learning. In order to determine the performance of this new approach, it was compared with the standard approach of directly optimizing paths over target estimation distributions. From this comparison, it is concluded that the approach presented in this paper performs well and provides an improved solution for this very general form of the autonomous search and tracking problem.

References

[1] Y. Bar-Shalom, *Tracking and Data Association*, Academic Press, San Diego, Calif, USA, 1987.

[2] Y. Bar-Shalom and X. R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*, YBS, Urbana, Ill, USA, 1995.

[3] S. S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, Norwood, Mass, USA, 1999.

[4] L. Stone, *Theory of Optimal Search*, Academic Press, New York, NY, USA, 1975.

[5] R. P. S. Mahler, *Statistical Multisource-Multitarget Information Fusion*, Artech House, Norwood, Mass, USA, 2007.

[6] F. Bourgault, T. Furukawa, and H. Durrant-Whyte, "Optimal search fora lost target in a bayesian world," in *Field and Service Robotics*, S. Yuta, H. Asama, E. Prassler, T. Tsubouchi, and S. Thrun, Eds., vol. 24 of *Springer Tracts in Advanced Robotics*, pp. 209–222, Springer, Berlin, Germany, 2006.

[7] C. M. Kreucher, A. O. Hero, K. D. Kastella, and M. R. Morelande, "An information-based approach to sensor management in large dynamic networks," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 978–999, 2007.

[8] B. Grocholsky, A. Makarenko, and H. Durrant-Whyte, "Information-theoretic coordinated control of multiple sensor platforms," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '03)*, vol. 1, pp. 1521–1526, September 2003.

[9] G. M. Hoffmann and C. J. Tomlin, "Mobile sensor network control using mutual information methods and particle filters," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 32–47, 2010.

[10] J. Tisdale, Z. W. Kim, and J. K. Hedrick, "Autonomous UAV path planning and estimation: an online path planning framework for cooperative search and localization," *IEEE Robotics and Automation Magazine*, vol. 16, no. 2, pp. 35–42, 2009.

[11] F. Bourgault, T. Furukawa, and H. F. Durrant-Whyte, "Coordinated decentralized search for a lost target in a bayesian world," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 48–53, October 2003.

[12] A. D. Ryan, H. Durrant-Whyte, and J. K. Hedrick, "Information-theoretic sensor motion control for distributed estimation," in *Proceedings of the International Mechanical Engineering Congress and Exposition (IMECE '07)*, November 2007.

[13] G. M. Mathews, H. Durrant-Whyte, and M. Prokopenko, "Decentralised decision making in heterogeneous teams using anonymous optimisation," *Robotics and Autonomous Systems*, vol. 57, no. 3, pp. 310–3320, 2009, selected papers from IEEE International Conference on Multisensor Fusion and Integration (MFI '06).

[14] J. G. Wood, B. Kehoe, and J. K. Hedrick, "Target estimate pdf-based optimal path planning algorithm with application to UAV systems," in *Proceedings of the Dynamic Systems and Control Conference (DSCC '10)*, pp. 749–756, ASME, September 2010.

[15] E. M. Wong, F. Bourgault, and T. Furukawa, "Multi-vehicle Bayesian search for multiple lost targets," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '05)*, pp. 3169–3174, April 2005.

[16] T. Furukawa, F. Bourgault, B. Lavis, and H. F. Durrant-Whyte, "Recursive Bayesian search-and-tracking using coordinated UAVs for lost targets," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '06)*, pp. 2521–2526, May 2006.

[17] J. Wood, *Search and tracking of an unknown number of targets by a team of autonomous agents utilizing time-evolving partition classification [Ph.D. thesis]*, University of California, Berkeley, Calif, USA, 2011.

[18] J. Wood and J. K. Hedrick, "Space partitioning and classification for multi-target search and tracking by heterogeneous unmanned aerial system teams," in *Proceedings of the Infotech@Aerospace*, American Institute of Aeronautics and Astronautics, 2011.

[19] I. R. Goodman, R. P. S. Mahler, and H. T. Nguyen, *Mathematics of Data Fusion*, Kluwer Academic, Boston, Mass, USA, 1997.

[20] G. Mathew, A. Surana, and I. Mezić, "Uniform coverage control of mobile sensor networks for dynamic target detection," in *Proceedings of the 49th IEEE Conference on Decision and Control (CDC '10)*, pp. 7292–7299, December 2010.

[21] G. Mathew and I. Mezi, "Metrics for ergodicity and design of ergodic dynamics for multi-agent systems," *Physica D*, vol. 240, no. 4-5, pp. 432–442, 2011.

[22] H. Chen, K. Chang, and C. S. Agate, "Tracking with UAV using tangent-plus-lyapunov vector field guidance," in *Proceedings of the 12th International Conference on Information Fusion*, pp. 363–372, July 2009.

[23] M. Shanmugavel, A. Tsourdos, B. White, and R. Zbikowski, "Co-operative path planning of multiple UAVs using Dubins paths with clothoid arcs," *Control Engineering Practice*, vol. 18, no. 9, pp. 1084–1092, 2010.

[24] J. Lee, R. Huang, A. Vaughn et al., "Strategies of path-planning for a UAV to track a ground vehicle," in *Proceedings of the 2nd Annual Symposium on Autonomous Intelligent Networks and Systems*, 2003.

[25] S. C. Spry, A. R. Girard, and J. K. Hedrick, "Convoy protection using multiple unmanned aerial vehicles: organization and coordination," in *Proceedings of the American Control Conference (ACC '05)*, pp. 3524–3529, June 2005.

[26] A. Sinha, T. Kirubarajan, and Y. Bar-Shalom, "Autonomous ground target tracking by multiple cooperative UAVs," in *Proceedings of the IEEE Aerospace Conference*, pp. 1–9, March 2005.

[27] P. Skoglar, *Planning methods for aerial exploration and ground target tracking [Licentiate thesis]*, Linköping University, Linköping, Sweden, 2009.

[28] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, John Wiley & Sons, New York, NY, USA, 2001.

[29] M. Godwin and J. K. Hedrick, "Stochastic approximation of an online vehicle routing problem for autonomous aircraft," Infotech@Aerospace Conference, 2011.

[30] M. Alighanbari and J. P. How, "A robust approach to the UAV task assignment problem," *International Journal of Robust and Nonlinear Control*, vol. 18, no. 2, pp. 118–134, 2008.

[31] Z. Kim and R. Sengupta, "Target detection and position likelihood using an aerial image sensor," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '08)*, pp. 59–64, May 2008.

[32] R. Sengupta, J. Connors, B. Kehoe, Z. Kim, T. Kuhn, and J. Wood, "Autonomous search and rescue with ScanEagle," Tech. Rep., 2010, prepared for Evergreen Unmanned Systems and Shell International Exploration and Production Inc.

[33] J. Garvey, B. Kehoe, B. Basso et al., "An autonomous unmanned aerial vehicle system for sensing and tracking," in *Proceedings of the Infotech@Aerospace Conference*, 2011.