2007-5

Probabilistic Searching Using a Small Unmanned Aerial Vehicle

Steven R. Hansen

Timothy W. McLain
Brigham Young University - Provo, mclain@byu.edu

See next page for additional authors

Follow this and additional works at: https://scholarsarchive.byu.edu/facpub

Part of the Computer Sciences Commons, and the Mechanical Engineering Commons

Original Publication Citation

BYU ScholarsArchive Citation
Hansen, Steven R.; McLain, Timothy W.; and Goodrich, Michael A., "Probabilistic Searching Using a Small Unmanned Aerial Vehicle" (2007). All Faculty Publications. 1509.
https://scholarsarchive.byu.edu/facpub/1509

This Peer-Reviewed Article is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Faculty Publications by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.
Probabilistic Searching Using a Small Unmanned Aerial Vehicle

Steven R. Hansen  Timothy W. McLain*  Michael A. Goodrich
Brigham Young University Provo, UT 84602

Ground breaking concepts in optimal search theory were developed during World War II by the U.S. Navy. These concepts use an assumed detection model to calculate a detection probability rate and an optimal search allocation. Although this theory is useful in determining when and where search effort should be applied, it offers little guidance for the planning of search paths. This paper explains how search theory can be applied to path planning for an SUAV with a fixed CCD camera. Three search strategies are developed: greedy search, contour search, and composite search. In addition, the concepts of search efficiency and search completeness are offered as metrics for search effectiveness. Simulation results comparing the effectiveness of the strategies are presented.

Nomenclature

\[ \delta_g \] node spacing in the probability grid
\[ \zeta(t^*) \] search completeness at \( t^* \)
\[ \xi(t^*) \] search efficiency at \( t^* \)
\[ \theta_o \] observation angle, between the observation direction and straight down
\[ \Lambda \] area detection probability
\[ \psi_c \] commanded heading
\[ \psi_g \] the current greedy heading
\[ \psi_{g,last} \] the last greedy heading
\[ \psi_{p,\text{max}} \] test angle, used to determine the benefit of a given direction
\[ \psi_{p}^{*} \] the \( \psi_{p} \) corresponding to the maximum \( \bar{p}_{\psi_{p}} \)
\[ A_k \] area represented by node group \( G_k \)
\[ A_{NU} \] the non-uniform probability region of the variable map
\[ A_{obj} \] cross-sectional area of a target object in \( m^2 \)
\[ A_p \] the projected area of one pixel onto a plane one meter away, in \( m^2 \)
\[ A_T \] the total area in the variable map
\[ A_U \] the uniform probability region of the variable map
\[ C_{\bar{d}} \] denotes that the target is contained in the area where detection occurs
\[ C_k \] denotes that the target is contained within \( A_k \)
\[ d \] absolute distance
\[ D \] target detection event
\[ d_G \] threshold distance from a UAV to a node group, used in group assignment
\[ d_{jk} \] closest distance from \( u_j \) to \( G_k \)
\[ D(t) \] target detection by time \( t \)
\[ \dot{D}(t) \] target detection before time \( t \)
\[ D^*(t) \] target detection at time \( t \)
\[ d_{jk} \] distance from UAV \( u_j \) to group \( G_k \)
\[ d_{un} \] distance of some other UAV, \( u_n \), to point \( q_{\psi_{p}} \)
\[ f \] focal length in meters
\[ f_{p} \] focal length, measured in average pixels (average between \( f_x \) and \( f_y \))

*Corresponding author, email: mclain@byu.edu
\( f_x \) focal length, measured in pixel widths
\( f_y \) focal length, measured in pixel heights
\( G_k \) node group \( k \)
\( h \) observer altitude
\( I(*) \) instantaneous glimpse of a target at *
\( k_1, k_2 \) radial distortion coefficients
\( k_3, k_4 \) radial distortion approximation coefficients
\( k_s \) image skew constant
\( k_{un} \) probability scale factor based on proximity to other UAVs
\( L_{NU} \) edge length for the non-uniform area of the variable map
\( L_s \) length of the sliding segment
\( L_T \) edge length for the total area of the variable map
\( M_{CC} \) connected component detection model
\( N \) the total number of nodes
\( N \dot{\dot{N}} \) node search rate
\( N_{\psi p} \) number of selected nodes around point \( q_{\psi p} \)
\( N_B \) total number of nodes contained in all bounded groups
\( N_C \) node threshold in the contour search
\( N_G \) node threshold in the greedy search
\( N_k \) the number of nodes in group \( G_k \)
\( N_{UAV} \) the total number of UAVs
\( p \) true image pixel (distortion removed)
\( \hat{p} \) an image pixel (appears distorted in original image)
\( \hat{p}_{jk} \) a probability rate estimate for \( u_j \) searching \( G_k \)
\( p(*) \) probability of *
\( \bar{p}_{\psi p} \) average probability in the vicinity of point \( q_{\psi p} \)
\( \bar{p}_{\psi p} \) some \( \bar{p}_{\psi p} \) for a UAV, modified by its proximity to every other UAV
\( P_s \) image area occupied by the target projection, in pixels
\( P_T \) probability threshold, used to determine node groups
\( q_{\psi p} \) a point at distance \( R_p \) from the UAV and angle \( psi_p \) relative to \( X_S \)
\( r_m \) distance from the image center in meters (at \( Z_C = 1 \) m)
\( \hat{r}_m \) distorted distance from the image center in meters (at \( Z_C = 1 \) m)
\( R_{NT} \) the ratio of non-uniform area to total area in the variable map
\( r_{obj} \) radius of a spherical target in meters
\( r_{objp} \) projected radius of a spherical target in pixels
\( r_p \) distance of a given pixel from the target center, in pixels
\( R_p \) range from the UAV, used in finding local probable benefits
\( s_1, s_2 \) CCD camera detection probability model parameters
\( t \) time
\( t^* \) time at the end of a search
\( t_A \) time between node group assignments
\( t_C \) time threshold for the contour search
\( t_G \) time threshold for the greedy search
\( U \) the target remaining un-detected
\( u_j \) UAV number \( j \)
\( V_g \) groundspeed
\( W \) search width
\( x \) lateral range
\( (x_0, y_0) \) image center point in the pixel frame
\( X, Y, Z \) axes of the world coordinate frame
\( X_S, Y_S, Z_S \) axes of the UAV stability coordinate frame
\( (x_{jk}, y_{jk}, z_{jk}) \) relative vector from \( u_j \) to the nearest point of \( G_k \)
\( (x_{tp}, y_{tp}) \) target point in the pixel frame
I. Introduction

This paper addresses the technical and theoretical issues necessary to apply fundamental search principles to the problem of planning effective search paths for small unmanned aerial vehicles (SUAVs, 3 to 5 foot wingspan). The motivation for this work comes primarily from wilderness search and rescue (WSAR) missions. Time is critical in WSAR—the likelihood that a lost victim will survive decreases with each passing hour, and the effective search radius increases by approximately 3 km per hour.\(^{1,2}\) In addition, the increasing capability of hand-launched SUAVs\(^{3}\) prompts research towards more complex mission objectives. Our hypothesis is that concepts in optimal search theory can be utilized by SUAVs to increase their search effectiveness.

These concepts have been the focus of several military projects since World War II. The book Search and Screening, General Principles with Historical Applications\(^{4}\) describes fundamental concepts in optimal search theory, and represents the combined efforts of many collaborators. Theirs was the task of increasing the effectiveness of U.S. Navy aerial search during World War II. Other subsequent military projects have continued work on probabilistic searching using this theory as their foundation.\(^{5}\)

Other work in probabilistic search is presented by Tang\(^{6}\) and Flint et al.\(^{7}\) Tang offers an approach to indicating the probable location of a target using Monte-Carlo simulations and a hospitality map. Both of these references indicate the benefit of using some probability map to guide the search. However, both references assume an overly-simplified sensor footprint that is either circular or rectangular around the agent, and both could benefit from a comparison of their approach to an optimal search allocation.

This paper follows the approach used in optimal search theory of (1) identifying an instantaneous detection probability based on target distance, (2) calculating the search width of the agent, and (3) determining the optimal search allocation for a given probability map. A brief introduction to the detection model is given along with details showing its application. A detailed derivation of the detection probability model is found in Ref. 8. This paper offers (1) a method of using a discrete probability grid to measure search rate, (2) definitions of search efficiency and search completeness, (3) three path planning approaches for search, and (4) a demonstration of the evaluation of search path plan effectiveness using the given tools.

II. CCD Camera Target Detection Model

This section is a brief introduction to a detection model that is more fully described elsewhere.\(^{8}\) The detection model depends on well-lit conditions and describes the probability that an unobstructed spherical target is detected in a single image taken from a charge-coupled device (CCD) camera such as the one shown in Figure 1. The image is analyzed by an algorithm that identifies possible target pixels based on pre-defined color limits in terms of hue, saturation and value. If a connected group of possible target pixels has at least 20 pixels in it, then it is considered to be a target. The threshold of 20 pixels was chosen based on experience to distinguish potential targets from image noise. Based on this detection method, Monte Carlo simulations were used to create a detection model of the form

\[
p(D|I(d, r_m), M_{CC}) = \max \left( \begin{array}{c} 1 - \exp \left( -\frac{(k_r(r_m)f_p)}{d} A_{obj} - s_1 \right) \\ 1 + \exp \left( -\frac{(k_r(r_m)f_p)}{d} A_{obj} - s_2 \right) \end{array} \right) \right), \quad (1)
\]

The term \(p(D)\) indicates the probability of a detection event, \(I(d, r_m)\) denotes an instantaneous look from distance \(d\) with the target appearing at radius \(r_m\) from \((x_0, y_0)\) in the image, and \(M_{CC}\) denotes that the probability is based on the connected component detection model. The constants in this model \((a_1 = 9.0, a_2 = 2.8, s_1 = 26, s_2 = 34)\) are dependent on the difference in color between the target and its background. The model is also based on a camera (640 × 480 pixels) that produces the following parameters in calibration: \(f_x = 503.652, f_y = 500.110, x_0 = 319.5, y_0 = 239.5, k_3 = 1.026, \) and \(k_4 = -0.217.\) The focal length \(f_p,\)

Figure 1. SUAV CCD camera.
measured in pixels, is an average between \( f_x \) and \( f_y \). The image skew constant, \( k_s \), is dependent on the radial distance of the target from the image center according to

\[
k_s(\bar{r}_m) = \frac{k_3}{2} + \sqrt{\left(\frac{k_3}{2}\right)^2 + k_4\bar{r}_m},
\]

(2)

For a numerical example, assume that a target has a cross-sectional area of \( A_{obj} = 1 \text{ m}^2 \), and is a distance of \( d = 80 \text{ m} \) away. Also assume that the target center appears at pixel \((x_{tp}, y_{tp}) = (500, 300)\) in the image. In this case the radial distance from the image center is

\[
\bar{r}_m = \sqrt{\left(\frac{500 - 319.5}{503.652}\right)^2 + \left(\frac{300 - 239.5}{500.110}\right)^2} = 0.378,
\]

which yields \( k_s = 0.939 \) and \( p(D|I(d, \bar{r}_m), M_{CC}) = 0.346 \). Thus, even though the target is unobstructed and in view, there is only a 34.6% chance of detecting it in the image. If no image skew is present \( (k_s = 1.0) \), then the instantaneous detection probability with distance is as shown in Figure 2. This model is used throughout the paper, along with the numerical constants given above. It predicts a target’s detection probability based on its distance from the camera and its position in the image.

![Figure 2. Instantaneous detection probability as a function of distance.](image)

III. Simulation Environment

This section presents a practical method of recording the detection probabilities as they accrue. This data can then be used to approximate an agent’s search width, as well as to measure UAV search performance in a given scenario. This data can also be used to direct UAV movement in real-time as the information is updated.

A. The Probability Grid

One method for planning a search and measuring its effectiveness is discrete search. Particularly, it is helpful to generate a probability grid containing \( N \) discrete nodes to approximate the probability distribution in the search area. Grid nodes are spaced some distance \( \delta_y \) apart in the world \( x \) and \( y \) directions over the region of interest. Each node, \( n_i \), represents an area of \( \delta_y^2 \), and records two probabilities:

1. \( p(C_i) \), the probability that the target is contained in area \( A_i \) (given by the initial probability map).
2. \( p(D(t)|C_i) \), the probability that the target is detected by time \( t \), given that it is contained in area \( A_i \).

At any given time in the search, the overall probability of successful detection can be calculated as

\[
p(D(t) \cap C_0) = \sum_{i=1}^{N} p(C_i)p(D(t)|C_i),
\]

(3)
where \( C_i \) indicates that the target is contained within the area where detection occurs. If an observer is allowed to view node \( n_i \) at time \( t \), then an update can be made to describe the new detection probability for a target in \( A_i \). The detection probability becomes

\[
p(D(t)|C_i) = p(\tilde{D}(t)|C_i) + p(D^*(t)|C_i) - p(D(t)|C_i) p(D^*(t)|C_i)
\]

where

\[
D(t) = \text{target detection by time } t,
\]
\[
\tilde{D}(t) = \text{target detection before time } t, \text{ and}
\]
\[
D^*(t) = \text{target detection at time } t.
\]

If the probability grid updates are made in this manner, then a quantity may be calculated called the area detection probability (ADP), \( \Lambda \). This is the average probability of successful target detection in an area, multiplied by that area, or

\[
\Lambda = \frac{\sum_{i=1}^{N} p(C_i) p(D(t)|C_i) A_N}{N} = \delta^2 \frac{\sum_{i=1}^{N} p(C_i) p(D(t)|C_i)}{N}.
\]

Here \( A_N \) denotes the area represented by the \( N \) nodes being considered. Taken over the course of a search, the area detection probability is a measurement of not only how much detection probabilities have changed, but over what area. One interesting quality of \( \Lambda \) is that when a UAV flies in a straight path over a previously un-searched region, the rate of change in the area detection probability is directly proportional to its search width according to

\[
\dot{\Lambda} = WV_g
\]

where

\[
W = \text{the UAV effective search width},
\]
\[
V_g = \text{the UAV groundspeed}.
\]

Thus, a maximum \( \dot{\Lambda} \) corresponds to a maximum \( W \) for a given groundspeed.

**B. Parameter Selection**

Before executing a 3D search simulation some underlying search parameters must be chosen. The search algorithms are intended to coordinate the activities of real SUAVs. As such, all of the search characteristics are based on the capabilities of small SUAVs that could be used for search. Utilizing a typical notebook computer for path planning and vision processing on the ground allows the probability map to be updated at a frequency of 2 Hz.

Another critical parameter to select is SUAV velocity. Due to the direct correlation between search rate and SUAV velocity, one might push the SUAV to its maximum velocity during a search. However, this paper assumes that only a single flight is possible for each SUAV during the search. Thus, the optimum flight velocity is the maximum range velocity, or approximately 13 m/s for the small SUAV platforms considered for this work [9, p.26].

The final two parameters to be chosen are the altitude above ground level, \( h \), and the observation angle\(^a\), \( \theta_o \). Selection of these two parameters was done to maximize \( W \) and \( \Lambda \). The maximizing set for the CCD detection model are \( h = 62 \text{ m} \) and \( \theta_o = 2 \text{ deg} \). The effective search width for this choice is 89.3 m, which is used to space search paths in both real and simulated searches.

\(^a\)The agent is assumed to have a fixed camera.
C. The Probability Map

At the beginning of a search each node has zero probability of having a detected target in its area, and the probability that a target is in any given area is defined by some initial probability map. This probability map is generated by considering factors such as the last seen position, probable target behavior, prominent landmarks and other evidence. The probability grid is the medium on which the probability map is stored and updated. It is a three-dimensional data grid that records whether or not all of the possible target locations have been searched, and how well. The underlying grid can be updated in simulation or during a real search, and can be helpful in planning effective paths. Grid updates require access to both the camera calibration parameters and UAV telemetry.

When dealing with a non-uniform discrete probability map it is useful generate an optimum detection function. The optimum detection function results from searching the nodes in order of descending probability under the assumption that the nodes may be consecutively searched in this order. This serves as a standard for comparison when judging the effectiveness of path planning approaches. To generate the optimum detection function, one must know the node search rate

\[ \dot{N} = \sum_{j=1}^{N_{UAV}} \dot{N}_j, \]

where \( \dot{N}_j = \dot{\Lambda}_j / \delta^2 \) for each agent, and \( \dot{\Lambda}_j \) is obtained from Eq.(6) for each agent. \( \dot{N} \) defines how fast the UAVs can collectively search nodes during straight and level flight.

Using this definition, an optimum detection function may be created using the following steps:

- Estimate or measure the search width and groundspeed for each agent.
- Calculate \( \dot{\Lambda} \) for each agent using Eq.(6).
- Order the nodes in descending probability from \( i = 1 \ldots N \).
- Assume that the nodes are searched at a rate of \( \dot{N} \).
- Step through the nodes from first to last, recording a new time and probability after each node. Thus, evaluation of node \( i \) yields \( \Delta t = 1/\dot{N} \) and \( \Delta p(D \cap C_i) = p(C_i) \).
To show how this can be used, consider the probability maps shown in Figure 4. A victim lost in the region of interest could be considered to be equally as likely to be anywhere within the region, as shown in Figure 4(a). However, if more information were considered, then a probability map such as that shown in Figure 4(c) could be generated to describe the victim’s probable location. This distribution is significantly biased towards roads, the group destination, the last-seen location, and other topological features. Generating an accurate probability map can be a significant step towards maximizing search efficiency. If this probability map is searched in high probability regions first, followed by a thorough sweep of its remaining areas, then the detection probability can be maximized at each point in time. The first step in optimal search is generating this map.

With a probability map in hand, an optimal search allocation may be calculated by using the steps above. When evaluating the uniform map, this generates a detection function that is the discrete equivalent to Koopman’s definite range law, as shown in Figure 4(d). However, this method also provides a useful analysis for more complex probability maps as shown in Figure 4(e). This method is a simple way to evaluate the progress of a search if the agents could be at the most beneficial location on the map at all instants of time. In practice, agents are unable to search the nodes in order of descending probability, but significant differences between the optimum detection function and an actual detection function tend to point out where improvements in the search strategy can be made.

Figure 4. When considering the possible locations of a lost victim, a uniform probability map may be chosen. (See Figure 4(a).) However, another option is to generate the map based on information like terrain data, landmarks, last seen location and probable target behavior. (See Figure 4(c).) The center map shows some key locations that might be associated with the lost victim: C = campground, D = destination, S = last seen location. On both probability maps, red indicates the areas where the victim is more likely to be, while green indicates the less likely areas. Notice that the initial detection probability increases more rapidly for an optimal search of the non-uniform map, because the highest probability nodes can be searched first.
IV. Path Planning

In this section the metrics of search efficiency and search completeness are offered as measures of search effectiveness. In addition, two path plans are developed and combined into a composite strategy. The first plan is designed to be efficient. The agents seek out high probability regions using a locally greedy search, coordinated by a global assignment. The second plan seeks to be complete. It generates offset paths from a local boundary according to search width, and assigns each agent to follow the nearest available path. Each plan is designed with the capability of controlling multiple SUAVs, even though coordinated search results are not shown in this paper. These plans are carried out in the search simulator, where UAVs are assigned approximate dynamic and aerodynamic constants according to the size, shape and weight of MAGICC Lab UAVs.

A. Efficiency vs. Completeness

The effectiveness of a search strategy is dependent on which quality is more important to the success of the search: efficiency or completeness. These two qualities apply particularly to a finite search where limited search effort is available. The search efficiency, \( \xi(t^*) \), and search completeness, \( \zeta(t^*) \), of a given detection function are shown in Figure 5. They are defined as

\[
\xi(t^*) = \int_0^{t^*} p(D(t) \cap C_i) dt \\
\text{and} \\
\zeta(t^*) = p(D(t^*) \cap C_i),
\]

where \( t^* \) marks the time at the end of the search.

Notice that the efficiency and completeness of two different detection functions are only comparable for a given ending time. As the search begins, a strategy that is more efficient is also more complete. However, as time progresses a less efficient strategy may attain higher completeness than its competitor. Thus, a highly complete strategy also has a certain level of efficiency, and a highly efficient strategy also has a certain level of completeness. An increase in efficiency leads to a decrease in mean detection time, and completeness is a measure of the target detection probability when all of the available search effort has been expended. In certain situations it is beneficial to switch off between an efficient plan and a complete plan as the updated probability map changes.

B. The Greedy Search

In order to design a path plan that focuses on efficiency, each SUAV must be drawn to the most profitable regions first. Remember, each node in the probability grid is keeping track of the probability that a target is within the node’s area, and the probability that it has been detected given that it is in the node’s area. However, the probability of interest to a UAV is the probability that any node’s area contains an un-detected target:

\[
p(U(t) \cap C_i) = p(U(t) | C_i) p(C_i)
\]

or

\[
p(U(t) \cap C_i) = (1 - p(D(t) | C_i)) p(C_i).
\]

Now consider the agent at position \( w \) and heading in the \( \psi \) direction. A greedy search\(^b\) may be carried out by evaluating local probabilities at a distance \( R_p \) and angle \( \psi_p \) relative to the SUAV, and then traveling

\(^b\)Refers to being locally greedy, or greedy in the spatial domain.
towards the highest probability. (See Figure 6.) To do this, a point \( q_{\psi_p} \) is located according to some assumed direction, \( \psi_p \), and some assumed range, \( R_p \). This point is

\[
q_{\psi_p} = \begin{bmatrix} w_x + R_p \cos (\psi + \psi_p) \\ w_y + R_p \sin (\psi + \psi_p) \\ 0 \end{bmatrix}.
\]

To examine the benefit of viewing \( q_{\psi_p} \), some measurement must be made to determine the average probability

\[
\bar{p}_{\psi_p} = \frac{1}{N_{\psi_p}} \sum_{i=1}^{N_{\psi_p}} p(U \cap C_i) / N_{\psi_p}
\]

in its vicinity, where \( N_{\psi_p} \) is some number of selected nodes around \( q_{\psi_p} \). This average measures the benefit of traveling in the \( \psi_p \) direction and viewing the surrounding area of \( q_{\psi_p} \). Note that the traveling SUAV will affect the probabilities along a path whose width is approximately equal to the agent’s search width. As such, an estimate for \( \bar{p}_{\psi_p} \) may be obtained by averaging \( p(U \cap C_i) \) for all nodes within a distance \( W/2 \) from \( q_{\psi_p} \).

Specific values for \( R_p \) and \( \psi_p \) must also be selected. If \( R_p \) is too small, then the UAV will end up getting a \( \bar{p}_{\psi_p} \) estimate that includes some recently-viewed nodes. On the other hand, if \( R_p \) is too large, then the UAV may change course before having an opportunity to pass over the selected \( q_{\psi_p} \). Fortunately, in the primarily downward-looking case, the UAV does not affect the current probability of nodes at a range more than \( W/2 \) away. This distance, in addition to the \( W/2 \) sampling radius, leads to a heuristic of \( R_p = W \).

At this distance, the \( \psi_p \) angle must be small enough to allow the SUAV to turn and view the associated \( q_{\psi_p} \) while traveling forward. However, a maximum \( \psi_p \) range is also desirable to allow the UAV greater choice among local maximums. This paper uses \( \psi_p = \{-2.0, -1.8, \ldots, 1.8, 2.0\} \) radians, which spans \( \pm 115^\circ \) because simulations have shown that the modeled SUAV can turn quickly enough to view \( q_{\psi_p} \) points that are \( W=89 \) m away at these angles.

An algorithm could certainly look further ahead than \( R_p \) if desired, and some researchers have investigated more sophisticated algorithms where agents fly along some locally-optimal path. However, the given procedure is one that fulfills the requirement of viewing nearby locations in a greedy fashion. Due to the fact that this algorithm is intended for use with one or more SUAVs, each UAV \( u_j \) is discouraged from searching near other agents by modifying \( \bar{p}_{\psi_p} \) according to

\[
\bar{p}_{\psi_p}^j = \bar{p}_{\psi_p} \prod_{n=1}^{N_{UAV}} k_{un}, \quad n \neq j,
\]

where \( N_{UAV} \) is the number of UAVs, and UAV \( u_j \) is the one making the directional decision. This uses

\[
k_{un} = \begin{cases} 
0 & \text{if } d_{un} < R_p \\
d_{un} R_p & \text{if } R_p \leq d_{un} < 2R_p \\
1 & \text{if } d_{un} \geq 2R_p 
\end{cases},
\]

where \( d_{un} \) is the distance from some other UAV, \( u_n \), to the point \( q_{\psi_p} \). In other words, \( k_{un} \) is a scale factor that modifies the average probability associated with \( q_{\psi_p} \) such that \( \bar{p}_{\psi_p}^j \) is zero within \( R_p \) of another UAV and rises linearly until \( \bar{p}_{\psi_p}^j = \bar{p}_{\psi_p} \) at \( d_{un} \geq 2R_p \). The range \( R_p \) is used in \( k_{un} \) because it is the same radius that is used in directional decision-making. Thus, the given logic always makes heading towards a proximate UAV look un-attractive in the greedy search.

American Institute of Aeronautics and Astronautics
With this logic, SUAVs can detect the most profitable direction. Then each UAV is commanded towards the maximum $\hat{p}\psi_p$ value at each time step according to

$$\psi_g = \frac{\tau \psi_{g,\text{last}} + \Delta t \psi_{p,\text{max}}}{\tau + \Delta t},$$

(13)

where $\psi_{p,\text{max}}$ is the $\psi_p$ corresponding to the maximum $\hat{p}\psi_p$ value, $\tau$ is a time constant, and $\Delta t$ is the time step size. Eq.(13) is a low-pass filter on the commanded heading, where $\tau$ has been tuned to produce commands consistent with the dynamic capabilities of the SUAV airframe. The time constant is chosen to allow the UAV to turn relatively quickly, but the low-pass filter prevents an over-aggressive response as the UAV travels over a changing probability terrain. For the simulated SUAV, a value of $\tau = 1.2$ s is used.

One of the problems in executing a purely greedy search is that agents can wander around for a long time, chasing a local maximum that is not nearly as profitable as in other regions of the map. The greedy search is part of an efficient plan, but lacks a holistic view. This can be added by directing agents to more profitable areas according to a global mandate, but still allowing them to execute a locally greedy search when they get there.

In order to steer agents towards the most profitable regions on a probability map, these regions must first be identified as shown in Figure 7. This is done using a method that is similar to that of a connected-component video processing algorithm. The nodes that are higher than a given probability threshold are identified and collected into groups. These groups are characterized according to their contained probability,

$$p(U \cap C_k) = \sum_{i=1}^{N_k} p(U \cap C_i),$$

(14)

and their total area,

$$A_k = \delta^2 N_k,$$

(15)

where $N_k$ is the number of nodes in group $G_k$, $\delta$ is the node spacing, and $p(U \cap C_i)$ is defined in Eq.(10).

Figure 7. On this probability density surface, red indicates a high $p(U \cap C_i)$ compared to other regions, and green indicates a low $p(U \cap C_i)$ compared to other regions. Groups of nodes with a probability above a certain threshold have been identified and bounded with a blue line.

When searching with more than one UAV, any given group, $G_k$, most likely has a different probability rate to UAV $u_j$ than it does to any other UAV. This rate can be defined as

$$\hat{p}_{jk} = \frac{p(U \cap C_k)V_g}{d_{jk} + A_k/W},$$

(16)

where $d_{jk}$ is the closest distance from UAV $u_j$ to group $G_k$. Notice that the denominator is an approximation of the travel distance required to get to group $G_k$ and search it, and the numerator contains a sum of the node probabilities within $G_k$. Multiplying this probability length density by $V_g$ yields an estimate of the probability rate associated with searching group $G_k$. Due to the fact that $p(U \cap C_k)$ and $A_k$ are constant
for a given node group, the maximum $\dot{p}_{jk}$ for a given node group also indicates the closest UAV $c$. However, Eq. (16) is only an estimate of the probability rate, and does not take into account the benefit of regions on the way to $G_k$.

The value $\dot{p}_{jk}$ can be used to assign each UAV to a highly beneficial node group as follows:

1. Each UAV-group pair is assigned a probability rate according to Eq. (16).
2. The list of probability rates is ordered from highest to lowest.
3. The UAV and node group corresponding to the maximum probability rate are assigned to each other.
4. The list is sequentially evaluated, and UAV/node group assignments are made when the probability rate corresponds to a UAV and node group that have been previously un-assigned.

This means that every UAV will be assigned a group if there are more groups than UAVs, but that some UAVs will go un-assigned if there are more UAVs than groups. The un-assigned UAVs continue to search in a locally greedy fashion. Once a UAV has been assigned to a group, it is commanded to fly towards it if it is further than $d_G$ from it. Once it comes within distance $d_G$ from the group it begins a local greedy search. This paper uses a value of $d_G = R_p$ to ensure that the UAV comes close enough to detect the desired node group.

Although it is desirable to direct UAVs to more profitable areas, adjusting their search by giving them a new assignment can pull them away from some smaller probabilities that may need to be re-searched later. Thus, it is beneficial to establish an update rate that will not allow the UAVs to search in un-profitable regions for too long, but which offers only periodic guidance. Accordingly, the groups and assignments are calculated once every $t_A = 20$ seconds, with node probability updates still occurring once every half second. This greedy approach ensures that profitable nodes are accounted for in the search plan whether they are nearby or not.

As mentioned previously, this global assignment depends on some number of identified nodes that have been placed into node groups, where the appropriate number of nodes is experimentally determined. There should be enough nodes to give the UAVs guidance, but not so many nodes that the UAVs are directed to huge node groups that contain both high and low probabilities. This paper groups $t_G = 7$ seconds-worth of nodes once every $t_A = 20$ seconds. Based on $t_G$, the actual number of nodes that is grouped is $N_G = \dot{N}t_G$, where $\dot{N}$ is defined in Eq. (7).

### C. The Contour Search

In contrast to the greedy search, which focuses on efficiency, the contour search is designed to perform well in terms of completeness. It searches a given area in a highly systematic fashion, and does not leave large holes or overlap. To do this, the SUAV follows a set of offset paths that are spaced $W$ apart rather than seeking out probability differences.

The generation of offset paths has been extensively studied in the field of pocket milling. Hatna claims that for regular shapes a zigzag pattern is more efficient, but for complex shapes a contour pattern is more efficient. (The word ‘efficient’ here refers to the shortest path length.) Yao and Gupta elaborate on this statement by developing a branch and bound method that generates a composite cutter path using a combination of zigzag and offset contour paths. They claim that this new path is guaranteed to be shorter than traditional zigzag or contour paths alone.

Boundaries that follow probability map contours are often quite complex. Thus, path generation could be done with the branch and bound method, but for simplicity this paper implements a pure contour pattern from the VRONI algorithm of Held. This algorithm generates contour paths by trimming offset contour segments to the boundaries of an extended Voronoi diagram. This method is used to generate offset paths at specified intervals (search widths) from the boundary.

Once these offset paths are created, it is important to note that great care has been taken to ensure that they do not overlap, and that they are spaced $W$ apart. Thus, a contour search requires that they be followed in some organized and non-overlapping fashion. This could be done by simply allowing the UAV to follow the contours using a vector-field approach. However, this method can cause a drastic and

---

$^c$ This assumes that an identical velocity and search width are common to all agents.

$^d$ As described in Section B a probability update occurs once every half second because that is the fastest that the given ground station is able to process a video frame while coordinating UAV flight.
un-predictable viewing pattern when the UAV encounters sharp direction changes. Another way to follow the contour lines is to use the sliding segment technique shown in Figure 8. This technique can be used to predict turns in the upcoming contour path, and keep the camera pointed at the path.

This technique is accomplished by keeping track of $q_c$, a point representing the forward progress of the UAV along the contour path. Using a segment length of $L_s = 90$ m, $q_f$ is created $L_s/2$ in front of $q_c$ along the contour path, and $q_b$ is created $L_s/2$ behind $q_c$ along the path. The UAV is then commanded to fly along segment $[q_b q_f]$ using a vector-field approach, and the segment midpoint, $q_m$, is used as a benchmark for UAV forward progress along the segment. At each time step the UAV moves forward some distance $\delta_m$ along segment $[q_b q_f]$, and point $q_c$ is advanced the same $\delta_m$ along the contour path. Then $q_f$, $q_b$, $q_m$, and $\delta_m$ are calculated over again.

The sliding segment method is one predictable way to control the UAV system when using a fixed camera. The UAV follows a ‘sliding segment’ that changes direction in anticipation of upcoming curves, keeping the camera largely pointed at the path. Together, offset contours and the sliding segment method represent one way to search an area in a systematic manner with a fixed camera.

Unlike the greedy search, the contour search does not have time or node thresholds. A boundary is simply drawn around all of the nodes having any chance of representing an un-detected target. The offset contours are then generated from this boundary and followed by the nearest available UAV. This method of assignment illustrates that one of the assumptions is probability uniformity when executing an offset contour search. UAVs are assigned to follow the nearest available contour because all contours are assumed to have equal benefit other than their distance from the UAVs.

D. The Composite Search

A composite search can be executed by switching off between the greedy search and the offset contour search at different times. In general, the greedy search is used unless a certain degree of uniformity is present, and then the contour search is used. The composite search method is similar to the greedy search. It simply uses a new time threshold, $t_C$, to determine how many seconds-worth of nodes must be within a single group before a contour search is warranted. The rate at which UAVs are capable of searching $(\dot{N})$ then determines the precise number of nodes according to $N_C = \dot{N} t_C$. Once contours have been generated they must all be followed before the UAVs are allowed to switch back to greedy searching.

To illustrate this logic, assume that there are two UAVs searching a probability map, each traveling at a velocity of 12 m/s and having a search width of 8 m. Also assume that the underlying probability grid has a spacing of $\delta_g = 20$ m. Under these conditions, and assuming no overlap, nodes are searched at a rate of

$$\dot{N} = \frac{N_{UAV} V_g W}{\delta_g^2} = 48 \text{ nodes/s,}$$

where $\dot{N}$ is the node search rate defined in Eq.(7), $N_{UAV}$ is the number of UAVs, $V_g$ is groundspeed, and $W$ is agent search width.

The greedy search and the offset contour search are both dependent on groups of nodes that have been identified and contained in a boundary. These nodes are identified because they are each above some probability threshold, $P_T$. For the greedy search, assume that at any one time it is desirable to have at least $t_G = 6$ s worth of bounded nodes, where $t_G$ is the time threshold for the greedy search. This means that it is desirable to have at least $N_G = \dot{N} t_G = 2.88$ nodes bounded at any one time, where $N_G$ is the node threshold for the greedy search.

Now consider the probability map and node list shown in Figure 9. If it is desirable to have at least $N_G = 2.88$ nodes bounded at any time, then by stepping down the node list to the first node with $i \geq 2.88$, the probability threshold $P_T = 0.10$ is found. This threshold can then be used in the spatial domain to identify and establish boundaries around all nodes with probabilities greater than or equal to $P_T$. The given method for selecting $P_T$ causes the collective number of nodes in all boundaries, $N_B$, to always be greater than $N_G$. 

\[\text{Figure 8. The sliding segment method.}\]
When $P_T$ is selected in this way, each resulting group, $G_k$, contains some number of nodes, $N_k$. However, all nodes with $p(U \cap C_i) \geq P_T$ are included, which makes it possible for the $N_k$ of some groups to become large. When this happens, the time threshold, $t_C$, is used to establish how many seconds-worth of nodes must be contained in any one group to justify starting a contour search. If $t_C = 23$ s, then the number of nodes that must be contained is $N_C = Nt_C = 11$ nodes, where $N_C$ is the node threshold for a contour search. If more than 11 nodes are contained within any one boundary, then a contour search begins.

To illustrate this procedure, assume that at some point the UAVs were able to search the probability map in Figure 9 such that the three nodes in the upper-right hand corner all have values of $p(U \cap C_i) = .05$. The new situation is shown in Figure 10. Stepping down the list to the node with $i \geq N_G$ selects a probability threshold of $P_T = .05$, which causes a boundary to be drawn around 12 nodes. This also causes $N_k$ to be greater than or equal to $N_C$, or $12 \geq 11$, which causes a contour search to begin.

This paper uses the values of $t_G = 7$ s and $t_C = 100$ s as the time thresholds for the greedy and contour search respectively. In the greedy search UAVs are directed towards profitable areas that are identified using $t_G$. This parameter must be high enough that the UAVs typically have bounded areas to work on, but low enough that they are not directed to large areas that offer little guidance. On the other hand, $t_C$ is chosen to be high enough that a contour search will begin when a sizable search area exists, but low enough that the search areas do not have to be excessively large for this to happen. Both $t_G$ and $t_C$ are chosen experimentally.
The presented logic is one way to identify when a sufficient degree of uniformity exists to begin a contour search. In a more general sense, it also indicates when to perform a search that focuses on completeness vs. one that focuses on efficiency. Notice that the regions of high probability are searched until a fairly uniform distribution remains. At this point it becomes profitable to not only change the region of search, but also the method of search.

V. Search Simulation

Selection of an appropriate strategy is based on the relative importance of efficiency and completeness, along with the available amount of search effort. This section is designed to give an understanding of the strengths and weaknesses of each search strategy. To do this, a constant area is searched for an amount of time that is consistent between experimental runs. However, the probability distribution changes in each run so that the trade off between efficiency and completeness can be seen.

Consider the variable probability map presented in Figure 11(a). This map is defined by a variable region of non-uniform probability, $A_{NU}$, and a variable region of uniform probability, $A_U$. Together these two regions make up the total area, $A_T = A_{NU} + A_U$, and are related through the ratio $R_{NT} = A_{NU}/A_T$. The variable map is generated based on this ratio, and the ratio is varied from zero to one. The total region is a square with edge length $L_T = 750$ m, and the non-uniform region is a square with edge length $L_{NU} = L_T \sqrt{R_{NT}}$. The regions $A_{NU}$ and $A_T$ share the same lower left-hand corner.

When calculating the node probabilities in the variable map, the first step is to allocate a probability of $1 - R_{NT}$ to all of the nodes within $A_T$. Then, an additional probability of $R_{NT}$ is allocated to the nodes that are within $A_{NU}$. This second allocation is done in a non-uniform fashion, based on a $5 \times 5$ matrix of Gaussian distributions. Each distribution has a standard deviation of 25 m and is placed at $x$ and $y$ increments of 150 m throughout $A_T$, all centered within $A_T$. However, these Gaussian distributions are only allowed to affect the nodes in $A_{NU}$. As $R_{NT}$ grows from zero to one, more of the map is affected by the Gaussian distributions, and becomes non-uniform.

(a) The variable probability map.  
(b) Detection functions on a map with $R_{NT} = 0.2$.

Figure 11. Figure 11(a) shows the concept of the variable probability map. Figure 11(b) shows the detection functions that result when each of the path plans is carried out on the map characterized by $R_{NT} = 0.2$.

A single UAV searches each generated map by starting in the center of $A_T$, and each experimental run stops at time $t^* = A_T/V_g W = 484.43$ s. Theoretically, this is the time necessary for the UAV to make one complete sweep of the total area. Figure 11(b) shows a set of detection functions that result when using the presented path plans on the variable probability map. The efficiency and completeness of each detection function may be computed according to Eq.(8). The results for all of the simulations are summarized in Figures 12 and 13.

It is important to note that the results are dependent upon the specific situation presented in each scenario. In some cases the map configuration may cause a UAV to end up finishing a contour right next to a region that needs to be searched, or being pulled away from something just before finishing its search. So, the key is to look for general trends rather than at specific data points.
Despite this variation, Figures 12 and 13 are very helpful in illustrating the trade-off between completeness and efficiency. Note that in almost every case the contour search is the method highest in completeness and lowest in efficiency. Also note that the composite algorithm seems to be advantageous compared to the greedy search within the range $R_{NT} = 0$ to 0.3. This is the range where a significant amount of uniform probability remains after the most profitable regions have been searched. This range depends on the relationship between $t_C$ and $A_T$. In other words, the range could extend higher if the total number of uniform nodes remaining were greater than $Nt_C$ after the more profitable (variable) regions had been searched.

For a map with certain uniformity, the composite search strategy can be beneficial in augmenting the completeness of an efficient search strategy. Remember that the importance of efficiency is affected by the available search effort, the search rate, and the time-dependent consequences of not finding the target. On the other hand, the offset path search strategy should be used if completeness is highly important relative to efficiency. These are the rules of thumb sought for in doing the variable-map experiments.

VI. Conclusion

This paper explores issues related to a search simulation where SUAVs and CCD camera sensing are used. This paper offers a discrete method for measuring search rate using a probability grid, and a method that determines an optimal detection function for any given probability map. This paper also presents the metrics of efficiency and completeness, and offers the greedy and offset contour methods as examples of a path plan that focuses on each metric. Simulation results are presented to illustrate the trade-offs between the two plans, and a composite search strategy is shown that is more efficient in most cases.

These considerations are prompted by previous work in optimal search theory, and show that an effective search strategy can be selected based on the importance of efficiency and completeness. It is important to note that a search simulation may be executed in tandem with a real-world search. Telemetry can be fed into the simulation describing the UAV search history, and the resulting probability distribution can then be used to guide the evolving search effort.
References


