AIAA Guidance, Navigation, and Control Conference and Exhibit
21 - 24 August 2006, Keystone, Colorado

AIAA 2006-6196

Occupancy Based Map Searching Using Heterogeneous Teams of Autonomous Vehicles

Christopher W. Lum,* Rolf T. Rysdyk,† and Anawat Pongpunwattana ‡

Autonomous Flight Systems Laboratory

University of Washington, Seattle, WA, 98105, USA

In typical search missions, the environment and the targets are not stationary and previous observations become less reliable as time progresses. In addition, the search is often initiated with only a rough idea of target location. In this work we consider a strategy for searching using a team of heterogeneous autonomous vehicles. The team members maintain a world model which includes the estimate of possible target states. The issue of compelling agents to converge on targets and to search unexplored regions is formulated as a model predictive control problem. The world model is propagated in time and strategic decisions are made autonomously based on its prediction. Agents formulate control decisions by optimizing an objective function which allows for control and timing constraints. Individual agents in the team are coupled to one another through the centralized occupancy based map. This coupling, in combination with the outlined search strategy, leads to an efficient, autonomous, and cooperative search.

Nomenclature

B	Set of $\overline{z}$ values defining spatial domain of occupancy based map
$\tilde{B}$	Set of $\overline{z}$ values defining center of occupancy based map cells
$D(k)$	Event of detecting the target at time step k
d	Prediction horizon
$f_\chi(z)$	Potential function for course deviation
$g(a, b)$	Function which samples from set a based on set b
$h(a, b)$	Calculates the predicted future state of the world given target state a and world state b
l	Lower bound for box set in $(\wp_2)$
L_x, L_y	Width and height of occupancy based map cell (x-direction and y-direction, respectively)
M	Number of particles in $(\wp_2)$
n	Dimension of particle state in $(\wp_2)$
$n^{[m]}(t)$	Noise added to particle m at time t in $(\wp_2)$
N_x, N_y	Number of columns and rows, respectively, of occupancy based map
$p(A\|B)$	Conditional probability of A given B
P_E, P_N	East and north position, respectively
V_{max}	Maximum velocity of agent
$rand(l, u)$	Function which produces a uniformly distributed random number in the range $[l, u]$
r_{max}	Maximum distance agent can travel in a single step
u	Upper bound for box set in $(\wp_2)$
$\overline{w}$	Decision vector in $\Re^{2\cdot d}$
$\overline{w}^\star$	Optimal solution to $(\wp_3)$
$w^{[m]}(t)$	Weight of particle m at time t in $(\wp_2)$
$(\wp_1)$	Subproblem of creating future world state estimates
$(\wp_2)$	Subproblem of finding desirable cells for agent to search

*Research Assistant, Dept. of Aeronautics and Astronautics, lum@u.washington.edu, AIAA student member
†Assistant Professor, Dept. of Aeronautics and Astronautics, rysdyk@aa.washington.edu, AIAA member
‡Research Associate, Dept. of Aeronautics and Astronautics, anawatp@u.washington.edu, AIAA member

$(\wp_3)$	Subproblem of finding trajectories for agent flight path
X	A box set
$x^{[m]}(t)$	Particle m at time t in $(\wp_2)$
$\tilde{x}^{[m]}(t)$	Resampled particle m at time t (no noise added yet) in $(\wp_2)$
$\overline{x}_{agt}(k)$	Agent state $(P_E\ P_N\ \chi)^T$ at step k
$\overline{x}_{tgt}(k)$	State of the target at time step k
$\hat{x}_{tgt}(k)$	Estimated target state at time step k
x_{min}, x_{max}	Minimum and maximum x value of occupancy based map
$x_w(k, \overline{z})$	Actual state of the world at time step k and location $\overline{z}$
$\hat{x}_w(k, \overline{z})$	Estimated world state at time step k and location $\overline{z}$ (used in $(\wp_1)$)
$x_{w,nom}$	Nominal score of occupancy based map
$\overline{x}^\star$	Average of all particles after particle filter terminates
y_{min}, y_{max}	Minimum and maximum y value of the occupancy based map
$\overline{z}$	(x, y) coordinate $(P_E\ P_N)^T$
$\overline{z}^\star$	Most desirable cell in $(\wp_2)$
$\overline{z}_0$	Agent's current coordinate
Z	Set of all locations that the agent can reach in d steps
α	Scalar tradeoff parameter for $f_\chi(\overline{z})$
$\chi(t)$	Particle set at time t
ΔT	Time between steps
ϵ	Small number
$\eta(a, b)$	Calculates absolute angular difference between angles a and b
τ	Time constant for occupancy based map scores

I. Introduction

Search-and-surveillance type missions typically require heavy human involvement. Tasks of assigning regions to search and coordinating sensor measurements are usually left to human decision making and analysis. In a noisy environment, it becomes difficult for a human operator to classify sensor readings and assign confidence in these readings. Determining regions of high target-location probability and coordinating nearby agents to converge on a particular spot while allowing other vehicles to continue searching is also difficult. Therefore, the primary limitation to concurrent operation of multiple vehicles remains lack of autonomy of these vehicles. Many of these search type missions are initiated with a poor estimate of the target's actual position. To aggravate matters, often the target is moving or evading, making old observations less reliable as time progresses. In order to minimize human interaction, an efficient method for searching a region for a target is needed.

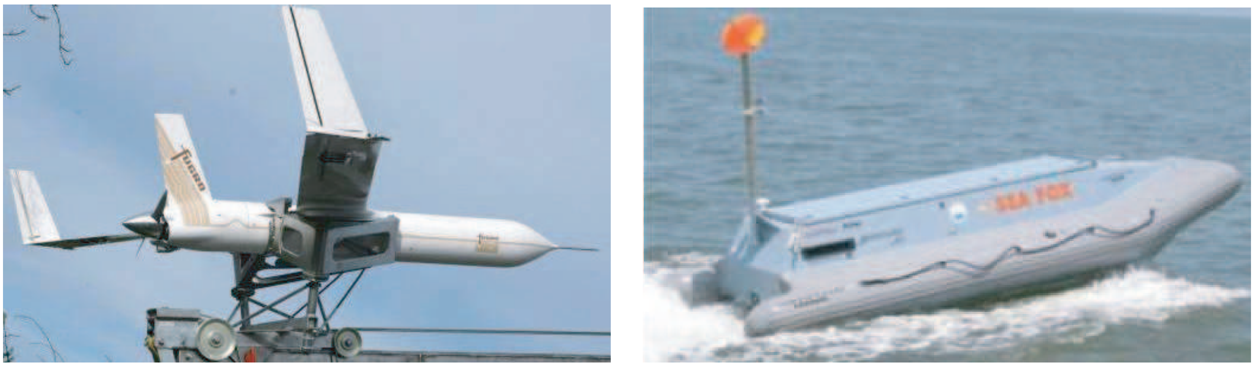
Groups such as Durrant-Whyte et al.[1] have studied the problem of searching for a target using a Bayesian probabilistic approach and have investigated some of the communication issues involved in such a search. Polycarpou et al.[2] have applied optimization techniques to generate search patterns over a finite amount of steps. The search strategy presented here follows a similar approach and investigates the effect of incorporating a predictive world estimate in the problem of finding an optimal search pattern.

In this simulated scenario, a team of heterogeneous agents are searching for a submerged submarine in a marine environment. Agents are equipped with sensors such as magnetometers (sensors which are able to search for targets based on their magnetic signatures) and sonar. This paper does not address the task of performing target identification with these sensors[3] but instead concentrates on developing a search strategy that can be used with a team of heterogeneous agents to coordinate a searching mission for a possibly moving target.

Traditionally, possible target locations in a two dimensional region have been represented by two dimensional probability density functions.[4] In this paper, an alternative method referred to as an occupancy based map is investigated. This method represents the belief of target locations by discretizing the search domain into a finite number of cells. This map represents the state of the world in the sense that cells with high scores correspond to areas where the target is more likely to be located. This occupancy based map is shared and updated by all agents in the team. This work does not consider the communication issues involved with sharing a centralized occupancy map[5] but rather focuses on how it is used to coordinate the search using multiple agents.

The occupancy based map provides the framework for formulating the search strategy. The search strategy for a single vehicle is decomposed into three main steps. The first step involves propagating the world state (the occupancy map) forward in time. By providing a predictive aspect to the problem, each agent can then make control decisions based on the predicted future state of the world rather than only using the current information. Once the system generates a predicted future world state, each agent determines a desirable coordinate to visit in the future. The definition of "desirable" is formed as an numerical optimization problem. This formulation allows for each agent in the team to have a different set of parameters and therefore, each agent in the team can have its own notion of desirability. Finally, once this desirable coordinate is determined, a flight path which transitions the agent from its current location to the desirable coordinate is determined by solving another well-posed optimization problem over a convex set.

The search strategy is formulated with a heterogeneous team in mind. Each agent may have different capabilities and bandwidths. For example, fast agents (Figure 1(a)) are able to maneuver more effectively but require faster processing speeds. In contrast, slower agents (Figure 1(b)) are more sluggish but the bandwidth requirements decrease as well. The search strategy must be scalable and able to accommodate these differences.



(a) The GeoRanger autonomous air vehicle

(b) The SeaFox autonomous surface vehicle

Figure 1. Possible agents of heterogeneous team involved in searching mission.

Section II describes the occupancy based maps and their features. Section III illustrates the above mentioned search strategy in detail for a single agent. Section IV discusses how the single agent search strategy is extended to multiple, heterogeneous agents and also looks at the resulting team behavior and actions. Finally, Section V presents some conclusions and future directions of research with this work.

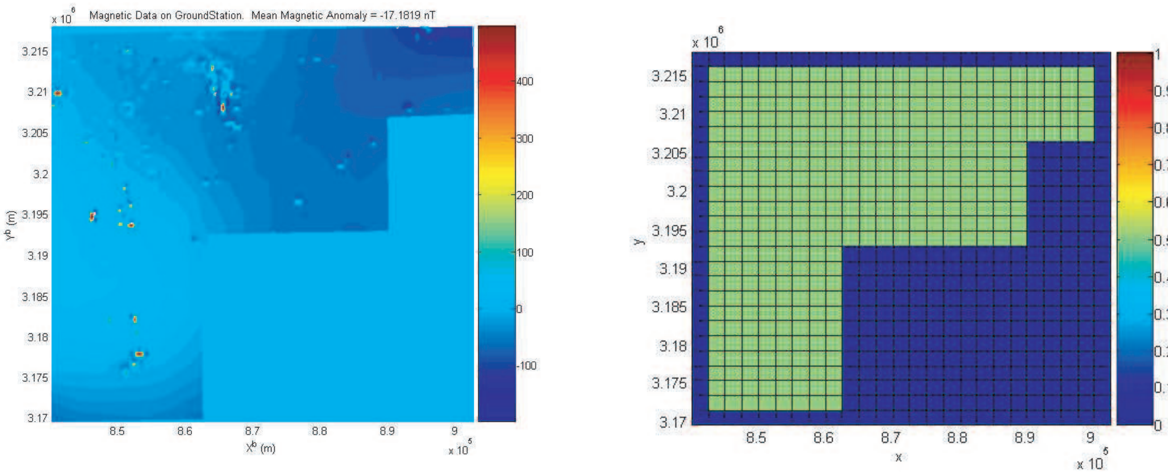
II. Occupancy Based Maps

In order to effectively search a two dimensional domain for a target, the system must keep track of state of the world in terms of possible target locations. To do this, an occupancy based map is employed. In this scheme, the search domain is discretized into rectangular cells. Each cell is assigned a score based on the probability that the target is located in that grid. This is similar to a two dimensional, discretized probability density function.[4] The spatial domain of the occupancy based map consists of a box where x is in between x_{min} and x_{max}. Similarly for the y dimension.

$$B = \left\{ \bar{z} \;\middle|\; \begin{array}{l} \bar{z}_1 \in [x_{min}, x_{max}] \\ \bar{z}_2 \in [y_{min}, y_{max}] \end{array} \right\} \tag{1}$$

The occupancy based map is a function defined over the set $B \times \Re \subset \Re^3$ which assigns a scalar in the range $[0, 1]$ to each element $\bar{z} \in B \subset \Re^2$ at a certain time step $k \in \Re$ ($x_w : \Re^3 \rightarrow \Re$).

This occupancy based map is shared and updated by all agents involved in the search. At each time step, guidance decisions for each agent are chosen based on this map. The state of the map at any time k is also referred to as the world state. This reflects the fact that the map represents the possible locations of target and other objects in the environment. In essence, the system's belief of the state of the world is embedded in the state of the occupancy based map. An example of an occupancy based map is shown below in Figure 2.



(a) Total magnetic intensity map of region to search

(b) Initial occupancy based map

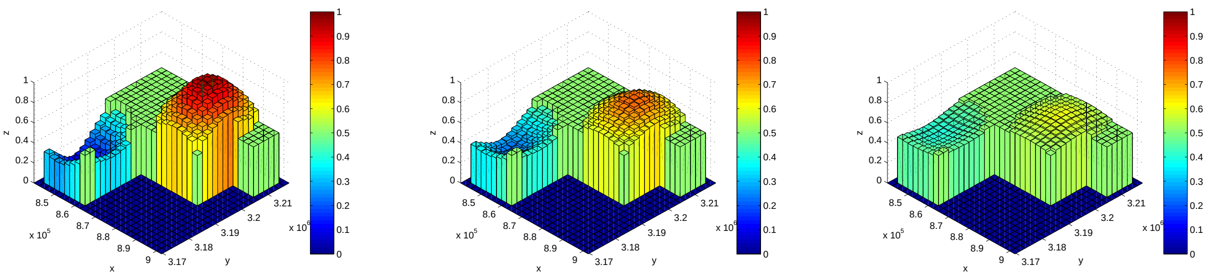
Figure 2. Discretization of search region into an occupancy based map.

In Figure 2(b), the blue sections represent cells with zero scores whereas the green represents scores of 0.5. This is the initial state of the occupancy based map. It represents the situation where no a priori knowledge of the targets location exists other than it cannot be in a section where no real data exists (sections of uniform blue in Figure 2(a)).

The occupancy based map is dynamic. It is constantly being updated at each time step. The agents are able to modify it to reflect their findings during the search mission. In addition, the function is time varying to model the fact that the target location estimate becomes more uncertain as time progresses. The time varying aspect is modeled as a simple linear dynamic model of the form

$$x_w(k+1,\bar{z}) = \tau x_w(k,\bar{z}) + (1-\tau)x_{w,nom} \quad \text{for } \bar{z} \in B \tag{2}$$

In Eq. 2, $x_{w,nom}$ is simply the nominal score and $\tau \in [0,1]$ is the time constant governing how fast the score decays back to the nominal. This effect is shown below in Figure 3. There is no estimate of target velocity so the regions of high probability do not translate in the x and y directions. However, as time progresses, the estimates return to their nominal values to model the phenomenon that old measurements cannot be relied upon to judge if the target is still located in a certain cell or not.



(a) $x_w(k,\bar{z})$ (b) $x_w(k+1,\bar{z})$ (c) $x_w(k+2,\bar{z})$

Figure 3. World estimates decaying back to the nominal score over time.

At this point, it becomes useful to define the cell set, $\tilde{B}$. This is all values $\bar{z}$ which correspond to the center of a cell in the occupancy based map.

$$\tilde{B} = \left\{ \bar{z} \middle| \bar{z} = \begin{pmatrix} x_{min} + L_x(i-1/2) \\ y_{min} + L_y(j-1/2) \end{pmatrix}, i = 1,2,\ldots,N_x, j = 1,2,\ldots,N_y \right\} \tag{3}$$

The occupancy based map and its associated features provides a versatile framework from which to build a searching algorithm.

III. Single Agent Search Strategy

This section concentrates on how a single agent is to find an optimal location to search using the occupancy based map. The goal for the agent is to attempt to converge on regions of high score (a high probability that the target is located there). Once an agent locates an anomaly, it should loiter there until a positive identification can be made. The overall flow of the search strategy for a single agent is shown below in Figure 4.

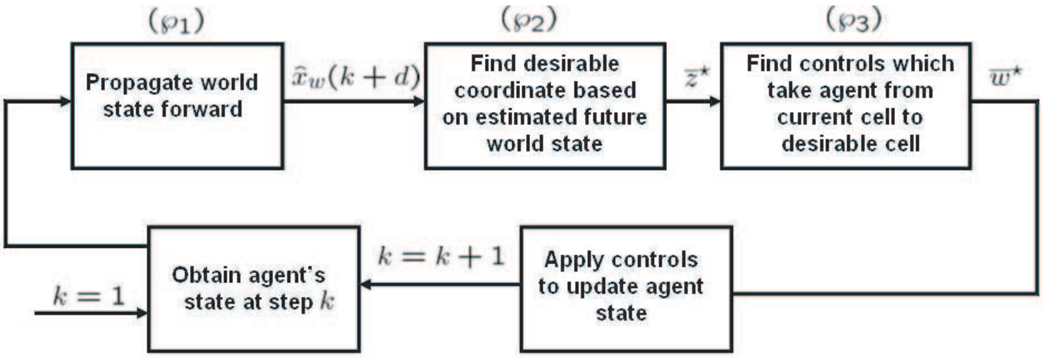


Figure 4. Flow diagram for single agent search strategy.

As can be seen, the process is comprised of three subproblems which are referred to as ($\wp_1$), ($\wp_2$), and ($\wp_3$). The process starts by finding the agent's state at the current time, $\overline{x}_{agt}(k)$. Next, ($\wp_1$) is solved to obtain the estimated world state at time $k + d$. Next, ($\wp_2$) is solved to find a desirable coordinate, $\overline{z}^\star$, which the agent will visit within the next d steps. Finally, ($\wp_3$) consists of finding an optimal set of waypoints/controls, $\overline{w}^\star$, which will take the agent from its current location to the location of the desirable coordinate found in ($\wp_2$).

A. ($\wp_1$) Predictive World Model

The first problem, ($\wp_1$), involves projecting the current state of the world forward in time to create an estimate of the world state at step $k + d$. The inputs are the estimated state of the target (position and velocity) at the current time step, $\hat{x}(k)$, and the current state of the occupancy based map, $x_w(k, \overline{z})$. In order to propagate an estimate of the target state, simple dynamics of the target are assumed

$$\hat{x}_{tgt}(k+1) = A_{tgt}\hat{x}_{tgt}(k) + B_{tgt}\hat{u}_{tgt}(k) \tag{4}$$

The world estimate at time $k + p$ is then a function of the estimated target state at time $k + p$ and the world state at the original time k.

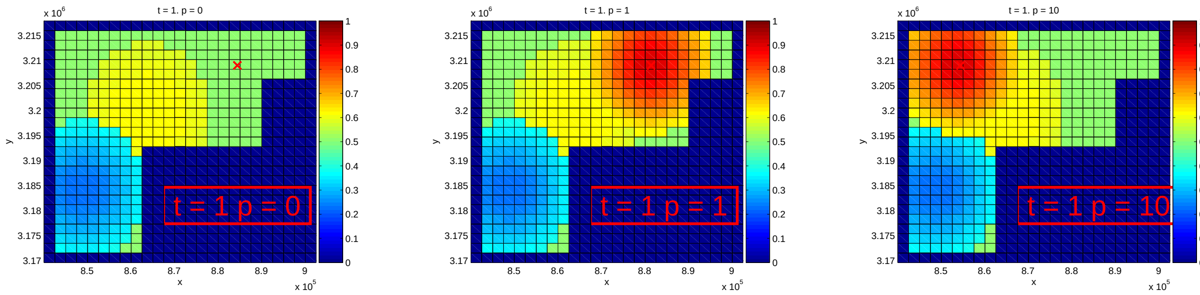
$$\hat{x}_w(k+p, \overline{z}) = h(\hat{x}_{tgt}(k+p), x_w(k, \overline{z})) \quad \text{for } p = 0, ..., d \tag{5}$$

In this example, the function $h()$ simply adds a two dimensional gaussian centered about $\hat{x}_{tgt}(k + d)$ to $x_w(k, \overline{z})$. An example of this is shown below in Figure 5 when the estimated target state is observed to be moving to the left at a constant velocity.

Now that the state of the world can be estimated at step $k + d$, the system attempts to find a coordinate which has desirable properties which is within the agent's reachable set. This is addressed in ($\wp_2$).

B. ($\wp_2$) Finding a Desirable Coordinate

In ($\wp_1$), the state of the world is propagated forward in time by d steps. The locations that the agent can reach in d steps are determined by parameters such as its maximum velocity and turn rate. The set of all locations that the agent can reach in d steps is referred to as the agent's reachable set, $Z \subset \Re^2$. The



(a) $\hat{x}_w(k, \overline{z}) = x_w(k, z)$ (b) $\hat{x}_w(k+1, \overline{z})$ (c) $\hat{x}_w(k+10, \overline{z})$

Figure 5. Estimated world states at different times for estimated target moving to the left.

subproblem ($\wp_2$) concerns finding a desirable coordinate (a desirable $\overline{z}$ value) for the agent to go to in d steps. The desirability of a coordinate is mathematically modeled as an objective function and the matter of determining the most desirable coordinate in the agent's reachable set is posed as an optimization problem.

$$(\wp_2) \ \underset{\overline{z} \in Z}{\text{maximize}} \ \ \hat{x}_w(k+d, \overline{z}) + \alpha \cdot f_\chi(\overline{z}) \tag{6}$$

The objective function is the sum of two functions. The first, $\hat{x}_w(k+d, \overline{z})$ is simply the estimated state of the world at step $k+d$. The second, $f_\chi(\overline{z})$ is a potential function which penalizes heading deviation and is mathematically expressed as shown in Eq. 7.

$$f_\chi(\overline{z}) = \eta\left(\chi_{uav}, 2\pi - tan^{-1}\left(\frac{\overline{z}_2 - y_{uav}}{\overline{z}_1 - x_{uav}}\right)\right) \tag{7}$$

In Eq. 7, the function $\eta(a, b)$ computes the absolute angular difference between the two angles, a and b. A simple absolute value of the difference of a and b is not sufficient and some simple heuristics are included in the function $\eta()$ to take care of situations such as where $a = 1$ degree and $b = 359$ degrees. A simple absolute value of the difference would return an angle of 358 degrees which is incorrect. However the function $\eta()$ returns the correct angular difference of 2 degrees. An example of this function for $\chi_{uav} = 45$ degrees is shown in Figure 6.

Maximizing $f_\chi(\overline{z})$ requires picking a location $\overline{z}$ which minimizes the course change required to visit this location ($\overline{z}$ locations which are in line with the blue arrow in Figure 6). Maximizing $\hat{x}_w(k+d, \overline{z})$ requires picking a location $\overline{z}$ which is in a cell with the highest score at the time step $k+d$. The parameter α is used to tradeoff course deviation and occupancy map score when solving the optimization problem. Small α values yield agents which will deem coordinates in high scoring cells as the most desirable regardless of the required course deviation. This may be appropriate for more agile agents such as small UAVs. Conversely, large α values correspond to agents which may deem a coordinate more desirable if it has a somewhat low score but is in line with its current course. This may be appropriate for cumbersome agents like large boats.

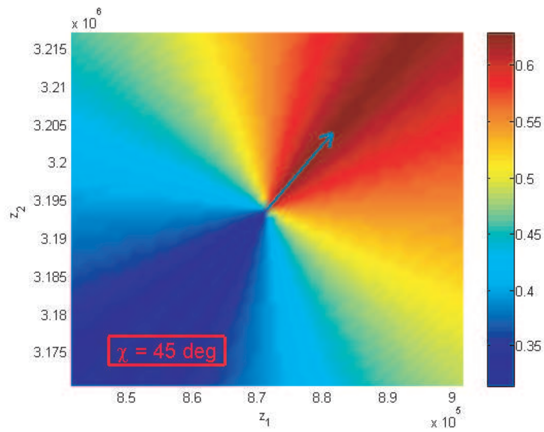


Figure 6. Potential function, $f_\chi(\overline{z})$ for $\chi = 45$ degrees.

($\wp_2$) is posed as a standard optimization problem but it is far from a well posed, convex program. Although the feasible set Z is typically a convex subset of $\Re^2$ (a perfect circle), the objective function is not as agreeable. The objective function in Eq. 6 is interesting for several reasons. First, the function is the combination of a numerical, discontinuous function (the world state, $\hat{x}_w()$) and a continuous, functional representation of a potential field (the course penalty, $f_\chi()$). This combination of numerical lookup table and analytical function creates a unique situation which requires a numerical algorithm to attempt to solve. A general method which provides a quasi-optimal solution to a general optimization problem over a specific set is now presented for solving this problem.

1. *Theory*

In order to find a desirable coordinate to move to, a method which mixes the ideas of probability collectives[6] and particle filters[7] is used. This provides a method to attempt to find a minimizer to the following general optimization problem.

$$(\wp) \text{ minimize } f_0(x) \text{ over } x \in X = \text{a box} \tag{8}$$

Recall that a box is defined by each element x_k of $x \in X \subseteq \Re^n$ being in a certain interval $I_k = [l_k, u_k]$.

$$X = \left\{ x \left| \begin{array}{c} x_1 \in I_1 = [l_1, u_1] \\ x_2 \in I_2 = [l_2, u_2] \\ \vdots \\ x_n \in I_n = [l_n, u_n] \end{array} \right. \right\} \tag{9}$$

The difficultly in solving Eq. 8 arises from the fact that the objective function may not be well behaved (i.e. non-convex, non-differentiable, etc.). This is especially true in ($\wp_2$). It may be difficult or impossible to find an optimal solution. An algorithm to find a quasi-optimal, feasible solution is now proposed.

1. Generate M particles (instances of $x \in X \subseteq \Re^n$) distributed over X in some fashion.
2. Assign weights to each particle based on its objective function value.
3. Resample the particles proportional to the weights.
4. Repeat step 2 and 3 until some stopping criterion is reached.

This algorithm can be seen as a type of genetic algorithm which employs adaptive sampling. Each particle represents a sample and each generation of particles is adaptively moved to locations where they matter the most.

2. *Initial Particle Distribution*

To find a quasi-optimal minimizer of $f_0(x)$, a finite set of possible minimizers are used. Each individual guess of a minimizer, $x^{[m]}(t)$ is called a particle and together the particles make up the particle set, $\chi(t)$.

$$\chi(t) = \bigcup_M x^{[m]}(t) = \left\{ x^{[1]}(t), x^{[2]}(t), \ldots, x^{[M]}(t) \right\} \tag{10}$$

To initialize the algorithm, it is necessary to assign actual values to the initial particle set. Since there is no a priori knowledge regarding the minimizer of $f_0(x)$, the initial distribution of the particles is chosen as a uniform distribution over the set X

$$x_k^{[m]}(0) = \text{rand}(l_k, u_k) \quad \text{for} \begin{array}{c} m = 1, \ldots, M \\ k = 1, \ldots, n \end{array} \tag{11}$$

3. *Assign Particle Weights*

A weight is now assigned to each particle based on its objective function value.

$$w^{[m]}(t) = -f_0(x^{[m]}(t)) \quad \text{for } m = 1, \ldots, M \tag{12}$$

Note that this assigns a higher weight to particles which yield a smaller objective function value.

4. Resample Particles

The next particle set is generated by first sampling from the current particle set proportional to the weights.

$$\tilde{x}^{[m]}(t) = g(\chi(t), w(t)) \quad \text{for } m = 1, \dots, M \tag{13}$$

Here, $g()$ is a sampling function which samples elements from the particle set, $\chi(t)$, proportional to the weights, $w(t)$. One popular method to do this is to use the roulette wheel method. In this method a roulette wheel with M slots is created. The weights are normalized so that they sum to 1. Each normalized weight then represents the angular percentage that this particle occupies on the roulette wheel. The wheel is spun and depending on where it lands, the corresponding particle $x^{[m]}(t) \in \chi(t)$ is selected as $\tilde{x}^{[m]}(t)$. This process is repeated M times.

As with many genetic algorithms, a mutation process must be performed when evolving one population to another. This is true here as well and the mutation operation is simply adding noise to each sample $\tilde{x}^{[m]}$. It is required that for each particle, $x^{[m]}(t) \in X \; \forall t$. Care must be taken so that the noise added does not "push the particle out of X". The noise must therefore be in the interval

$$n^{[m]}(t) \in [l - \tilde{x}^{[m]}(t), u - \tilde{x}^{[m]}(t)] \quad \text{for } m = 1, \dots, M \tag{14}$$

Finally, the new particle set is determined by

$$x^{[m]}(t+1) = \tilde{x}^{[m]}(t) + n^{[m]}(t) \quad \text{for } m = 1, \dots, M \tag{15}$$

This formulation guarantees that each particle $x^{[m]}(t) \in X \; \forall \, t$ (each particle represents a feasible solution to $(\wp)$). It has the feature that as this evolves from generation to generation, the particles with a higher weight (i.e. lower objective function value) are more likely to continue on to the next population.

5. Stopping Criterion

Step 2 and 3 are repeated until some stopping criterion is reached. An example stopping criterion could be: "terminate when the variance of the particles is reduced below some threshold". In this case, the process is simply repeated for T steps. The quasi-optimal minimizer is then computed from the average of the final particle set.

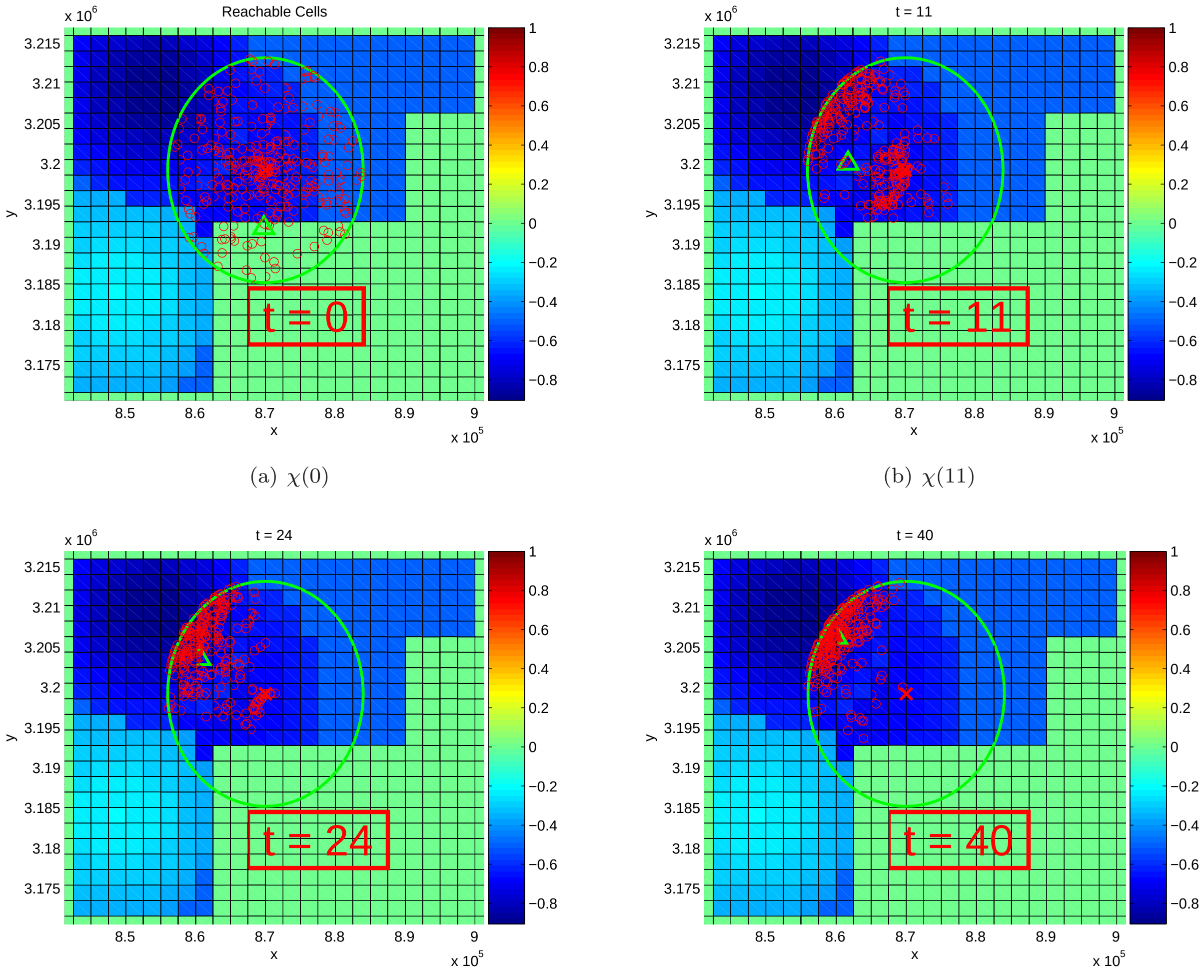
$$\bar{x}^{\star} = \frac{1}{M} \sum_{m=1}^{M} x^{[m]}(T) \tag{16}$$

6. Application to $(\wp_2)$

The above described method can be applied to the searching problem by parameterizing the reachable set Z using a radius and an angle.[8] By placing an upper and lower bound on the radius and angle, the set Z can be represented as a box. An example of applying the probability collective/particle filter method to solve $(\wp_2)$ is shown in Figure 7 for $\alpha = \epsilon$.

The individual particles are shown as red circles and the centroid of the particles is shown as a green triangle. After 40 iterations, the quasi-optimal minimizer settles near the true optimal solution. Note that it does not achieve the true optimal but it does achieve a feasible solution which is near the true optimal solution.

The above method is suitable for solving a general optimization problem over a box. Although $(\wp_2)$ can be solved using this method, it is computationally intensive due to the relatively large number of particles required and the necessity to evolve the population through several generations. The benefit of this method is that it is able to produce a quasi-optimal solution $\overline{z}^{\star} \in Z$. This method may be suitable for slower agents which may have more time to plan/compute between waypoints. But in light of the computational intensity of the above method, an alternative approximation is desired for faster agents which operate at a higher bandwidth. An approximation is to use an exhaustive search of the objective function over the set $\tilde{B} \bigcap Z$ (the occupancy based map cell center set intersected with the reachable set). This amounts to evaluating $\hat{x}_w(k+d, \overline{z}) + f_{\chi}(\overline{z})$ at the center of each cell inside in Z (i.e. inside the green circle in Figure 7) and choosing the argument which produces the largest objective function value as $\overline{z}^{\star}$. Although this method is



(a) $\chi(0)$ (b) $\chi(11)$

(c) $\chi(24)$ (d) $\chi(40)$

Figure 7. Progression of probability collective process. True minimum is located in upper left corner of reachable cells.

less computationally intensive, the resulting approximate solution, $\bar{z}^{\star}$ is at the center of an occupancy map cell. If the cells are large, this may not be desirable since the resolution is greatly reduced using this method.

C. ($\wp_3$) Convex Formulation

The final subproblem, ($\wp_3$), concerns finding feasible waypoints which take the agent from the current location, $\bar{z}_0$, to the desirable cell location found in ($\wp_2$). This is formed as a convex optimization problem. From an optimization standpoint, the corresponding decision vector $\bar{w} \in \Re^{2\cdot d}$ is

$$\bar{w} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ \vdots \\ w_{2\cdot d-1} \\ w_{2\cdot d} \end{pmatrix} = \begin{pmatrix} \bar{z}_1(k+1) \\ \bar{z}_2(k+1) \\ \bar{z}_1(k+2) \\ \bar{z}_2(k+2) \\ \vdots \\ \bar{z}_1(k+d) \\ \bar{z}_2(k+d) \end{pmatrix} \tag{17}$$

The goal is to move the agent from its current position to the desired position $\bar{z}^{\star}$, so the objective function for $(\wp_3)$ is formulated as

$$f_0(\overline{w}) := \sum_{t=1}^{d} ||\bar{z}(k+t) - \bar{z}^{\star}||_2^2 \tag{18}$$

In Eq. 18, each coordinate $\bar{z}(k)$ is a waypoint which dictates where the agent should be located at time k. Obviously, the minimum of this function is zero. This corresponds to all the waypoints being placed at the desirable cell located at $\bar{z}^{\star}$. However this in general would not be feasible. The agent can only travel a distance $r_{max} = \Delta T \cdot V_{max}$ in a single step. So in order for the waypoints to be feasible, it becomes necessary to introduce constraints of the form

$$f_1(\overline{w}) := ||\overline{z}(1) - \overline{z}_0||_2^2 - r_{max}^2 \leq 0 \tag{19}$$

$$f_i(\overline{w}) := ||\overline{z}(i) - \overline{z}(i-1)||_2^2 - r_{max}^2 \leq 0 \text{ for } i = 2, \ldots, d \tag{20}$$

Physically, these constraints enforce that each waypoint must be within a distance r_{max} of the previous waypoint. This ensures that flight path generated is a feasible one. The problem can now be formally stated as

$$\begin{aligned} (\wp_3) \text{ minimize } & f_0(\overline{w}) \text{ over } \overline{w} \in \Re^{2\cdot d} \\ \text{subject to } & f_i(\overline{w}) \leq 0 \text{ for } i = 1, \ldots, d \end{aligned} \tag{21}$$

In order to analyze what type of optimization problem this is, a closer look at the objective function and constraint functions is required. The objective function can be written as

$$f_0(\overline{w}) = \frac{1}{2}\overline{w}^T H \overline{w} + f^T \overline{w} + r \tag{22}$$

$$\begin{aligned} \text{where} \quad H &= 2I_{d\times d} \\ f^T &= 2\begin{pmatrix} \bar{z}_1^{\star} & \bar{z}_2^{\star} & \bar{z}_1^{\star} & \bar{z}_2^{\star} & \ldots \end{pmatrix} \\ r &= d||\bar{z}^{\star}||_2^2 \end{aligned}$$

This is a strictly convex function since it is in a quadratic form and the Hessian is equal to H which is positive definite (all eigenvalues are equal to 2).

The constraint functions can be analyzed in a similar fashion. The first constraint $f_1(\overline{w})$ can be written as

$$f_1(\overline{w}) = \frac{1}{2}\overline{w}^T H_1 \overline{w} + f_1^T \overline{w} + r_1 \tag{23}$$

$$\begin{aligned} \text{where} \quad H_1 &= \text{diag}(2I_{2\times 2}, 0_{d-2\times d-2}) \\ f_1^T &= \begin{pmatrix} -2\overline{z}_{1,0} & -2\overline{z}_{2,0} & 0 & \ldots & 0 \end{pmatrix} \\ r_1 &= -r_{max}^2 + ||\overline{z}_0||_2^2 \end{aligned}$$

And the constraint functions $f_2(\overline{w})$ through $f_d(\overline{w})$ can be written as

$$f_i(\overline{w}) = \frac{1}{2}\overline{w}^T H_i \overline{w} + f_i^T \overline{w} + r_i \text{ for } i = 2, \ldots, d \tag{24}$$

$$\begin{aligned} \text{where} \quad H_i &= \text{diag}(N_i, \text{diag}(A, M_i)) \\ f_i^T &= \begin{pmatrix} 0 & \ldots & 0 \end{pmatrix} \\ r_i &= -r_{max}^2 \\ N_i &= \text{zeros}(2(i-2)) \\ M_i &= \text{zeros}(2d - 4 - 2(i-2)) \\ A &= \begin{pmatrix} 2 & 0 & -2 & 0 \\ 0 & 2 & 0 & -2 \\ -2 & 0 & 2 & 0 \\ 0 & -2 & 0 & 2 \end{pmatrix} \end{aligned}$$

In Eq. 23 and 24, $\text{diag}(x, y)$ represents a block diagonal matrix with submatrix x in the upper left block and submatrix y in the lower right corner. Similarly, $\text{zeros}(p)$ represents a square zero matrix of size $p \times p$. Although the formulation appears complicated, note that only H_i is changes with each i. The formulation just describes that H_2 is a block diagonal matrix with A in the upper left corner and zeros elsewhere. H_3 is a block diagonal matrix where the submatrix A moves two columns to the right and two rows down. This process of moving the A matrix by 2 rows and columns with each i is described by the N_i and M_i submatrices.

One can now see that the constraint functions $f_i(\overline{w})$ for $i = 1, \ldots, d$ are convex functions because they are in quadratic forms and their respective Hessians are all positive semi-definite.

Therefore, $(\wp_3)$ consists of a strictly convex objective function over a convex set, so this is a convex programming problem. It can also be shown that it is well posed and the feasible set is non-empty, so a unique optimal solution exists. In a similar fashion to the previous two problems, $(\wp_3)$ can be packaged nicely into a system with inputs of the vehicle capabilities and desirable coordinate and then process the optimization problem to obtain an optimal set of waypoints or controls to take the agent to the desired cell. An example of the solution for the example situation (with $d = 10$) is shown in Figure 8.

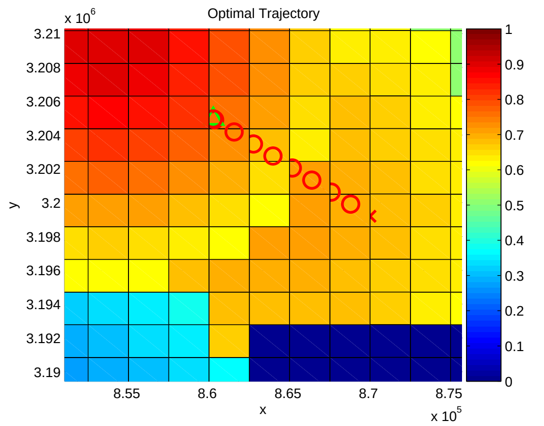


Figure 8. Optimal solution $\overline{w}^{\star}$ to $(\wp_3)$ zoomed into area of interest with $d = 10$

In Figure 8, the red x represents the current location of the agent, the green triangle is the desired destination $\overline{z}^{\star}$ from $(\wp_2)$, and the red circles represent the optimal waypoints $\overline{w}^{\star}$. Notice that although $d = 10$, there are only 8 visible waypoints. This is because waypoints 8, 9, and 10 are overlapping and all are equal to desired destination ($\overline{z}_8 = \overline{z}_9 = \overline{z}_{10} = \overline{z}^{\star}$). Furthermore, constraints $f_1(\overline{w})$ through $f_7(\overline{w})$ are active. This shows that the formulation of the objective function yields waypoints which place the agent at the optimal solution in the shortest possible time. This is the desired behavior during a searching type application where it is desired that the agent location and verify the target as soon as possible.

D. Updating Occupancy Map

Once the controls for each agent are assigned, it becomes necessary to update the score of each cell based on the agent's findings. As the agent finishes searching a cell, if no anomaly is discovered, the score of the cell can be updated using the agent's sensor model. If an anomaly is present, the probability that the agent detects this anomaly is given by $p(D(k)|\overline{x}_{agt}(k), \overline{x}_{tgt}(k))$. Therefore, the probability that the agent will "miss" detection is simply the complement.[4]

$$p(\overline{D}(k)|\overline{x}_{agt}(k), \overline{x}_{tgt}(k)) = 1 - p(D(k)|\overline{x}_{agt}(k), \overline{x}_{tgt}(k)) \tag{25}$$

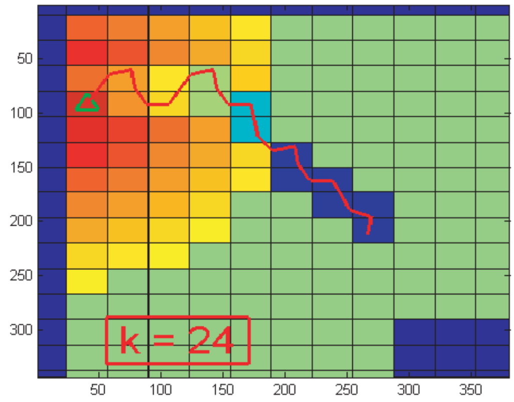
The score of each cell in the event of not detecting an anomaly is now updated using

$$x_w(k+1, \overline{z}) = p(\overline{D}(k)|\overline{x}_{agt}(k), \overline{x}_{tgt}(k))x_w(k, \overline{z}) \tag{26}$$

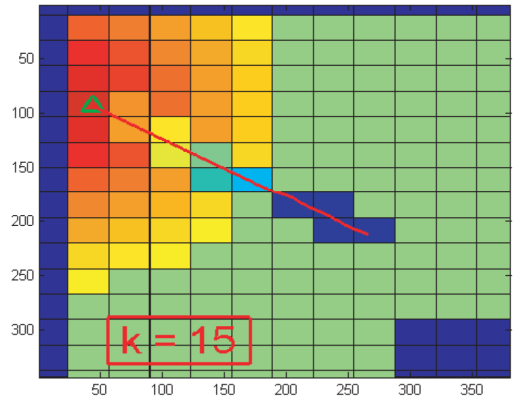
E. Scenarios

This strategy benefits the agent in several ways. The most obvious is that the agent is choosing control actions which will benefit it d steps in the future rather than "only thinking one step ahead". This effect is illustrated below in Figure 9.

At step $k = 0$, the system receives an estimate of the target state, $\hat{x}_{tgt}(0)$. This notifies it that the target is directly north of the agent's current position and is moving west with a constant velocity. The trajectory in Figure 9(a) is generated when the agent employs a greedy algorithm where at each time step ($k = 0, 1, \ldots, 24$)) the agent chooses a control which transitions it to the most desirable coordinate that it can reach in 1 step. The result is that the agent is constantly trying to "catch up" up with target. In contrast, the trajectory in Figure 9(b) is generated when the agent employs the above described search strategy with



(a) Using greedy algorithm



(b) Using single agent search strategy

Figure 9. Flight paths generated using greedy algorithm and single agent search strategy.

a look ahead window of $d = 10$. The agent is now choosing control actions which will benefit it in 10 steps. This yields to much smoother and faster convergence with the target.

IV. Multi-Agent Teams

A. Heterogeneity and Team Coupling

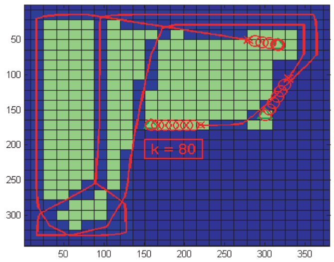
The single agent search strategy can easily be extended to a team situation. This involves each agent in the team using the above outlined algorithm. The parameters for each team member can be tailored to match the agents capabilities. The method is easily scalable to any number of agents.

The method is not completely decentralized because each agent in the team must have access to the centralized occupancy map. As each agent performs the search, it must be able to both read from and write to the map. Therefore, although the agents do not have explicit knowledge about each other, they are coupled though the centralized occupancy map. Using these simple set of rules, the emergent behavior of the overall system is that the team operates together in a coordinated fashion when searching for targets.

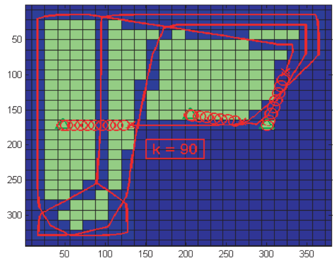
B. Results

A simulated search mission using the above framework and strategy is shown in Figure 10. In Figure 10, the agents are represented by the red circles. They all start the search from the same location with headings evenly spaced out. The solution to ($\wp_3$) is shown as the series of red circles. These show the planned waypoints for the next d steps which the agent is following. The target is located near the center of the map. As they begin searching, they all update the centralized occupancy map depending on whether they encounter the target or not. Initially, none of the agents encounters the target so they all discount the score of the cells that they have searched (cells turn blue). At time $k = 104$, one of the agents encounters an anomaly. It is very confident that this is target so the scores of the surrounding cells are increased. Figure 10(d) shows that due to the positive target encounter by one agent, the other agents immediately change course to converge on the target and help with the target identification and tracking process. In Figure 10(f) the locations around the target become a darker shade or red as the other two agents arrive at the cell and also make positive identifications.

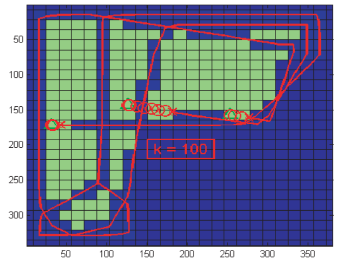
In a second scenario shown in Figure 11. In this scenario, agent 1 encounters and anomaly which it determines may be the target with only moderate confidence (Figure 11(b)). At this point, it stops and loiters in an attempt to make a more positive identification. In Figure 11(c), the agent 2 arrives at the location to assist with the target identification process. Agent 2 determines that the target is no longer at the same cell that the agent 1 identified and begins to search the surrounding cells for the target. Note that the agent 3 continues to search other areas and does not converge on the same spot. In Figure 11(d),



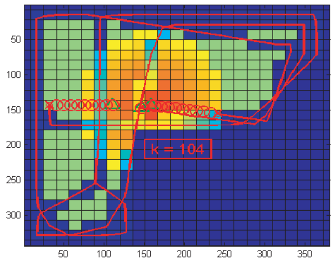
(a) $x_w(80, z)$



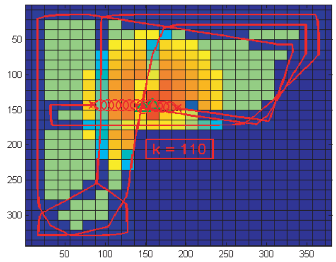
(b) $x_w(90, z)$



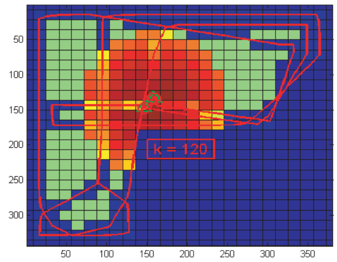
(c) $x_w(100, z)$



(d) $x_w(104, z)$



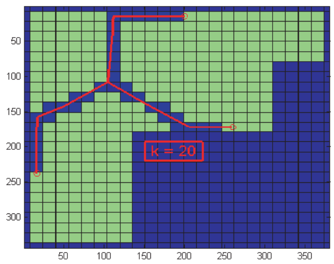
(e) $x_w(110, z)$



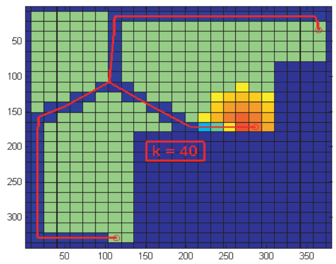
(f) $x_w(120, z)$

Figure 10. Multi-agent search for target: scenario 1.

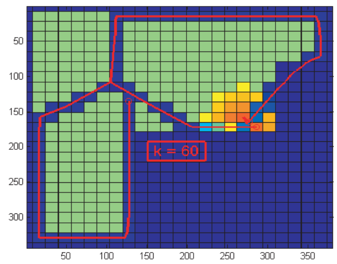
agent 3 encounters the target and increases the scores of the map near its location. In Figure 11(e) agent 2 finishes searching the nearby cells of the false positive identification and decides to assist agent 3. Finally, in Figure 11(f), the agent 2 arrives and makes a positive identification at the location of agent 3 (the scores are increases further to reflect two positive identifications instead of just one).



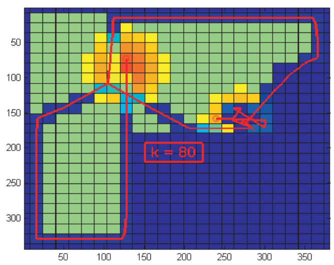
(a) $x_w(20, z)$



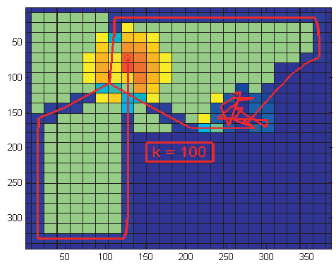
(b) $x_w(40, z)$



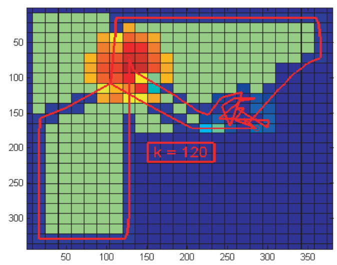
(c) $x_w(60, z)$



(d) $x_w(80, z)$



(e) $x_w(100, z)$



(f) $x_w(120, z)$

Figure 11. Multi-agent search for target: scenario 2.

V. Conclusion and Further Research

Increasing performance in a searching mission typically involves increasing the number of agents involved in the search. The main challenge with current manned systems is that increasing the number of agents greatly increases operator workload required to manage the team. If the team is comprised of heterogeneous agents with different capabilities, the mission management task becomes even more complicated. This paper presents a scalable searching algorithm that can be used to coordinating large numbers of heterogeneous agents involved in a searching mission. The centralized occupancy based map represents the system's belief of the state of the world. The map can be propagated forward in time to provide the estimate of the future state of the world at step $k + d$. Each agent then decides which coordinate is the most desirable to search in the next d steps. The team can be comprised of different types of agents with different capabilities. The formulation allows each agent to determine what is desirable for its individual capabilities. Each agent then computes control decisions based on the predicted future state of the world. Although these actions may not be optimal in a single step, they will benefit the agent in the future.

Although there is no explicit cooperation between agents in the team, the agents are implicitly coupled through the centralized occupancy map. The algorithm remains scalable because each agent does not need to explicitly know about the existence of other agents. Each agent executes the searching algorithm and the emergent behavior is that team performs a coordinated search.

The simulated scenario was the search for a submarine in a marine environment but the framework developed here is readily extendable to other situations. One alternative example is the detection and tracking of radiological/biological clouds. The main goal of this type of work would be to use unmanned systems in dirty and dangerous missions to track these hazards. The location of possible clouds could easily be represented using the occupancy based maps. As the clouds drift due to wind, the predictive world model would allow for agents to be able to quickly converge on possibly moving clouds.

Future work in this area is directed towards tailoring the cost function which governs the agent's actual flight path (Eq. 18). By only penalizing the terminal state, the current formulation yields flight paths which are straight lines from the current waypoint to the waypoint of the desirable coordinate. In some situations, this yields paths which traverse areas of low scores (Figure 10(e)). The agent does not acquire any new information by constantly traversing these areas of low scores. A more useful flight path would be one which traverses areas of high scores as the agent moves to the end location. This would allow the agent to search more efficiently as it executes its flight path. Current research is directed towards applying evolutionary computation methods[9] to solve this numerical path planning problem.

In addition, the method of updating the cells scores can be improved for the scenario where there is only a single target. Currently, the scores of the cells are not coupled. In a true, two dimensional probability density function, all of the discrete probabilities must sum to unity. The world state could be made as a true probability density function by coupling the cell scores and ensuring that they all summed to unity at all times. This would create the effect where an agent would search a cell and decrease its score, but then scores in the neighborhood would increase to preserve the unity sum. Although this complicates the score update process (since the score of one cell is now coupled to the scores of surrounding cells), this can lead to a more dynamic and effective search strategy for the individual agent and team in general.

VI. Acknowledgements

This work is sponsored in part by the Washington Technology Center (WTC) under grants F04-MC2 and F05-MC3. Juris Vagners at the University of Washington and Matt Wheeler and Brad Schrick at The Insitu Group also contributed to this work.

References

[1] Bourgault, F. and Durrant-Whyte, H. F., "Communication in General Decentralized Filters and the Coordinated Search Strategy," *Proceedings of the 7th International Conference on Information Fusion*, Australian Centre for Field Robotics, Stockholm, Sweden, 2004.

[2] Flint, M., Polycarpou, M., and Fernandez-Gaucherand, E., "Cooperative Control for Multiple Autonomous UAV's Searching for Targets," *Proceedings of the 41st IEEE Conference on Decision and Control*, University of Cincinnati, Las Vegas, NV, 2004.

[3] Lum, C. W., Rysdyk, R. T., and Pongpunwattana, A., "Autonomous Airborne Geomagnetic Surveying and Target Iden-

tification," *Proceedings of the 2005 Infotech@Aerospace Conference*, Autonomous Flight Systems Laboratory, Arlington, VA, September 2005.

[4]Bourgault, F., Furukawa, T., and Durrant-Whyte, H., "Coordinated Decentralized Search for a Lost Target in a Bayesian World," *Proceedings of the 2003 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, Australian Centre for Field Robotics, Las Vegas, NV, October 2003.

[5]Polycarpou, M. M., Yang, Y., and Passino, K. M., "A Cooperative Search Framework for Distributed Agents," *Proceedings of the 2001 IEEE International Symposium on Intelligent Control*, Mexico City, Mexico, 2001.

[6]Huang, C.-F., Wolpert, D. H., Bieniawski, S., and Strauss, C. E., "A Comparative Study of Probability Collectives Based Multi-agent Systems and Genetic Algorithms," *Proceedings of the GECCO 2005 Conference*, 2005.

[7]Fox, D., Hightower, J., Liao, L., and Schulz, D., "Bayesian Filtering for Location Estimation," *IEEE Pervasive Computing*, July 2003, pp. 23–33.

[8]Lum, C. W., "A Single Agent Search of a Two Dimensional Space using Probability Collectives and Convex Optimization," Tech. rep., University of Washington, Seattle, WA, 2006.

[9]Pongpunwattana, A. and Rysdyk, R. T., "Real-Time Planning for Multiple Autonomous Vehicles in Dynamic Uncertain Environments," *AIAA Journal of Aerospace Computing, Information, and Communication*, December 2004, pp. 580–604.