

Multiscale Search using Probabilistic Quadrees

Timothy H. Chung

Department of Systems Engineering
Naval Postgraduate School
Monterey, California, USA
thchung@nps.edu

Stefano Carpin

School of Engineering
University of California, Merced
Merced, California, USA
scarpin@ucmerced.edu

Abstract—We propose a novel framework to search for a static target using a multiscale representation. The algorithm we present is appropriate when the target detection sensor trades off accuracy versus covered area, e.g., when a UAV can fly and sense at different elevations. A structure based on quadtrees is used to propagate a posterior about the target location using a variable resolution representation that is dynamically refined in regions associated with higher probability of target presence. Probabilities are updated using a Bayesian approach accounting for erroneous sensor readings in the form of false positives and missed detections. The model we propose is coupled with a search and decision algorithm that determines where to sense next and with which accuracy. The search algorithm is based on an objective function accounting for both probability of detection and motion costs, thus aiming to minimize traveled distances while trying to localize the target. The paper is concluded with simulation results showing our approach outperforms commonly used methods based on uniform resolution grids.

I. INTRODUCTION

Probabilistic search for one or more objects in an area of interest continues to be a topic of active research in the autonomous systems community. With relevance to numerous applications, including but not limited to search for lost objects such as downed aircraft, search and rescue of disaster victims, and detection of submerged threats in maritime environments, the ability to intelligently gather information relevant to the search mission is essential. Due to the ubiquity of mobile robotic sensors, such as unmanned aerial vehicles (UAVs), approaches which address their effective employment and efficient integration of information have also engendered much interest. In particular, the nominal challenges include relatively large spatial scales (that is, the search area is large compared to the sensor footprint), limited search resources such as endurance or search effort, or combinatorial complexity in search planning.

This paper investigates the use of a quadtree representation of both the environment and the underlying probability distribution of the presence of targets. Many search problems face the challenge of having to find small target(s) in a relatively large search region. Further, both sensors and platforms likely have different characteristics when gathering and processing low resolution (large spatial scale) versus high resolution (small spatial scale) information. This multiscale nature highlights the trade off between *exploring* the search region to achieve a broad perspective and *exploiting* focused information to obtain high fidelity data. Notwithstanding, most former research in this area has embraced models based on uniform representations.

The main contributions of this work are three. We first define a novel data structure coined *probabilistic quadtree* that dynamically captures the multiscale evolving posterior of the target location. Then, we propose a formulation of a search decision process where a single stationary target may or may not be in the search region and the objective is specifically associated with the probability of the target's presence at a location. The applicability of a Bayesian formulation combined with the probabilistic quadtree data structure provides a novel framework in which to conduct a family of search-related problems. Finally, we provide extensive simulation results contrasting our proposed method with search strategies based on uniform grids.

The sequel presents relevant research found in the literature, seen to be multidisciplinary in nature given the breadth of probabilistic search applications. In Section III, we provide a probabilistic model formulation of the search problem for one object which defines how the searcher's belief is represented using a quadtree data structure. The associated probability map evolves by integration of imperfect observations via appropriate Bayesian update expressions. In the presence of possible false positive and false negative detection errors, Section IV identifies a search plan for effectively searching the area of interest while maintaining an efficient representation, whereas Section V provides the overall search and decision algorithm. Numerical investigations of the proposed approaches are presented in Section VI, followed by a summary of results and discussion of future avenues in Section VII.

II. RELATED WORK

The historical context for the theory of search is the hunt for submarines during World War II, which led to the application of analytic probability and optimization models to aid in the detection and defeat of these threats [9]. Prior information on the likelihood of a target's presence was used to inform the optimal allocation of constrained resources, that is, the search platforms and the search time allotted to inspect discrete sectors. Though these early works considered arbitrary discretization of the environment, in practice regular rectangular grids are most often used for clarity and ease of representation for most planning purposes. Coarse uniform partitions can lead to inefficiencies in representing spatially-varying quantities, as collective regions of low value (e.g., low probability of target presence) are represented at identical resolution scales as those with high value. Alternatively,

uniform grids at fine resolution suffer from the combinatorial increase in the environment state space.

Cellular decomposition methods which provide weighted, non-uniform cells include Voronoi partitioning [6] and mesh generation [7], and have been extensively used in the robotics literature for problems of robotic coverage [4] and path planning [11]. Unlike the irregularly-shaped cells output by some approaches, regular meshes, such as rectilinear ones easily represented by quadtrees, provide mission relevance and simplicity, including algorithmic methods for, e.g., identifying adjacent cells [7]. These practical considerations are important, e.g., when specifying areas of responsibility for a given search asset within a larger operational context.

From the longstanding probabilistic search community, numerous search theoretic results examine variations on the constrained path optimal search problem. Benkoski *et al.* offer a survey of the classical literature [1], including the “whereabouts” search problem introduced by Kadane [8] in which the searcher must correctly locate the target under search budget constraints or otherwise be able to offer the target’s most likely location. The use of search decision thresholds is considered in [10], but the probability of a target’s presence in a cell is assumed independent of other cells. Other examples of relevant works in the robotics community involving probabilistic search include [3], which investigates the use of an aerial platform conducting search for a lost target. However, the search area and the associated probability maps utilize a fine uniform grid representation. Waharte *et al.* [14] recently investigated more realistic models for sensing, including observation of partial areas and/or overlapping multiple grid cells, which arise in, e.g., variable altitudes of the aerial sensor platform. The role of multiscale resolution, however, is not explicitly incorporated in the representation of the search environment nor leveraged by the searcher to improve its search effectiveness. This work proposes a probabilistic data structure that addresses the joint probabilities of target presence as well as efficiently mitigates the computational complexity that arises from large state spaces.

III. MODEL DEFINITION

In this section we present the models used to maintain the posterior about possible target locations. We first present a hierarchical structure to compute posteriors using multiscale information. Next, we introduce a sensing model intimately tied to the chosen representation, and we then discuss how sensor readings can be used to update probabilities using a Bayesian approach. This section ends with a short description of how the data structure is constructed, updated, and maintained.

A. Probabilistic quadtrees

Our method builds upon a data structure dubbed *probabilistic quadtree*. Informally speaking a probabilistic quadtree is a quadtree where every node is associated with a probability value, and where some constraints relating these

probabilities and the tree topology are enforced. In the framework we present, the topology of the tree varies while the search process advances, i.e., new nodes may be added to the tree or pruned away. In either case, the constraints presented in the following are enforced throughout the process.

We start by shortly recalling some basic concepts from computational geometry (the reader is referred to [7] for a more comprehensive treatment of this topic). A quadtree is a rooted tree in which every internal node has four children. A quadtree is said to be *complete* if all its leaves are at the same depth. Every node (internal or leaf) is associated with a planar region. For sake of simplicity, we herein assume these regions are square, but in general they could also be rectangular. For a given node n , let $R(n) \subset \mathbb{R}^2$ be the square planar region associated with n . Since the association between a node n and its region $R(n)$ is univocal, in the following we will consider the terms “node” and “region” synonymous. If n is an internal node and n_1, \dots, n_4 are its children, then the various regions $R(n_i)$ are obtained by splitting the parent’s area into four equally sized squares. Therefore $R(n) = \bigcup_{i=1}^4 R(n_i)$ and $\text{int}(R(n_i)) \cap \text{int}(R(n_j)) = \emptyset$ whenever n_i and n_j are siblings¹. If \mathcal{T} is a quadtree, then $R(\mathcal{T})$ is the region associated with its root node. Moreover, if n is a node, let $c(n)$ be the geometrical center of its associated square region $R(n)$. In the following, unless otherwise noted, if n is an internal node then n_1, \dots, n_4 will be used to indicate its children. In this paper we consider quadtrees of bounded depth, i.e., while the tree is being modified during the search process we will ensure its depth never exceeds a fixed value. For a tree \mathcal{T} , let $D(\mathcal{T})$ be the maximum depth. Given a tree \mathcal{T} with a fixed maximum depth, we indicate with $N(\mathcal{T})$ the set of nodes of the complete quadtree of depth $D(\mathcal{T})$, while $A(\mathcal{T}) \subseteq N(\mathcal{T})$ is the set of nodes actually in \mathcal{T} . In our framework the tree \mathcal{T} is modified during the process, so while $N(\mathcal{T})$ is fixed, $A(\mathcal{T})$ varies. Finally, for a node n let $d(n)$ be its depth and let 0 be the depth of the root.

Starting from the definition of quadtree we now define a probabilistic quadtree (PQ). This data structure is called *Type1 PQ* because it is aimed to model search problems where at most one target needs to be located. In fact, with some modifications it is also possible to consider scenarios with multiple targets. This extension, called *Type2 PQ* is not discussed in this manuscript for lack of space, but is the topic of further investigations.

Definition 1 (Type1 PQ): A *probabilistic quadtree* of Type1 is a quadtree where each node n is annotated with the probability p_n that a target is located inside its associated region $R(n)$. In a Type1 PQ the following constraint is enforced: if n is an internal node, then

$$p_n = p_{n_1} + p_{n_2} + p_{n_3} + p_{n_4}. \quad (1)$$

A Type1 PQ is used when considering a search task where there is at most one stationary target located in $R(\mathcal{T})$. In this case it is convenient to define a Bernoulli random variable X_n associated with every node in the tree, with $X_n = 1$ indicating the target is inside $R(n)$. The probability linked

¹ $\text{int}(R(n_i))$ is the internal of region $R(n_i)$.

to a node n is then $p_n = \Pr[X_n = 1]$. Let n be an internal node and $n_i \neq n_j$ be two of its children. Because it is assumed there is at most one target, the events $X_{n_i} = 1$ and $X_{n_j} = 1$ are disjoint, and then the constraint given in Eq. 1 holds. Since the possibility that no target is present in the search area is also legitimate, it is convenient to include a further node n_\emptyset to track this hypothesis, and let p_{n_\emptyset} be the probability of this event. In a Type1 PQ let $P(H)$ be the probability that one target is located in $R(\mathcal{T})$. Then, $P(H)$ is given by the sum of the probabilities defined on the leaves in \mathcal{T} , while the probability that no target is located therein is then $1 - P(H) = p_{n_\emptyset}$.

It is pivotal to observe that in a Type1 PQ all variables X_i are related with each other because we assumed there is at most one target located in $R(\mathcal{T})$. Therefore when updating one probability value p because of a new sensor reading, it is necessary to update all the others as well. These considerations are key in order to understand the Bayesian update procedures described after the sensor model is described.

B. Sensor model

We assume the availability of a sensor capable of detecting a target in an assigned sensing area. As commonly done in related literature, we model the sensor reading with a Bernoulli random variable assuming value one when detection occurs. When searching for a target in a region $R(\mathcal{T})$ we consider the availability of a single sensor that can be moved throughout the region. The sensing area and the sensor performance depend on where the sensor is placed. We assume the sensor can be positioned only at a discrete set of locations defined as

$$\mathcal{S} = \{s \in \mathbb{R}^2 \mid \exists n \in N(\mathcal{T}) \text{ with } c(n) = s\}$$

i.e., it is possible to sense only at the centers of the nodes of the complete tree of maximal depth associated with the search area. The reader should note that \mathcal{S} defines the set of all possible sensing locations, but when selecting where to sense next we will restrict the search domain to $A(\mathcal{T})$, i.e., the set of nodes currently in the tree. When sensing at point $s = c(n)$ the sensing area is $R(n)$, i.e., the square associated with the center where it is positioned. We hypothesize that sensing errors may occur in the form of false positive or false negative detections. Let Z_n^t be the random variable describing the outcome of an observation performed at point $c(n)$ at time t . In the following we will sometimes write just Z^t to indicate a sensor reading at time t without specifying where it was taken. The sensor performance is given by the following parameters:

$$\Pr[Z_n = 1 | X_n = 0] = \alpha(n)$$

$$\Pr[Z_n = 0 | X_n = 1] = \beta(n)$$

where the superscript t has been dropped because these detection probabilities are assumed stationary. The false positive (i.e., “false alarm”) rate $\alpha(n)$ and the false negative (“missed detection”) rate $\beta(n)$ are functions of the area being sensed, i.e., $R(n)$. Specifically, we formalize the intuition

that when sensing a large area errors of either type are more likely to occur, whilst when concentrating the sensing effort on focused regions accuracy increases. These principles are formalized by the following constraints:

- 1) $d(n_1) < d(n_2) \Rightarrow \alpha(n_1) \geq \alpha(n_2)$,
- 2) $d(n_1) = d(n_2) \Rightarrow \alpha(n_1) = \alpha(n_2)$.

Identical trends are supposed for $\beta(n)$. These constraints model, for example, the scenario where a UAV may fly at different altitudes while searching for a target in a given area by using a nadir-pointing camera. When the UAV flies at high altitudes it senses a larger region but its error rate is higher. Conversely, if the UAV lowers its altitude it senses a smaller region but with greater fidelity. These two different choices are associated with choosing a sensing point close to the root of \mathcal{T} versus a sensing point at higher depth. In the following we will repeatedly refer to the *uniform* nature of the sensing process. This means that when detection occurs in a given region the detected target could be located anywhere therein with the same probability. Similarly, if an error occurred, it could have happened anywhere and with the same probability.

C. Bayesian updates

We now discuss how probabilities are updated when a new sensor reading Z_n^t is received. In the following we use standard notation for Bayesian estimation, i.e.,

$$p_n^t = \Pr[X_n = 1 | Z^1, \dots, Z^t]$$

where it is furthermore assumed that the prior p_n^0 is available. This process is conceptually similar to the problem described in [5], but some changes in the posterior update method emerge because of the representation embraced in this manuscript. The starting point are the following equations (see [5] for details):

$$\Phi(Z_n^t) = (1 - Z_n^t)(1 - \alpha(n)) + Z_n^t \alpha(n)$$

$$\Psi(Z_n^t) = (1 - Z_n^t)\beta(n) + Z_n^t(1 - \beta(n))$$

$$\Theta_n(Z_m^t) = \begin{cases} \Psi(Z_m^t) & \text{if } m = n \\ \Phi(Z_m^t) & \text{if } m \neq n \end{cases}$$

$$p_n^t = \frac{\Theta_n(Z_m^t)p_n^{t-1}}{\Phi(Z_m^t)(1 - p_m^t) + \Psi(Z_m^t)p_m^t}. \quad (2)$$

In the last relationship the reader should note that in general it could be $n \neq m$, i.e., we also need to update the posterior probability for cells different from the one where sensing occurred. This is consistent with the former statement saying that all cells are correlated with each other. The reader should notice that in principle Eq. 2 could also be used to compute p_{n_\emptyset} , although in practice one can also obtain this value as one minus the sum of probabilities stored in the leaves of \mathcal{T} .

We now discuss how these relationships are applied to update posteriors in a Type1 PQ. Upon receiving a sensor reading Z_m^t Eq. 2 is first applied to compute p_m^t and then

p_n^t for all nodes n that are leaves of \mathcal{T} . If m is an internal node then the change in its probability value needs to be propagated to its children. The change in probability is redistributed proportionally to the children, i.e., considering the constraint given by Eq. 1 we set:

$$\delta p = p_m^t - p_m^{t-1}$$

and we then update the probability of all m 's children setting

$$p_{m_i}^t = p_{m_i}^{t-1} + \left(\frac{p_{m_i}^{t-1}}{p_m^{t-1}} \right) \delta p. \quad (3)$$

Changes applied to the m 's children are then recursively propagated to their descendants until they reach the leaves. Once no more updates are necessary, a further sweep of \mathcal{T} is needed in order to reinforce the constraint on the internal nodes. This is easily done by visiting the tree bottom up, starting from the leaves and repeatedly applying Eq. 1. Overall, the complexity of the Bayesian update is $\mathcal{O}(|A(\mathcal{T})|)$, where $|A(\mathcal{T})|$ is the number of nodes in \mathcal{T} .

D. Construction and Update of a Probabilistic Quadtree

Before the search procedure starts, \mathcal{T} must be initialized. Two approaches are possible. If no prior information is given, then a tree consisting of just the root is built, with $R(\mathcal{T})$ equal to the area where the search takes place. An initial probability value assigned to the root ($p_n = p_n^0$) gives the *a priori* confidence that one target is present in the area. If this value is not provided, then one may set $p_n = 0.5$ to indicate total lack of *a priori* information. Alternatively, the initial decomposition of the environment can be extracted from a prior belief probability distribution. For example, when searching for a lost ship one may be given a distribution related to the last known location. From this *a priori* distribution a PQ may be extracted in a number of ways, including restrictions on the maximum depth or number of nodes in the initial quadtree. Such considerations may have practical relevance in terms of computation. Figure 1 shows two possible initializations of the search area, $R(\mathcal{T})$, using the probabilistic quadtree structure. Figure 1(a-b) represents a decomposition induced by the initial belief probability distribution while restricting the maximum number of nodes (a.k.a. the *order*) of the tree to $A(\mathcal{T}) = 300$ nodes. The resultant quadtree has nodes at varying depth, with high depth (i.e. high resolution) corresponding to areas of high spatial variations in the belief probability. Alternatively, consider Figure 1(c-d), where the quadtree is constructed to a specified initial maximum depth, which highlights the fact that the uniform grid decomposition is a special case of the hierarchical quadtree representation. Note that the uniform grid may not provide the finest granularity necessary for the search process, i.e., the depth of the initial tree may not equal $D(\mathcal{T})$. Moreover, in the framework we propose the searcher starting with a coarse uniform grid will iteratively refine it to gather more information in regions where the target is more likely to be located. On the contrary, approaches based on flat uniform grids do not refine their representation adaptively.

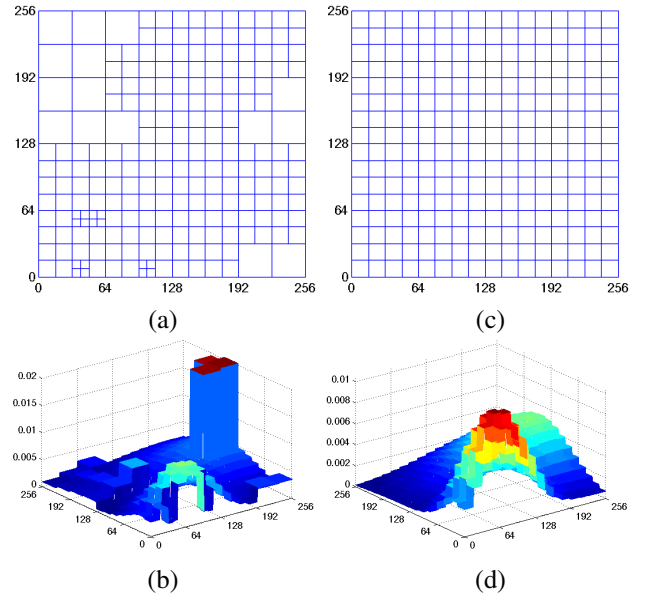


Fig. 1. Illustration of the initial partitioning of the search region and associated belief probability map using a probabilistic quadtree with (a-b) maximum initial number of nodes of 300, or with (c-d) maximum initial depth of the tree, $D(\mathcal{T}) = 4$.

During the search process the tree will be expanded by splitting a leaf node into four children, in the spirit of quadtree representations. The probability values stored in the children of the node being split follow from the constraints being enforced and from a uniform sensing assumption. That is to say that when a leaf node n in a Type1 PQ is split, each of its children is initialized with a value $p_{n_i} = p_n/4$. Hence, by splitting leaf nodes, the search algorithm is able to expand the depth of the tree in a focused way driven by the sensing information collected while searching. Pruning a tree is also a straightforward process because the constraint given by Eq. 1 is always enforced in the tree. Therefore pruning is achieved by simply removing all descendants of an internal node n . When this happens, no further probability updates are needed because n already satisfies Eq. 1 and seamlessly becomes a leaf. Hence, \mathcal{T} may be updated with time complexity $\mathcal{O}(1)$.

E. Problem definition

Starting from the elements described above, we can now formally define the search and decision problem as follows. Given \mathcal{T} , determine a sequence of valid sensing operations aimed to locate a possible target located in $R(\mathcal{T})$. The process shall eventually output either:

- the decision *No target present*;
- the decision *Target present* together with node n with $d(n) = D(\mathcal{T})$ (i.e., at maximum depth) such that the target is inside $R(n)$.

Moreover, while doing so we aim to optimize the overall distance traveled by the sensor while moving between successive sensing locations.

IV. SEARCH PLANNING

The objective of planning the searcher's route through the discretized search area of interest is to appropriately

identify the presence (and location) or absence of a single target. Specifically, we wish to maximize the probability of detecting the target by choice of not only *where* to gather observations but also at *what resolution*.

Consider the initial belief distribution along with an initial quadtree representation of the probabilistic environment to be given. The searcher seeks to maneuver to the location containing the maximal probability of detection, weighted by the cost to transit between cells as well as the value of the resolution of observations as measured by depth in the quadtree structure. These principles lead to the definition of the objective function, J , to be optimized as

$$J \triangleq \frac{p(n) \cdot A^{d(n)}}{\text{cost}(n', n)}, \quad n, n' \in A(\mathcal{T})$$

where n' is the searcher's current node location and n is the decision variable over which the optimization occurs.

The inclusion of transit costs $\text{cost}(n', n)$ represents the challenge of constrained paths for physical robotic sensor platforms, while the dependence on depth within the tree captures the challenge of obtaining information at desired and/or necessary resolutions. In this manner, the transit cost includes not only the spatial (e.g., Euclidean) distance between (the centers of) two regions, but also the effect of moving to lesser or greater depth in the quadtree. Returning to the example of the UAV search application, additional costs are incurred if the UAV ascends to a higher altitude to obtain a broader field of view, while possible benefits are accrued for descent at the expense of a smaller sensor footprint. For the purposes of formulating the model, let us propose a simple measure of the total transit cost given by:

$$\text{cost}(n', n) \triangleq \begin{cases} \|n' - n\| + \kappa_{\text{ascent}} \Delta(n', n) & \text{if } \Delta(n', n) > 0, \\ \|n' - n\| + \kappa_{\text{descent}} \Delta(n', n) & \text{if } \Delta(n', n) \leq 0. \end{cases}$$

where $\|n' - n\|$ is defined as the Euclidean distance between the center coordinates of the n -th and n' -th regions, and $\Delta(n', n) = d(n') - d(n)$ is the net change in quadtree depth. The scaling parameters, κ_{ascent} and κ_{descent} , describe the relative cost or benefit of ascending or descending, respectively, within the quadtree structure. Note that $\text{cost}(n', n)$ may not be symmetric if the costs to ascend and descend are not equal. Figure 2 illustrates the determination of transit costs for an example path including ascent, level transit, and descent segments.

A. Myopic Optimization

The computational complexity of the constrained path search optimization problem (NP-complete for maximizing probability of detection or NP-hard for minimizing expected time till detection) [13] motivates search approaches which mitigate this computational burden. Consider a myopic optimization of the objective function [15], such that the next cell for the searcher to inspect is

$$n^* = \arg \max_n J, \quad n \in A(\mathcal{T})$$

The best node to search, n^* , represents the best reward-to-cost ratio, which in special cases of the discrete object

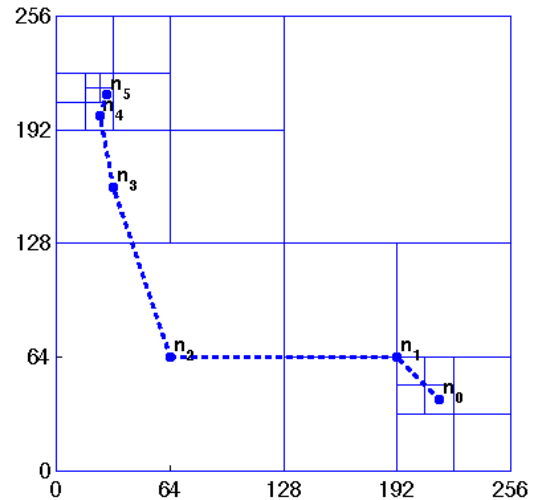


Fig. 2. Depiction of an illustrative searcher route with path given by the sequence $\{n_0, n_1, \dots, n_5\}$. The leg from node n_0 to n_1 has a change in depth of $\Delta(n_0, n_1) = 3$ (ascent), segment n_2n_3 incurs only Euclidean distance cost during level transit, and segments n_2n_3 , n_3n_4 , and n_4n_5 correspond to descent legs. The net change in depth of the path is $\Delta(n_0, n_5) = -1$, representing descent of one level.

search problem can be shown to be the optimal (e.g., [2]) node to inspect. Further, this search optimization strategy is *adaptive* [12], in that the resulting search plan utilizes the dynamically updated belief probability distribution to guide the search effort at every time step. Such search strategies offer robustness to uncertainty in the face of dynamic environments, and the proposed myopic strategy additionally offers computational ease in computing the best node to search.

B. Route Planning

Once the goal node, n^* , has been determined, the searcher must transit to the desired location; however, the mobile platform is assumed in this work to necessarily travel through adjacent regions en route to the goal. We can straightforwardly determine the adjacency relationships between the current location of the searcher and all neighboring regions, where two regions are considered adjacent if they share (any portion of) one or more boundaries [7]. By convention, the adjacency list contains the current node as well as its ancestors and descendants. The location-dependent adjacencies induce a weighted graph on the set of active nodes $A(\mathcal{T})$ in the PQ, where an edge exists between two active nodes n and n' iff they are adjacent, and the edge weight is precisely the transit cost, that is, $\text{cost}(n', n)$, as defined previously. Leveraging the adaptive nature of the myopic search optimization, we consider only the next location en route to the goal node by selecting a one-step reachable region adjacent to the current location that possesses minimal transit cost. The searcher then recomputes the next best location and continues to move toward it via adjacent nodes until the goal is reached. Only upon reaching the goal node, n^* , does the searcher make an observation of the region $R(n^*)$ and update the belief probabilities according to the sensor and Bayesian update

models described earlier. This assumption that the searcher does not perform any inspections while in transit stems from practical considerations, such as diminished information capture at transiting speeds or operational protocols requiring immediate and focused response to any positive contact. However, the model presented herein can easily address continuous observations during transit as well as other sensing schedules or modes, and is left for future study.

V. SEARCH DECISION AND TREE EVOLUTION

The decision process terminates with a negative answer when $p_{n_0} > P_{null}$, where P_{null} is a predetermined threshold. The positive answer is instead produced when the probability of a node at the maximum depth $D(\mathcal{T})$ exceeds a given confidence level P_{found} , and the region associated with that node is output as the target location. It is important to stress that termination with a positive answer occurs only when high confidence is achieved at the deepest level, so that a positive answer is associated with a target location at the highest resolution (i.e., smallest area of the associated region). A hypothetical high confidence associated with a node n close to the root will not result in a positive decision being taken because n is linked to a coarse resolution region.

Finally it is necessary to address how the tree will be refined during the search process. Every time a sensor reading Z_n is received, the posterior is updated according to the method outlined in Section III-C. If node n is internal, then the probability change is recursively propagated to its descendants according to Eq. 3. However, if $Z_n = 1$ and n is a leaf with $d(n) < D(\mathcal{T})$, the node is split and four children are generated, each initialized with probability $p_n/4$. This refinement of the tree thus extends the search space around locations that returned a positive detection. Moreover, since the newly added children are at a deeper level, the search space is enriched with locations associated with more accurate sensor readings, i.e., lower values for α and β . On the contrary, if n is an internal node and a reading $Z_n = 0$ is received, the node shall be considered for pruning, provided that its associated probability falls below a given threshold. However, rather than removing branches altogether we adopt a slightly different strategy, i.e., the branch is not pruned but rather marked as *cleared*. Cleared nodes are not removed from the tree, but are not considered part of $A(\mathcal{T})$ when searching for a value optimizing J . However, during the Bayesian update their probabilities are still updated and if they cross back above the threshold, the *cleared* mark is removed from the branch and the nodes come back into the active tree. We stress that the threshold is level dependent, with higher values associated with deeper nodes. This choice aligns with the conservative intuition that it should be harder to declare clear nodes closer to the root, which are associated with large search regions, and therefore their threshold is lower and harder to achieve. In the pseudocode we indicate with $\underline{B}(d)$ this depth dependent value. By marking nodes as cleared, rather than removing them altogether, regions formerly removed from the search space can be reconsidered as soon as one of its components is not cleared anymore

without having to recreate the structure constructed earlier. The overall search and decision strategy is sketched in pseudocode in Algorithm 1.

Algorithm 1 Overall search and decision algorithm

```

 $\mathcal{T} \leftarrow \text{InitializeTree}(\text{Prior})$ 
 $p_{n_0} \leftarrow 1 - P(\text{root}(\mathcal{T}))$ 
loop
   $n \leftarrow \arg \max_n J, \quad n \in A(\mathcal{T})$ 
  Determine route and move to node  $n$ 
   $Z_n^t \leftarrow \text{SensorReading}$ 
  Update posterior and  $p_{n_0}$ 
  for all  $n \in N(\mathcal{T})$  do
    if  $p_n \geq \underline{B}(d(n))$  and  $n$  is cleared then
      Remove clear mark from  $n$  and all its descendants
    if  $p_{n_0} > P_{null}$  then
      Output No target present. Stop
    else if  $d(n) = D(\mathcal{T})$  and  $p_n > P_{found}$  then
      Output Target found at location  $R(n)$ . Stop
    if  $Z_n^t = 1$  and  $n$  is a leaf and  $d(n) < D(\mathcal{T})$  then
      Expand  $\mathcal{T}$  by splitting node  $n$ 
    if  $Z_n^t = 0$  and  $n$  is internal and  $p(n) < \underline{B}(d(n))$  then
      Mark as cleared all its descendants

```

VI. EXPERIMENTAL RESULTS

Define an $L \times L$ search area of interest, where L is the length of the side of the square search area $R(\mathcal{T})$, assuming unit length for the side of region $R(n)$ for nodes at maximum depth. Consider the search area illustrated in Figure 1, which will be used for demonstrating the proposed search formulation and methodology using the probabilistic quadtree. The PQ \mathcal{T} in this case has $L = 256$, or in other words, $D(\mathcal{T}) = 8$. Throughout the numerical experiments conducted, the search parameters associated with the sensor characteristics, search decision thresholds, transit cost factors, etc., are held fixed such that the randomness introduced in the replications is due to the imperfect sensor observations. The initial aggregate belief is 0.75, and the search mission duration is 1000 time steps, in that the decision process reports "No target present" if after 1000 time steps no cell at the deepest level crossed the threshold P_{found} . Recall that the sensor characteristics as well as the search decision thresholds are dependent on the depth within the quadtree, such that for $d = 0, \dots, D(\mathcal{T})$, the false positive and negative detection error probabilities are given by $\alpha(d) = 0.2 - 0.01d$ and $\beta(d) = 0.3 - 0.02d$, respectively. Similarly, the lower search decision thresholds is $\underline{B}(d) = 0.0 + 0.01d$, and $P_{found} = 0.9$. For transiting, the ascent and descent factors are $\kappa_{\text{ascent}} = 1000$ and $\kappa_{\text{descent}} = 0$. Finally, the searcher commences its search by observing the search area initially at the quadtree's root node. Exploration of the factor space and the sensitivities therein is subject of future investigations.

A. Belief Evolution and Trajectory

The evolution of the aggregate belief describes the dynamic change in the probability that the target is present

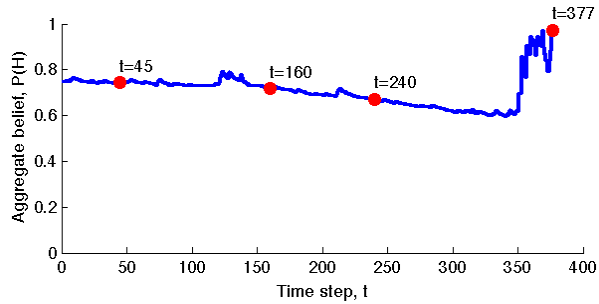


Fig. 3. An example of the evolution of the aggregate belief, namely the probability of target presence in the search region, $R(T)$. The initial belief value is 0.75, and the search process concludes at time step $t = 377$ correctly locating the target at maximum resolution (i.e., at maximum depth in the quadtree). The time step labels correspond to the snapshots of the searcher’s trajectory shown in Figure 4(a)-(d).

in the search area. An illustrative evolution of this search process is depicted in Figure 3, with Figures 4(a)-(d) providing snapshots of the searcher’s trajectory through the search region. Starting at the root node and with an initial belief of 0.75 that the target is present, the searcher employs the myopic search strategy, leading it to relatively nearby regions of high likelihood of finding the target. In Figure 4(a), the searcher begins to inspect the nearby peak (eastern region), but upon mostly clearing that area, finds it beneficial to transit and search the southwestern peak (shown in (b)). Only after a thorough but unsuccessful search, illustrated by the searcher track in (c), does the searcher revisit the eastern section, where it finally arrives at the true target location and investigates more deeply until declaring completion of the search process. An accompanying video shows how the posterior is updated during the search and how it drives the search process².

Numerical studies with 1000 replications of the simulation yields an approximate probability distribution, illustrated in Figure 5, of the time till the correct search decision is made, with expected number of time steps $\mathbb{E}[TTD] \approx 419$. Note that for these simulation experiments, the stationary target is located at $c(n) = (200.5, 200.5)$, for the region at maximum depth, i.e., a 1×1 unit area.

B. Computational benefit of hierarchical structure

In addition to the advantages offered by the PQ representation for improving the search performance, another benefit is in the reduction of computational overhead in data storage and maintenance. In order to achieve the desired search objective at the required resolution (i.e., detection of a target within unit area), the state space for a uniform grid is necessarily $L \times L$, or in the presented example, $256^2 = 65,536$ nodes.

In contrast, the proposed approach in this paper leverages the exponential character of the probabilistic quadtree to dramatically reduce the state space, which also benefits the

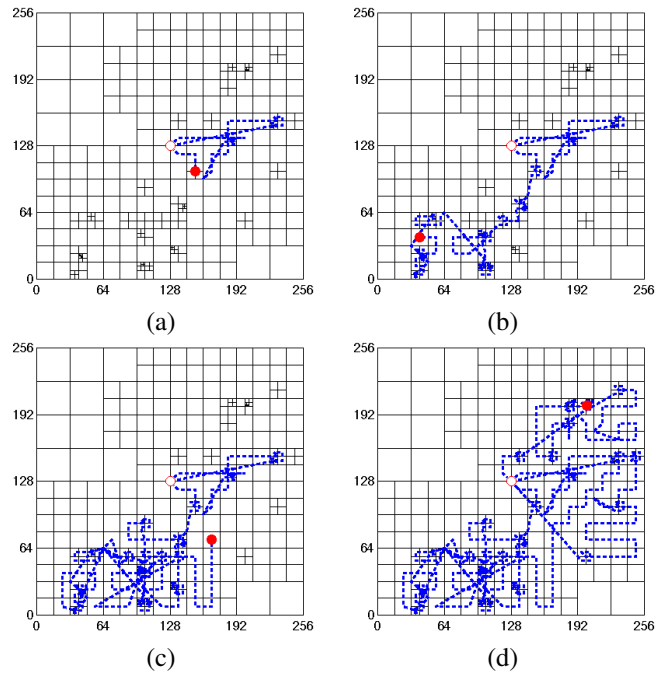


Fig. 4. Illustration of an example search trajectory, with the searcher starting at the root node (i.e., center location). Panels (a)-(d) correspond to time steps, $t = 45, 160, 240, 377$, resp. (also shown in Figure 3). Note that the searcher first inspects one region of high target probability, then transits to another peak, before returning to the first region, where it successfully detects and localizes the target at maximal resolution.

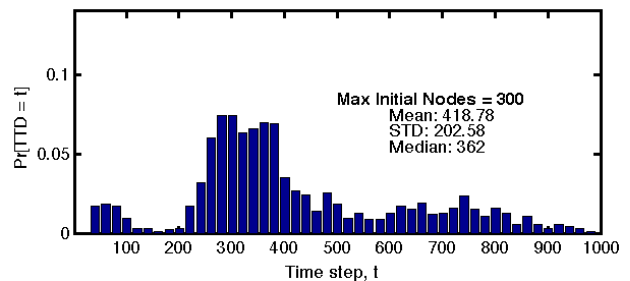


Fig. 5. Probability distribution of the time till correct decision for the probabilistic quadtree T initialized with a maximum of 300 nodes. The expected time for a correct search decision is $\mathbb{E}[TTD] \approx 419$ time steps.

search optimization and Bayesian update subroutines. Examining Figure 6, we see that on average over 1000 simulation trials, the size of the PQ is two orders of magnitude less than that of a flat data structure representing a uniform grid (i.e., $\sim 600 \ll \sim 65,000$ nodes). Additional computational advantage is obtained by including the clearing and/or pruning of nodes during the evolution of the search process.

Alternatively, suppose the searcher has limited computational resources and must restrict the state space of its representation of the belief. The trade off in this case is with the achievable resolution of the search. For example, in order for a uniform grid over a 256×256 search area to have a comparable number of nodes as the PQ construction, the searcher is limited to localizing the target to no better than within an 8×8 grid cell, which requires a state space of 1024, versus accuracy to a 1×1 area with the PQ.

²Matlab® code producing these results is available for download at <http://robotics.ucmerced.edu>.

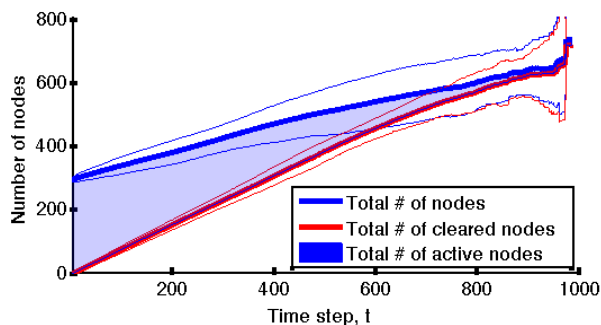


Fig. 6. Evolution of the total number of nodes in the quadtree (initially restricted to a maximum of 300 nodes). The upper curve represents the growth of the quadtree size without clearing low probability nodes, whereas the lower curve shows the number of nodes cleared from the tree, representing the amount of search area cleared. The shaded region between these curves represent the number of active nodes $A(\mathcal{T})$ in the PQ, i.e., the size of the state space of the search problem. Thin bounding lines represent 3σ error intervals over 1000 simulation runs.

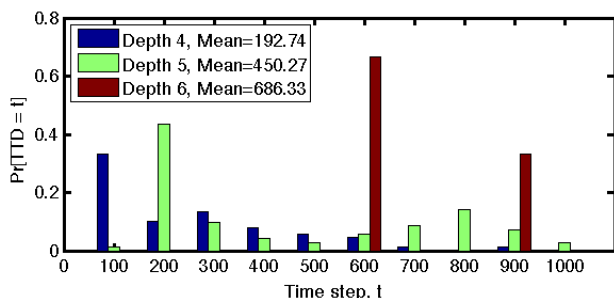


Fig. 7. Decision time probability distributions for search on a uniform grid. Three different resolutions, corresponding to equivalent PQ depths 4, 5, and 6, are represented, with mean decision times of $t \approx 193, 451, 687$, resp. Averages were taken across $N = 100$ simulation runs for depth 4 and 5, and across $N = 10$ runs for depth 6.

Another computational trade off can also be considered in the context of search performance. If the search process is required to terminate within a fixed time budget, which is often the case for many operational scenarios, we can also study the expected time until the correct search decision is made for uniform grids of varying resolution. Figure 7 shows the probability distribution of the decision times for different uniform discretizations of the search area. Equivalent to a PQ at depths 4, 5, and 6, the uniform region resolutions are 16×16 , 8×8 , and 4×4 , respectively. The resultant probability distributions on search decision times show that if the search must be completed in, for example, ~ 500 time steps (c.f. Figure 5(a)), then the hierarchical structure is once again superior. In other words, within the specified mission duration, the PQ approach enables localization of the target down to the maximum 1×1 resolution, whereas one must use a uniform grid with an inferior 8×8 resolution (i.e., depth 5) to meet the time limit.

VII. CONCLUSIONS

This paper proposed a novel probabilistic framework for the spatial search of a stationary target using a single searcher with imperfect sensors. We introduced the concept of probabilistic quadtrees which provide an efficient computational

data structure for representing the search region and the corresponding belief probability map of target presence at multiple resolutions or spatial scales. In addition to addressing the Bayesian update expressions for integrating possibly erroneous detection observations on the quadtree, we proposed myopic search optimization and route planning algorithms to provide autonomous and adaptive search to best accomplish the search objective. Various benefits of the proposed hierarchical approach for both search performance and computational advantages over uniform grid representations were highlighted via results from simulation studies.

There are numerous avenues for future exploration. Various analyses on the effect of different initialization schemes for the probabilistic quadtree are appropriate, including dependence of the search performance on the initial number of nodes in the PQ. Exploration of different transit cost measures, including sensitivities to changing ascent and descent costs, are also of interest, which can incorporate realistic constraints such as fuel or time late to goal. Additionally, extension to multiple searchers with different sensing and motion capabilities will offer new insights into multi-agent search. Calibration of physical depth-dependent sensor characteristics, e.g., computer vision detection algorithms for a UAV-borne camera, is forthcoming in live field experimentation efforts at facilities located at Camp Roberts, CA. Other elements can further be implemented for live-fly field experiments demonstrating the efficacy and operational relevance of the proposed work.

REFERENCES

- [1] Stanley J. Benkoski, Michael G. Monticino, and James R. Weisinger. A Survey of the Search Theory Literature. *Naval Research Logistics*, 38(4):469–494, August 1991.
- [2] William L Black. Discrete Sequential Search. *Information and Control*, 8(2):159–162, 1965.
- [3] Frederic Bourgault, Tomonari Furukawa, and Hugh F. Durrant-Whyte. Optimal Search for a Lost Target in a Bayesian World. *Field and Service Robotics (STAR Springer Tracts in Advanced Robotics)*, 24:209–222, 2006.
- [4] F. Bullo, J. Cortés, and S. Martínez. *Distributed Control of Robotic Networks*. Princeton, 2009.
- [5] T.H. Chung. On probabilistic search decisions under searcher motion constraints. In G.S. Chirikjian et al., editor, *Workshop on Algorithmic Foundations of Robotics*, volume 57 of *STAR*, pages 501–516. Springer, 2010.
- [6] J. Cortés, S. Martínez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.
- [7] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry*. Springer, 2000.
- [8] J. B. Kadane. Optimal Whereabouts Search. *Operations Research*, 19(4):894–904, July 1971.
- [9] Bernard O Koopman. Search and Its Optimization. *The American Mathematical Monthly*, 86(7):527–540, 1979.
- [10] Moshe Kress, Roberto Szechtman, and Jason S Jones. Efficient Employment of Non-Reactive Sensors. *Military Operations Research*, 13(4):45–57, 2008.
- [11] S.M. LaValle. *Planning algorithms*. Cambridge academic press, 2006.
- [12] L.D. Stone. *Theory of Optimal Search*. Academic Press, 1989.
- [13] K E Trummel and J R Weisinger. The Complexity of the Optimal Searcher Path Problem. *Operations Research*, 34(2):324–327, 1986.
- [14] Sonia Waharte, Andrew Symington, and Niki Trigoni. Probabilistic Search with Agile UAVs. In *2010 IEEE International Conference on Robotics and Automation*, pages 2840–2845. IEEE, May 2010.
- [15] Alan R Washburn. *Search and Detection*. Topics in Operations Research Series. INFORMS, 4th edition, 2002.