Sensor-Driven Area Coverage for an Autonomous Fixed-Wing Unmanned Aerial Vehicle

Liam Paull, Carl Thibault, Amr Nagaty, Mae Seto, and Howard Li

***Abstract*—Area coverage with an onboard sensor is an important task for an unmanned aerial vehicle (UAV) with many applications. Autonomous fixed-wing UAVs are more appropriate for larger scale area surveying since they can cover ground more quickly. However, their non-holonomic dynamics and susceptibility to disturbances make sensor coverage a challenging task. Most previous approaches to area coverage planning are offline and assume that the UAV can follow the planned trajectory exactly. In this paper, this restriction is removed as the aircraft maintains a coverage map based on its actual pose trajectory and makes control decisions based on that map. The aircraft is able to plan paths *in situ* based on sensor data and an accurate model of the on-board camera used for coverage. An information theoretic approach is used that selects desired headings that maximize the expected information gain over the coverage map. In addition, the branch entropy concept previously developed for autonomous underwater vehicles is extended to UAVs and ensures that the vehicle is able to achieve its global coverage mission. The coverage map over the workspace uses the projective camera model and compares the expected area of the target on the ground and the actual area covered on the ground by each pixel in the image. The camera is mounted on a two-axis gimbal and can either be stabilized or optimized for maximal coverage. Hardware-in-the-loop simulation results and real hardware implementation on a fixed-wing UAV show the effectiveness of the approach. By including the already developed automatic takeoff and landing capabilities, we now have a fully automated and robust platform for performing aerial imagery surveys.**

***Index Terms*—Coverage path planning, hardware-in-the-loop, information theory, unmanned aerial vehicles.**

I. Introduction

THERE IS an increasing demand for surveillance and map building using unmanned aerial vehicles (UAVs). Not only do UAVs make dangerous tasks safer for humans, they often cost less to operate. Many applications require the UAV to cover an area of terrain. Examples include law enforcement, disaster management, defense, natural resource conservation, search and rescue, fire management, information services, agriculture and aerial photography.

Complete autonomy is the next phase of development for these UAV systems. This is particularly valuable when the mission is either dull, dirty, or dangerous. With the benefit of full autonomy comes the requirement for increasingly sophisticated control techniques. One challenge is to develop reliable path planning algorithms that help UAVs achieve complex mission objectives without any human intervention. The restrictions on the path planning module are usually severe since it is operating within a larger system with limited resources onboard a small airborne vehicle. For example, the path planning algorithm must be able to plan one or more safe paths of specified length that achieve the objective in real-time based on acquired sensor data without hogging system resources such as CPU cycles or memory.

The majority of area coverage path planning research has been done for ground vehicles. Many of these algorithms use the presence of obstacles to generate a cell decomposition (e.g., the Boustrophedon cellular decomposition [1]). However, it is normally assumed that prior knowledge of the environment is perfect, and plans are made completely offline without the ability to adjust if problems are encountered. In addition, coverage algorithms that are implemented using predefined waypoints or tracks, such as in [2], lack the ability to optimize the coverage based on the actual trajectory of the vehicle rather than the preplanned path.

The sensor-driven approach proposed here plans paths online based on sensor data gathered from the field. Coverage with the payload camera is based on a target detection model using the standard projective camera model geometry and the actual UAV pose. As such, the vehicle can accurately model which areas have already been covered and make control decisions *in situ*. This is achieved within an information theoretic framework that selects control actions that will maximize the expected information gain. However, to avoid the normal exponential horizon scalability problem (e.g., [3]) that occurs when information gain is formulated directly over control plane values, optimizations are done over the outer loop heading reference leaving the inner loop heading tracking controllers untouched. The downside to such an approach is that there is a lag between desired heading and actual heading that corresponds to the settling time of the heading tracking control. To compensate for this, the gimbal angles of the camera are also optimized at a higher frequency. Since maximizing information gain is a greedy approach, a method presented by

Manuscript received July 12, 2012; revised June 18, 2013; accepted October 28, 2013. Date of publication November 21, 2013; date of current version August 14, 2014. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), and in part by the Canada Foundation for Innovation. This paper was recommended by Associate Editor T. Huntsberger.

L. Paull, C. Thibault, A. Nagaty, and H. Li are with the Department of Electrical and Computer Engineering, University of New Brunswick, Fredericton, NB E3B 9P8, Canada (e-mail: liam.paull@unb.ca; carl.t@unb.ca; amr.nagaty@unb.ca; howard@unb.ca).

M. Seto is with the Department of Mechanical Engineering, Dalhousie University, Halifax, NS B3H 4R2, Canada (e-mail: mz715250@Dal.Ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2013.2290975

Fig. 1. University of New Brunswick collaboration based robotics and automation (COBRA) group fixed-wing powered glider with a 2.4 meter wing span that was used for experimental validation of the proposed area coverage algorithm (photo courtesy Yvon Thibault).

the authors in [4] for autonomous underwater vehicles called branch entropy is used, showing that the generic approach is applicable to different autonomous platforms.

To summarize, the benefits of the proposed approach are the following.

1) Detailed knowledge of the workspace is not required *a priori*.
2) The coverage map is maintained and paths are planned based on the actual UAV trajectory rather than assuming that the vehicle exactly follows a preplanned path.
3) Paths are planned on-the-fly so that the UAV is able to complete its mission regardless of unexpected disturbances.
4) The planner converges to complete coverage.

The proposed method has been tested using a hardware-in-the-loop simulation and experimental validation on a fixed-wing UAV shown in Fig. 1.

The following section will provide some background and a more detailed literature review. Section III describes how the coverage map is maintained during flight. Section IV details the proposed approach for generating the heading reference values. Section V describes how the camera angles are optimized. Section VI will describe the setup used for simulations and field trials. Section VII will provide the results. A short discussion is given in Section VIII and finally, the conclusion is given in Section IX.

II. Background of Research

In path planning, the configuration space, C, is the space of all possible robot configurations and has the same dimension as the number of degrees of freedom of the platform. Let the free configuration space, $C_{free} \subseteq C$, be the subset of all possible configurations for which there is no contact between the robot and any obstacle [5].

We can then define the navigation task of path planning as finding a curve in the free configuration space, C_{free} that connects a start configuration, $q = q_i$ to a goal location, $q = q_g$. More specifically, as described in [6], the path can be expressed as a curve $\tau : [0, 1] \rightarrow C_{free}$ with $\tau(0) = q_i$ and $\tau(1) = q_g$.

Algorithms abound in the literature for solving this problem with different design considerations, such as complete versus incomplete knowledge of the environment, deterministic versus stochastic estimation of robot pose and many others [7]. Popular methods include bug algorithms, roadmap-based methods, potential fields, cell decomposition methods, and sampling-based methods [5]. Many of these methods provide a framework for generating a graph or tree which can then be searched using any classical search algorithm, such A*, D*, Dijkstra's algorithm, as well as advanced biologically-inspired search techniques such as particle swarm optimization and genetic algorithms.

A. Coverage Path Planning

Coverage path planning (CPP) is related to the classical start-to-goal path planning objective with notable differences. Consider a mobile robot with an onboard sensor with some 2-D swath Z. A trajectory through the workspace W will result in a set of N sensor readings: $\{Z_1, Z_2, ..., Z_K\}$. The goal of CPP is to generate a path through the workspace such that

$$\bigcup_{k=1}^{K} Z_k \supseteq W \tag{1}$$

meaning that at some point the sensor swath has passed over every point in the workspace. If the path can be broken down into a series of waypoints, then the problem can be decomposed into several start-to-goal path planning problems. However, treating the problem in this context can be problematic because the coverage obtained depends not only on the waypoints, but also on the trajectories between the waypoints.

In the 2001 survey paper [8], Choset states that CPP algorithms can be classified as either based on heuristics or on cell decomposition.

Heuristic methods usually define a set of behaviors or rules that should be followed. For example, in [9], a heuristic method is developed that combines three types of behaviors: inward spiral, shifting spiral, and greedy. In [10], three independent behaviors are defined: spiral path tracking, wall following and virtual wall following.

In cell decomposition, the workspace is subdivided into smaller areas. A simple method of covering each cell is specified and then once every cell has been covered, the mission is guaranteed to be complete. Within the cell decomposition methods, essential considerations are how to generate the cells and what shape they should be. The Boustophedeon cell decomposition is first introduced in [1] and is based on determining critical points in the workspace where the connectivity of a slice of free space changes. This is expanded in [11] and [12] to include a method in which critical points can be detected online. This decomposition is referred to as a Morse decomposition. The connectivity of the cells is represented by the Reeb graph. In [13], the algorithm is extended to apply to a robot whose sensor footprint is larger than the platform footprint. In this approach, two heuristics

are combined: one for coverage of wide open spaces based on a Boustrophedon search, and one for constricted areas which is based on the generalized Voronoi diagram.

A popular cell decomposition method for solving the coverage problem is the spanning tree coverage algorithm [14]. The workspace is decomposed into cells equal in size to the sensor footprint and then a Hamiltonian cycle is determined which visits each cell exactly once. The main drawback of this method is that the path generated contains many sharp turns which violate the dynamic constraints of any non-holonomic platform.

More recently, a trapezoidal cell decomposition was proposed for an agricultural application [15]. In this approach a split-merge on the cells was used to reduce the search space. Another coverage algorithm for farming applications was presented in [16]. In this case, paths are optimized based on the topology of the terrain.

Also of interest is the work presented in [2], as it is one of the few area coverage algorithms designed for UAVs. Here an exact cell decomposition is used and individual cells are covered with back and forth motions. The focus here is on finding the minimum width of the polygonal workspace and being able to efficiently decompose a concave polygon into many convex polygons.

The only known experimental implementation of fixed-wing area coverage algorithm is presented in [17]. In their implementation, tracks are preplanned based on the Boustrophedon approach. In order to avoid missed coverage caused by the dynamics of the aircraft, a curlicue motion is executed at the start of each track so that the aircraft is properly aligned at the start of the track. They are able to achieve 95% coverage with the greedy planner and 99% coverage by including the curlicue motions.

An important aspect of any coverage planner is whether paths are planned online or offline. An offline planner assumes that perfect prior knowledge of the workspace exists. Online planners are required if it is desired for the agent to be able to adapt to the environment. If the planner uses real-time sensor data then the approach is said to be sensor-driven [15]. The categorization of CPP algorithms from [8] still applies to offline planning schemes, but some new sensor-driven approaches cannot fit into these categories.

Algorithms can also be evaluated based on whether they are provably complete. However, as is noted in [1], the term complete is used in the motion planning sense, not in the operating research field sense. Indeed it is acknowledged that while a plan might achieve provable coverage there is no guarantee of actual area coverage if there is incomplete or incorrect information about the workspace being searched or if the vehicle deviates from the preplanned path. The algorithm presented here is probabilistically complete, meaning that it converges toward complete coverage, and is capable of adapting to incorrect, missing, or non-existent information about the workspace and to unexpected disturbances to the aircraft.

This, and our previous work [4] on autonomous underwater vehicle seabed coverage, extend the coverage path planning problem to account for variable coverage sensor performance that is dependent on the environment and account for deviations from the preplanned path due to disturbances or other factors.

B. UAV Path Planning

One of the most challenging problems for developing autonomy in aerial vehicles is path planning. Since UAVs have limited sensing range and endurance, the time spent surveying specific areas needs to be optimized. Research results on path planning for ground vehicles and robot manipulators are well documented. However, these routes may not be flyable or maneuverable because they do not meet the kinematic and dynamic constraints of the UAV [18]. Existing path planners for UAVs produce a list of nodes that are connected by straight lines, which are eventually connected to the start and finish points. Navigable trajectories are often computed from the high level paths using Dubins curves [19]. For example, the algorithm in [20] investigates using Dubins curves to design paths to view partially occluded targets and solving the travelling salesman problem with a Dubins vehicle is rigorously treated in [21]. However, decoupling waypoint planning and trajectory generation can be problematic from a sensor coverage standpoint. In this paper, this problem is partially overcome by optimizing the gimbal angles of the onboard camera for increased coverage, similar to the approach in [22] for target acquisition.

Other fixed-wing UAV path planning applications published in the literature include following coastlines [23], target localization [3], and energy conservation for solar powered vehicles [24], among many others.

1) *Information Theoretic Approaches to UAV Planning:* Information theory, originally introduced in [25], is a useful way of quantifying the uncertainty of a random variable. In the realm of UAV path planning, these concepts can be applied in a number of different ways. For example, one of the most common applications is to localize ground targets from the air by minimizing their position uncertainty (or maximizing information). The approach is based on optimal sensor placement theory developed in [26]. UAV paths are planned to provide optimal position estimation of a fixed or moving target. For example, in [27] a target is localized using a team of quadrotors. The state of the target is estimated using a particle filter, and a formulation for evaluating the entropy of a particle set is introduced. A similar approach for a fixed-wing UAV using particle filters and information gain is presented in [28]. In [29], multiple UAVs cooperatively perform exploration and gather information about objects of interest. An optimization of vehicle heading is performed to minimize mission time using the Shannon channel capacity equation to quantify information.

Perhaps the most closely related research to our own is [3]. In this approach, fixed-wing UAVs are used to localize a target where the utility of actions is formulated as information gain or probability of detection (PoD). In this case the search is planned over the set of possible trajectories and it is assumed that the relationship between control inputs and outputs is perfect and deterministic. This assumption assumes that once the search over a finite horizon is complete, the UAV is able

to exactly follow the optimal trajectory. A similar approach is presented in [30] where the complexity of information theoretic planning over a fixed control horizon is also noted. Also very closely related is [31] where an optimization is performed over vehicle headings to minimize the probability of misclassification of a target as an alternative to an information gain objective function. The problem of classification of objects of interest at known locations is related to the coverage problem studied here. For example, a similar possible solution is presented in [32], where Shannon information is used to plan paths. In this paper, we make no assumption about *a priori* knowledge of the locations of objects of interest and instead wish to cover an entire area. The relationship between these two problems is akin to the difference between the traveling salesman problem and Chinese postman problem [33]. The problem of optimal planning of angles of viewing, or looks of rectangular objects at known location but unknown orientation is investigated in [34].

A further related application of information theory is for adaptive exploration. Again, originally developed for ground robotics [35] and more recently used for planetary exploration [36], the problem can be posed through the use of information as an optimal control problem [37]. For example, in [38], an algorithm using information is presented to define optimal paths for a UAV to navigate through a region and obtain information about objects of interest. Exploration and coverage of an unknown environment are closely related [39]. However, as noted in [40], a pure information gain solution to the exploration task results in a greedy solution, and therefore suffers from reduced performance in the case that the mission objective is complete coverage.

In our approach, we use information for complete coverage where paths are planned that minimize the uncertainty of binary coverage map variables. The optimization takes place over the outer loop control heading reference value to steer the UAV around the workspace. The coverage map over the environment is maintained as the vehicle traverses the workspace and then information gain is quantified based on this up-to-date map. In addition, actions that support the global objective represented by the branch entropy behavior are traded off against the greedy short term coverage which is quantified through the information gain.

III. Maintaining the Coverage Map

The approach presented here differs from most coverage planning methods in that paths for coverage are generated based on a coverage map that is actively maintained as the vehicle traverses the workspace. This allows paths to be generated online and can result in path planning methods that are adaptive to sensor data gathered *in situ*.

A. Problem Formulation

Decompose the workspace to be covered, W, into a grid of N small cells c^i, at locations $[x_{c^i}, y_{c^i}]^T$, $i = 1..N$. The size of the cells is sufficiently small that the coverage over the cell can be treated as uniform.

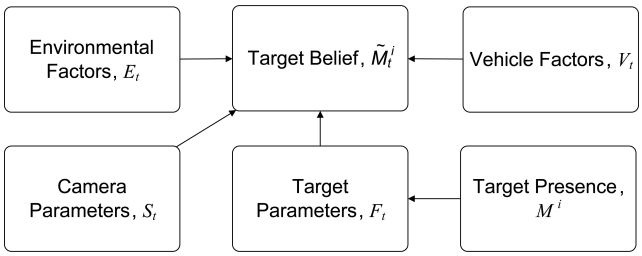

Fig. 2. Bayesian network representation of the target detection event.

Define the random variable (RV) $M^i \in \{target, \overline{target}\}$ to represent the actual presence of a target in cell c^i, $i = 1..N$ of the discretized workspace. Then, define the RV $\tilde{M}_t^i \in \{target, \overline{target}\}$ to represent the target detection event at c^i at time t. There are four possible scenarios for each cell.

1) $\tilde{M}_t^i = target$ and $M^i = target$: target detected.
2) $\tilde{M}_t^i = target$ and $M^i = \overline{target}$: false alarm.
3) $\tilde{M}_t^i = \overline{target}$ and $M^i = target$: missed target.
4) $\tilde{M}_t^i = \overline{target}$ and $M^i = \overline{target}$: no target detected.

Finally, we define a third binary RV $T_t^i \in \{0, 1\}$ that represents whether cell c^i will be correctly classified as either containing a target or not

$$\begin{aligned} p(T_t^i = 1) = \;& p(\text{target detected}) + p(\text{no target detected}) \\ = \;& p(\tilde{M}_t^i = target, M^i = target) \\ & + p(\tilde{M}_t^i = \overline{target}, M^i = \overline{target}) \\ p(T_t^i = 0) = \;& p(\text{false alarm}) + p(\text{missed target}) \\ = \;& p(\tilde{M}_t^i = target, M^i = \overline{target}) \\ & + p(\tilde{M}_t^i = \overline{target}, M^i = target). \end{aligned} \tag{2}$$

Based on the formulation (2), this data collection problem has been framed as a sensor coverage problem where our objective is now to maximize the values of $p(T_t^i = 1) \triangleq p(T_t^i)$ over the entire workspace. The collection of RVs T_t^i, $i = 1..N$ is referred to as the coverage map.

The target detection event can be described by the Bayes' network (BN) model shown in Fig. 2, similar to [4] and [41]. Environmental factors, E, could include lighting conditions. Vehicle factors, V, include orientation of the airframe as well as altitude and velocity. Also note that the gimbal angles of the camera will be included in the vehicle factors. Camera parameters, S, are prespecified or calibrated parameters inherent to the physical camera sensor. Target parameters, F, include size and possibly color of the targets, which are assumed to be known at least approximately known prior to mission launch.

From the BN

$$\begin{aligned} & p(\tilde{M}_t^i, E_t, V_t, S_t, F_t, M^i) \\ & = p(\tilde{M}_t^i | E_t, V_t, F_t, S_t) p(F_t | M_t^i) p(E_t) p(V_t) p(S_t) p(M^i). \end{aligned} \tag{3}$$

Based on the coverage map formulation, we can treat M^i as given and reformulate (3) in terms of T_t^i

$$\begin{aligned} p \qquad & (T_t^i, E_t, V_t, S_t, F_t) = \\ & p(T_t^i | E_t, V_t, F_t, S_t) p(F_t) p(E_t) p(V_t) p(S_t). \end{aligned} \tag{4}$$

If we define the set of parameters and factors at time t to be $Z_t \triangleq \{E_t, V_t, S_t, F_t\}$, then (4) can be further reduced to

$$p(T_t^i, Z_t) = p(T_t^i|Z_t)p(Z_t) \tag{5}$$

and the final probability of detection $p(T_t^i)$ can be calculated by marginalizing Z_t out of the joint probability

$$p(T_t^i) = \int p(T_t^i, Z_t)dZ_t = E_{Z_t}[p(T_t^i|Z_t)] \tag{6}$$

where E_{Z_t} denotes the expectation over the variable Z_t. The distribution $p(T_t^i|Z_k)$ represents the probability that, given a set of parameters and factors, we will correctly classify location c^i as either containing a target or not.

B. Projective Camera Model

The ideal projective camera model is used to obtain a 3-D vector in the camera frame for each point on the image plane (Fig. 3). A measurement $z = (u, v)$ is the projection of a target at position[1] $p^C = \begin{bmatrix} x^C & y^C & z^C \end{bmatrix}^T$ in the camera frame, onto the image plane [42], [43]

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} x^C/z^C \\ y^C/z^C \\ 1 \end{bmatrix} \tag{7}$$

where x^C/z^C and y^C/z^C are the camera frame target coordinates normalized by the target's range and $\mathbf{K}$ is the 3×3 camera intrinsic calibration matrix. The measurement model can be obtained by inversion of (7) followed by coordinate transformation, scaling and translation:

$$\bar{\mathbf{p}}^I = \begin{bmatrix} x^I/z^C \\ y^I/z^C \\ z^I/z^C \end{bmatrix} = \mathbf{R}_{C\to I}(\alpha_{az}, \alpha_{el}, \phi, \theta, \psi)\,\mathbf{K}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{8}$$

where $\mathbf{R}_{C\to I}(\alpha_{az}, \alpha_{el}, \phi, \theta, \psi)$ transforms coordinates from the camera frame to the navigation frame based on the aircraft's roll ϕ, pitch θ, yaw ψ and the gimbal's elevation α_{el} and azimuth α_{az} angles. Using the flat world assumption, the target's range can be approximated by

$$\hat{z}^C = \frac{\begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \mathbf{r}^I}{\begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \bar{\mathbf{p}}^I} \tag{9}$$

where $\mathbf{r}^I$ is the translation vector between the navigation frame and the aircraft's body frame. Finally, the measurement model is given by

$$\begin{bmatrix} \hat{x}^I \\ \hat{y}^I \end{bmatrix} = \mathbf{h}(u, v) = \begin{bmatrix} \hat{z}^C & 0 & 0 \\ 0 & \hat{z}^C & 0 \end{bmatrix} \bar{\mathbf{p}}^I + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{r}^I. \tag{10}$$

[1]The superscript notation is used to denote the frame in which the variable is represented. For instance, V^C denotes that V is represented in the camera frame and V^I denotes that V is represented in the navigation frame.

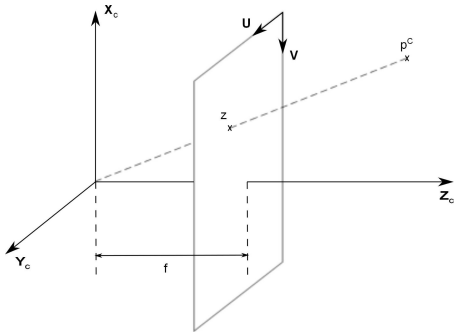

Fig. 3. Projective Camera Model. The camera frame is defined by the three unit vectors $\begin{bmatrix} \mathbf{X}_c & \mathbf{Y}_c & \mathbf{Z}_c \end{bmatrix}$. The image pixel frame is defined by the three unit vectors $\begin{bmatrix} \mathbf{U} & \mathbf{V} & \mathbf{Z}_c \end{bmatrix}$ parallel to the $(\mathbf{X}_c, \mathbf{Y}_c)$ plane at the focal length f along the vector $\mathbf{Z}_c$.

C. Target Detection Model

The target detection model is built on the projective camera model and is similar in concept to the Johnson model used in [31]. Based on how our target recognition system is automatically detecting targets, we can determine the PoD values $p(T_t^i)$ as a function of all of the parameters in the BN.

Target recognition is done through simple blob detection based on thresholding in the hue-saturation-value (HSV) colorspace. It is well known that the HSV image representation is more robust to illumination changes since color is completely represented by only one channel, the hue.

In general, the image will be corrupted by noise. As a result, there is a non-zero probability that any pixel will be inside the threshold, denoted p_{false}. For example, if the hue channel is represented with eight bits then the hue, h takes on a value between 0 and 255. Define the lower and upper thresholds on the hue for target detection to be h_l and h_h, respectively.[2] In that case the probability of having a pixel falsely detected can be roughly estimated as

$$p_{false} = \frac{h_h - h_l}{256}. \tag{11}$$

However, if a target is characterized by a blob of pixels that all pass the threshold, the probability of falsely detecting a target increases with the number of pixels in the blob, n, according to $(p_{false})^n$ which rapidly decays to 0 as n increases. Conversely, the probability that the target detection event is accurate, which is the same as the PoD, at location c^i resulting from measurement Z_t is given by

$$p(T_t^i|Z_t) = 1 - (p_{false})^n. \tag{12}$$

The number of pixels in the target blob is directly related to the viewable area of the target on the ground plane, and inversely related to the area on the ground plane covered by each pixel

$$n = \frac{\text{viewable area of target on ground}}{\text{area covered by individual pixel on ground}} \triangleq \frac{A_t}{A_p}. \tag{13}$$

[2]Note that in some cases it is necessary to have two sets of thresholds to determine red colors, which are represented by a hue of 0.

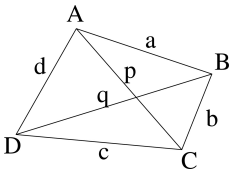

Fig. 4. General non-convex quadrilateral.

The viewable area of the target, A_t can be assumed to be at least approximately known assuming that we know what type of target we are looking for. However, the area covered by an individual pixel on the ground will be a function of the vehicle parameters at the time that the picture was taken. Specifically, based on the projective model of the camera, the altitude and aspect of the camera.

The area on the ground plane that is covered by an individual pixel (u, v) will define a convex quadrilateral with vertices at $\{h(u, v), h(u+1, v), h(u, v+1), h(u+1, v+1)\} \triangleq \{A, B, C, D\}$ where h is the mapping from the pixel frame to the ground plane in global coordinates (10). The general formula for calculating the area of a convex quadrilateral can be used to calculate the area on the ground plane that is covered by an individual pixel

$$A_p = \frac{1}{4}\sqrt{4p^2q^2 - (a^2 + c^2 - b^2 - d^2)^2} \tag{14}$$

where $a = ||A-B||$, $b = ||B-C||$, $c = ||C-D||$, $d = ||D-A||$, $p = ||A - C||$, $q = ||B - D||$ as shown in Fig. 4.

In the actual implementation, it is reasonable to specify a minimum blob size required for blob detection, denoted by n_{min}. So, the actual PoD is given by

$$p(T_t^i|Z_t) = \begin{cases} 1 - (p_{false})^n & \text{if } n > n_{min} \\ 0 & \text{else.} \end{cases} \tag{15}$$

IV. Heading Reference Generation

The core of the approach for coverage is the generation of heading reference values by evaluating an objective function that operates over the actual coverage map that is being maintained. Possible headings are evaluated based on how much closer they bring us to the complete coverage objective in both the short term (information gain) and the long term (branch entropy).

A. System Structure

The overall system structure is shown in Fig. 6. Vehicle state data (position, orientation, and linear and angular velocities) is sent through a wireless link to the ground control station (GCS). Control decisions are made on the GCS and uploaded as heading reference values back to the UAV where they are tracked by the inner loop control onboard.

The GCS is running the mission-oriented operating suite (MOOS) middleware [44] as shown in Fig. 7, and uses the interval programming (IvP) [45] multiobjective piecewise optimization framework to reconcile behaviors at runtime. Each block in Fig. 7 represents a separate MOOS application

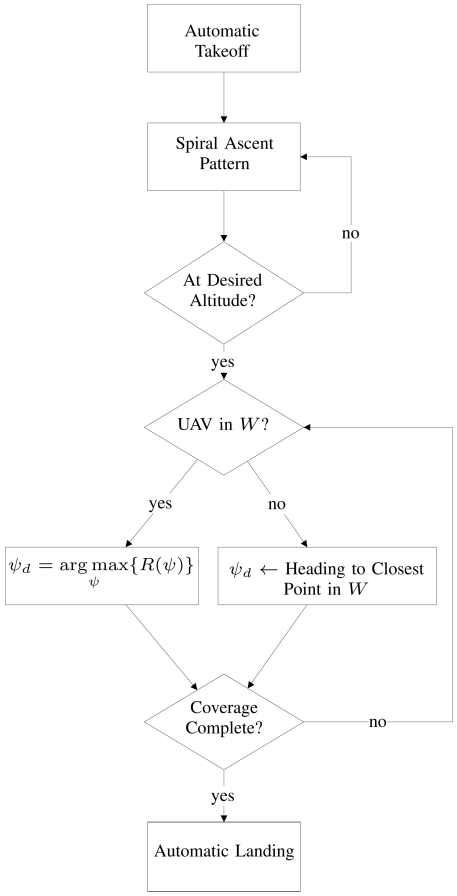

Fig. 5. Flowchart for fully autonomous coverage mission.

with the exception of the camera model block which is linked to each of the others.

Every MOOS community must have a MOOSDB, which handles the marshalling of all data as a central database. The GCS/MOOS interface connects to the UAV through a wireless UDP link and can receive state data and send the desired headings that are reconciled by the IvP optimization directly back to the UAV. The image processing application is built using OpenCV [46] to perform the frame capturing, recording, and automatic target recognition. The state data and camera model are used to geo-reference the targets that are automatically detected. The coverage model is responsible for maintaining the coverage map (see Section III). The camera optimizer optimizes the roll and pitch angles of the camera gimbal to maximize information gain as described in Section V. The IvPHelm is responsible for performing the multiobjective IvP optimization where the behavior *bhv_Information_Gain* is described in Section IV-B and the behavior *bhv_Branch_Entropy* is described in Section IV-C.

A flow chart representing mission progression is shown in Fig. 5. The UAV takes off automatically and ascends to its desired altitude, z_d, and speed, v_d using a spiral ascent pattern. Once at the desired altitude and speed, the area is covered by optimizing the desired heading over the utility function R. If the UAV flies out of the boundary, it is returned to the workspace by choosing the desired heading ψ_d to point to the

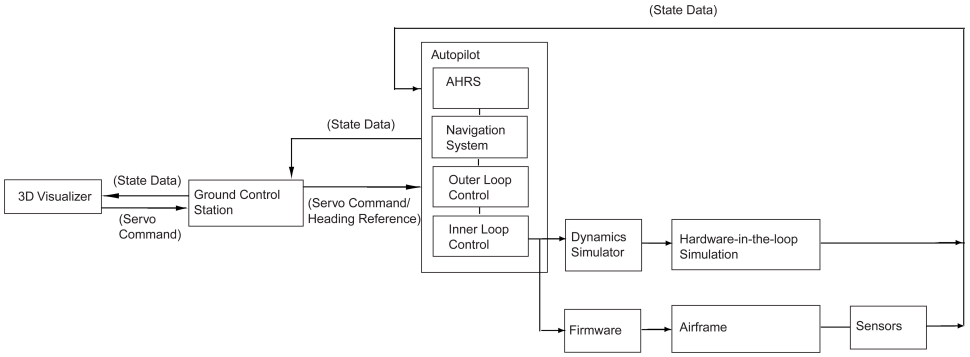

Fig. 6. Block diagram showing system structure for hardware-in-the-loop simulation and real hardware operation.

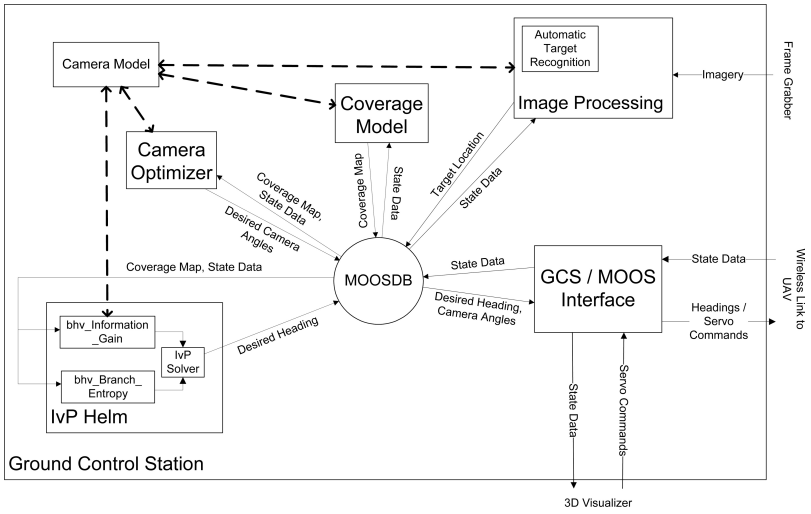

Fig. 7. MOOS community running on the ground control station.

closest point on the environment boundary. Once the coverage objective is complete, the UAV autonomously lands.

When the vehicle is inside the workspace, headings are generated by optimizing the following weighted multiobjective function:

$$\psi_d = \text{argmax}_\psi\{R(\psi)\} = w_B B(\psi) + w_G G(\psi) \tag{16}$$

where R is the total utility shown in Fig. 5, B is the information gain, G is the branch entropy, and w_B and w_G are the respective weights. In general, the function $B(\psi)$ prioritizes headings that will result in the most information gain and the function $G(\psi)$ prioritizes over headings that will direct the vehicle toward unfinished areas of the workspace.

B. Information Gain Objective Function

The Shannon entropy of a continuous RV X, with probability density $p(x)$, is defined as

$$\begin{aligned} H(X) &= -E\{\log p(x)\} \\ &= -\int p(x)\log p(x)dx \end{aligned} \tag{17}$$

and represents a measure of the compactness of a distribution [47].

Considering two RVs X and Z, then the conditional entropy of X given Z is defined as

$$H(X|Z) = -E\{\log p(x|z)\} = -\int p(x|z)\log p(x|z)dx. \tag{18}$$

Now we can define the mutual information, or entropy reduction as

$$\Delta H(X|Z) = H(X) - H(X|Z). \tag{19}$$

The mutual information defines a scalar quantity that represents the amount of information about X contained in Z, or how much the entropy of X will be reduced by the information provided by Z. Typically, X would represent some state vector, and Z would represent some sensor measurement. *A priori*, we do not have direct access to the measurement because it has not been made yet, therefore we cannot evaluate the mutual information directly. As a result, it is useful to define the expected entropy reduction (EER), which represents the expected reduction in entropy that will come about by making measurement Z

$$E_Z[\Delta H(X|Z)] = H(X) - E_Z[H(X|Z)]. \tag{20}$$

The essential aspect of this definition is that it specifies a way of combining sensor measurements additively. Consider

some control action U_t at time t that results in a set of K independent measurements $\{Z_1, Z_2, ..., Z_K\}$. The total expected information gain of U_t can be expressed as

$$B(U_t) = \sum_{k=1}^{K} E_{Z_k}[\Delta H(X|Z_k)]. \quad (21)$$

Recall from (2) that the coverage objective has been formulated in terms of maximizing $p(T_t^i)$ over the entire workspace. The entropy of T_t^i is

$$H(T_t^i) = -p(T_t^i)\log(p(T_t^i)) - (1-p(T_t^i))\log(1-p(T_t^i)). \quad (22)$$

If we take the limit as $p(T_t^i)$ tends to one in (22)

$$\lim_{p(T_t^i)\to 1} H(T_t^i) = 0. \quad (23)$$

This implies that maximizing the PoD at location c^i is equivalent to minimizing the entropy $H(T_t^i)$. As a result, the information gain objective function is formulated in terms of gaining information (reducing entropy) about T_t^i, $i = 1..N$.

To define the information gain objective function, information gained must be formulated as a function of potential headings ψ. This is achieved by defining a path starting at the UAV's current location, (x, y), and traveling a fixed distance, r, at every potential heading ψ. The measurements that will be made can be predicted and then (21) can be used to evaluate the expected information gained from traveling along the prospective path. Let the path to be evaluated be represented by $\mathcal{C}$ and parameterized by s. The path begins at the UAV's current location, (x, y) and moves a distance r at heading ψ

$$\begin{aligned} \mathcal{C} : [0,1] \to \quad & C_{free}, s \to \mathcal{C}(s) \\ \mathcal{C}(0) \quad &= (x, y, \psi) \\ \mathcal{C}(1) \quad &= (x + r\cos(\psi), y + r\sin(\psi), \psi) \\ \mathcal{C}(s) \quad &= (x + \frac{r}{s}\cos(\psi), y + \frac{r}{s}\sin(\psi), \psi). \end{aligned} \quad (24)$$

Let the proposed action, U_t from (21) be defined by following the proposed track. Since r, x, and y are assumed constant, the information gain resulting from following the track can be defined as a function of the heading, ψ. It is noted that some tracks will not be able to be followed in reality due to dynamic constraints of the aircraft; however, this formulation defines an expected benefit of choosing the given reference heading. These desired headings are used as a reference input to an inner loop controller that generates actuator values.

Assume that following the track will result in a series of K pictures $\mathbf{Z} = \{Z_1, Z_2, ..., Z_K\}$. Then, (20) can be used to compute the expected entropy at location c^i in the workspace as a result of any individual measurement $Z_k, k = 1..K$

$$\begin{aligned} E_{Z_k}[\Delta H(T_t^i|Z_k)] = \int p(Z_k)[&-p(T_t^i|Z_k)\log p(T_t^i|Z_k) \\ &- (1-p(T_t^i|Z_k))\log(1-p(T_t^i|Z_k))] \end{aligned} \quad (25)$$

where $p(T_t^i|Z_k)$ is the PoD of a target at location c^i after measurement Z_k. Combining subsequent images of the same location amounts to a sensor fusion problem. In this case, these subsequent looks [34] are considered statistically dependent and can be combined by taking the maximum PoD of any of the looks. PoD values are initialized to 0.5 at time zero which corresponds with maximum entropy. The values of $p(T_t^i|Z_k)$ are generated by the target detection model as described in Section III-C.

The EER at location c^i caused by measurement Z_k follows from (20) as:

$$E_{Z_k}[\Delta H(T_t^i|Z_k)] = H(T_t^i) - E_{Z_k}[(T_t^i|Z_k)]. \quad (26)$$

To determine the expected information gain resulting from performing a single measurement, Z_k, we can sum over all points in the workspace that are in the sensor swath

$$E_{Z_k}[\Delta H(T_t^{1:N}|Z_k)] = \sum_{i \text{ s.t. } c^i \in Z_k} E_{Z_k}[\Delta H(T_t^i|Z_k)]. \quad (27)$$

Finally, the total expected information gain brought about by moving along the path $\mathcal{C}$ can be found by summing the expected information gain from each of the measurements

$$B(\psi) = \sum_{k=1}^{K} E_{Z_k}[\Delta H(T_t^{1:N}|Z_k)]. \quad (28)$$

C. Branch Entropy Behavior

The information gain method has been shown to be effective for solving the path planning problem when *a priori* knowledge of the environment, obstacles, and targets is available [41]. However, it is common that this information will not be available, or will not be completely accurate. In the sensor-driven approach, the information gain B is useful for evaluating the benefits of each of the potential next moves, but when complete coverage is the goal, this approach reduces to a greedy-first search (GFS).

It is necessary to include a parameter in the objective function that helps the UAV achieve its global goal, this is referred to as the branch entropy (BE) and was originally proposed in [4].

The workspace is decomposed into a hexagon cell tiling. Each cell in the decomposition has at most six neighbors, $k = 0..5$ where by convention the neighbor above is number zero incrementing in a clockwise fashion as shown on the hexagon on the right of Fig. 8. The average entropy of the coverage map RVs that fall in each hexagon is used to determine which general areas of the environment are not covered. A formula is derived whereby each neighbor of the cell currently occupied by the UAV is given a value representing the benefit of heading toward that particular cell. The value is determined by how much entropy there is down that branch of the directed acyclic graph (DAG), with priority given to high entropy areas that are nearby. The result is that, by applying a formula on the decomposition and without performing an exhaustive search, the UAV can determine what areas of the map are left to be explored. Fig. 8 shows a workspace in purple, which is covered by a hexagon tiling. The resulting DAG for the UAV at the position indicated on the figure is shown by the arrows. The hexagon tiling is only computed once and the DAG is updated whenever the UAV moves from one hexagon to another.

There will be a value of BE for each neighbor of the current cell C_p as each neighbor has its own branch in the DAG. In

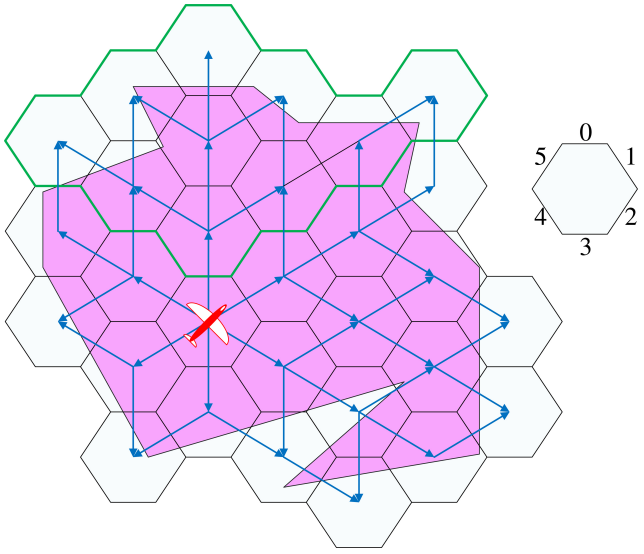

Fig. 8. Workspace shown in purple with hexagon tiling. The DAG generated when the UAV is at the location shown is given by the blue arrows. The hexagon on the right hand side shows the convention for numbering the children or branches of a cell. As an example the cells that are in the branch 0 are outlined in green.

order for the BE to provide the benefits desired, cells that are at higher levels in the graph must be given more weight. For each neighbor, $k = 0..5$, of C_p, the BE, g_k, for a DAG with a total of L levels is given by (29). m_{lk} is the number of nodes in level l of branch k

$$g_k = \frac{\sum_{l=2}^{L}(L-l+1)\frac{\sum_{i=1}^{m_{lk}}\hat{H}_i}{m_{lk}}}{\sum_{l=1}^{L-1} l}. \quad (29)$$

Fig. 9 shows a small example of the transformation from hexagon cells to DAG. The cell labeled C_p is the cell that the UAV is currently in, and the values in all of the other cells represent their average entropies of the coverage map variables that lie in each cell.

The corresponding BE for each of the three neighbors are calculated as

$$\begin{aligned} g_4 &= 1/3((2)(0.6) + (1)(0.1)) = 0.433 \\ g_3 &= 1/3((2)(0.5) + (1)(0.1)) = 0.367 \\ g_2 &= 1/3((2)(0.2) + (1)(1/2)(0.95 + 0.90)) = 0.442. \end{aligned} \quad (30)$$

In this case g_2 is the highest. This algorithm was originally proposed for AUV seabed coverage in [4].

V. Camera Aspect Optimization

In the formulation of the information gain behavior, the optimization takes place over the outer loop reference heading and avoids solving the inverse dynamics model of the vehicle to find the information that will be gained by executing specific control actions. This approach has benefits and shortcomings.

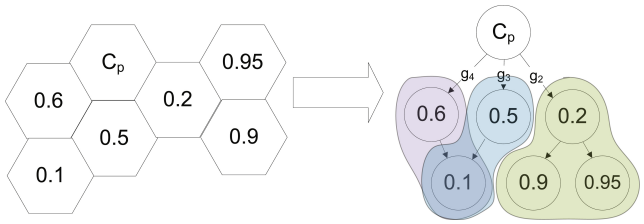

Fig. 9. Transformation from cell to DAG (numbers in cells/nodes represent average cell entropy).

A main benefit is that the typical search horizon problem and rapidly growing search space can be avoided by encapsulating many control actions into one outer loop reference optimization. However, the result is that the trajectory itself is not solved for explicitly and therefore specific locations in the workspace cannot be guaranteed to be covered, only approached. Rigidly connecting the camera to the airframe severely couples the sensor coverage to the vehicle dynamics, which produces additional challenges for planning. To compensate, gimbal systems whereby the relative angle of the onboard camera to the airframe can be controlled are common even for small UAVs [22]. Here we propose to optimize the gimbal angles to compensate for the discrepancy between the desired path and the actual path.

A similar approach to what was presented in Section IV-B can be used to optimize these gimbal angles.

In this case a straight track is defined from the current UAV position in the direction of the current UAV heading and the expected information gain is evaluated over the doman of all gimbal angles. Define the camera gimbal roll and pitch as ϕ_g and θ_g respectively. Once again assuming that following the track will result in K pictures from the camera: $Z_k, k = 1..K$ then (25)–(28) can be used to define optimal desired gimbal roll and pitch, ϕ_g^d, θ_g^d respectively, based on the expected information gain

$$\phi_g^d, \theta_g^d = \arg\max_{\phi_g,\theta_g} B(\phi_g, \theta_g) = \arg\max_{\phi_g,\theta_g} \sum_{k=1}^{K} \bar{I}(W|Z_k). \quad (31)$$

Note that the frequency of this loop should be much higher since the response of the camera gimbal servo motors will be orders of magnitude faster than the response of the dynamics of the aircraft. As a result note that the simulated track length for this inner loop optimization should be much shorter. The domain of the optimization should be consistent with the physical constraints of the gimbal, and the resolution of the search domain should be chosen so that the full optimization can be evaluated in a time period consistent with the desired control frequency.

VI. Experimental Setup

The aircraft used was developed in house by the authors and is briefly describe below. In addition to the airframe a set of sensors and a nested loop control system were added to allow fully autonomous operation.

A. Hardware

The following hardware was used and comprises a combination of off-the-shelf and custom built components.

1) *Airframe:* The airframe selected is a powered glider with a 2.4 meter wing span, as shown in Fig. 1. This airframe is powered using a brushless DC motor with lithium polymer batteries and is stable and efficient.

2) *Actuators:* The UAV has four primary actuators that can be used to control motion. The forward speed of the plane is controlled by the propeller, the roll movement by the deflection of the ailerons, the pitch movement by the deflection of the elevator, and the yaw movement by the deflection of the rudder.

3) *Sensors:* For the aircraft to operate autonomously a minimum knowledge of attitude, position, airspeed, and system health is required. For the UAV to perform automatic target detection a payload sensor is also required.

Attitude: A CHRobotics inertial measurement unit (IMU) was used that includes three solid state Micro-Electro-Mechanical Systems (MEMS) accelerometers, three MEMS gyroscopes and three magnetometers. This device is used to track the orientation of the vehicle as well as accelerations. Vehicle heading and angular rotation are estimated from the fusion of the data from these sensors.

Position: A U-Blox GPS to triangulate the receiver's position based on precisely synchronized signals from the 24 GPS satellites.

Altitude: The altimeter on board the aircraft is a barometric pressure sensor that is sensitive enough to measure the change in absolute pressure from the atmosphere. This sensor is sensitive to climate change; however, it can be fused with GPS altitude estimations.

Airspeed: A Freescale sensor is used to measure the stagnation pressure of the free stream air entering a pitot tube installed under the vehicle's wing. This measurement is critical when operating in the presence of wind since the generation of lift is dependent on the air speed rather than the ground speed.

Health: To operate the vehicle safely, critical information such as power consumption is monitored with voltage and current sensors.

Payload: A daylight camera is installed. The raw imagery is transmitted to the GCS for processing.

4) *Wireless Communications and Telemetry:* There is currently no frequency band allocated to civilian UAV operations. Therefore it is practical to limit ourselves to the commonly available components on unregulated bands. The bands of interest are primarily 2.4 GHz, 900 MHz, and 72 MHz.

The standard for radio controlled models is 72MHz. The live video streams are assigned to the 900MHz channels. The standard wireless module for robotics applications is the WiFi router available on 2.4 GHz.

B. Controller

The autopilot uses nested PID controllers to pilot the UAV as shown in Fig. 10. An outer loop controller is designed to track the heading reference values. The inner loop control is for attitude stabilization which controls the pitch and roll of the airframe through angular rate and Euler angle feedback.

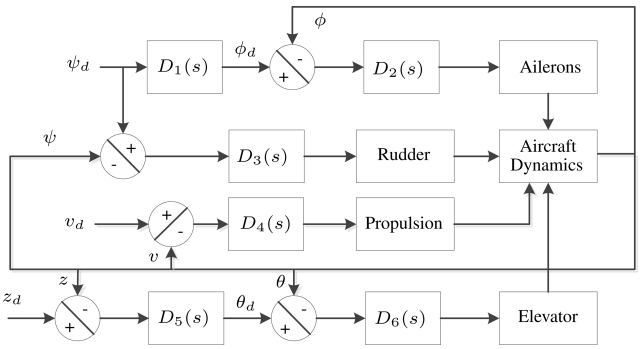

Fig. 10. Nested control system structure.

Each PID controller $D_j(s)$, $j = 1..6$ in Fig. 10 takes the following standard form in the s domain:

$$D_j(s) = k_j^p + \frac{k_j^i}{s} + k_j^d s,\ j = 1..6 \tag{32}$$

where k_j^p,k_j^i and k_j^d are the proportional, integral and derivative gains respectively. The aircraft's equations of motion are linearized about flight conditions and the controller gains are chosen to achieve desired tracking performance in terms of rise time and overshoot.

C. Firmware

The onboard computer consists of three components: the Overo Fire Gumstix [48] running Ubuntu 10.04LTS, its carrier board called Toby, and an extension I/O board with an Atmega 128 microprocessor board called Robostix. Custom firmware was developed for the Robostix. The firmware is used to control the DC motor, the servo motors, to read the GPS, pressure sensors, voltage sensor, and current sensor. In addition, the firmware has a manual override to allow the UAV to be remotely piloted in the event of a malfunction. Data and command signals are sent between the Gumstix and Robostix through I2C and the Gumstix uses a wireless 802.11b card for communication to the GCS.

VII. Results

The proposed method was tested on the bench using a hardware-in-the-loop (HWIL) setup as well as on the real airframe.

A. Simulation

The HWIL simulator was used to test the proposed methods before field trials during a number of different operating conditions. A sample output path is shown in Fig. 11 where the mission was considered complete when an average PoD of 0.995 was obtained. A lawn mower plan was also generated to cover the same workspace and is shown in Fig. 12. The actual UAV path is shown in white and preplanned tracks are shown in red. The UAV does not exactly follow the tracks resulting in some areas being missed as can be seen in the area coverage

Fig. 11. Path taken to cover workspace using HWIL simulation for proposed sensor-driven approach.

Fig. 12. Path taken to cover workspace using preplanned lawn mower path in HWIL simulation.

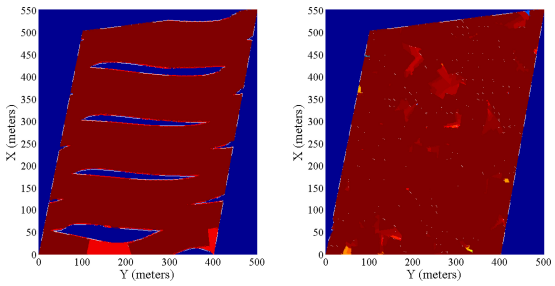

Fig. 13. Area coverage over the workspace for lawn mower (left) and proposed sensor-driven approach (right). More red indicates higher PoD.

plots shown in Fig. 13. The final average PoD for the lawn mower mission was 0.945. In both cases there was a constant wind of 3 m/s applied from the south.

In Fig. 14, the average PoD and entropy, desired heading, camera and UAV roll, and UAV altitude are plotted over the course of the mission shown in Fig. 16. It was determined during experimentation that for the fixed-wing UAV it is only necessary to optimize the gimbal angle in the roll direction and that gimbal optimization in the pitch direction did not provide any substantial benefits. Specifically regarding Fig. 14(c), it is noted that in most cases the camera optimization algorithm preferentially stabilizes the camera to face downward, however, in cases when the UAV is flying over previously covered areas, the camera angle is chosen to cover new areas

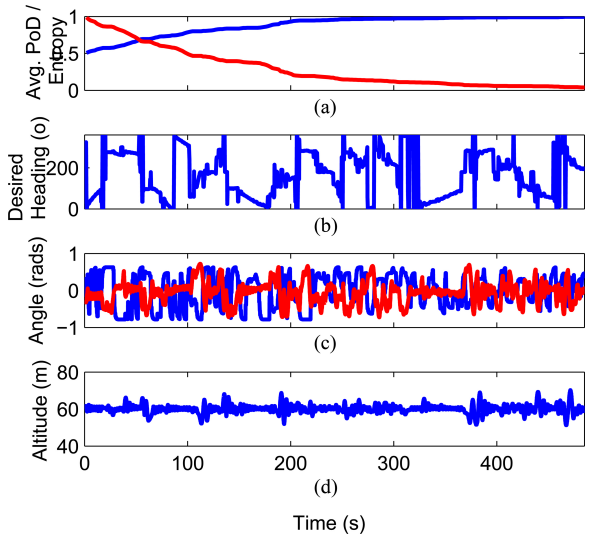

Fig. 14. Data over the course of the HWIL simulated mission shown in Fig. 11. (a) Average PoD (blue) and average entropy (red) are shown to converge to 1 and 0 respectively. (b) Desired heading output from the multiobjective optimization. (c) UAV roll (red) and the camera gimbal angle in the roll direction (blue). (d) UAV altitude.

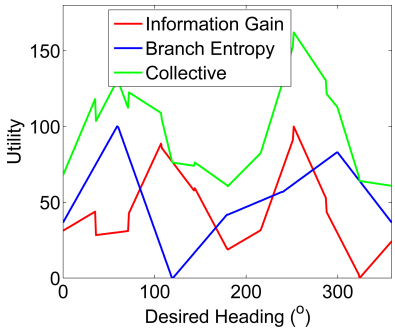

Fig. 15. Information gain, branch entropy, and collective IvP functions at the timestep in Fig. 11 indicated by a light blue dot.

if possible. For example, consider the case at approximately l50 s where the vehicle flies very close to an already traversed path. In this case, the camera gimbal angle is moved all the way to its negative threshold value ($\pi/4$) such that the camera may view some new terrain.

In Fig. 15, the information gain, branch entropy, and collective IvP functions are shown at the timestep in Fig. 11 indicated by the blue dot (t=130s). The weighting for the information gain behavior is 0.8 and the weighting for the branch entropy behavior is 1. The information gain behavior has two peaks, one at about 260^o and one at about 110^o indicating the directions that will result in the maximum expected reduction of entropy in the short term. The branch entropy behavior also has two peaks, but the bigger peak is at 60^o because the entire north-east corner of the map is uncovered at this point. However, the second smaller peak nearly coincides with the information gain peak so the heading that trades off the short term and long term benefits is chosen, in this case 252^o, as can also be verified from Fig. 14(b). Notice in Fig. 11 that the UAV makes a sharp turn toward the west directly after the point indicated.

Fig. 16. Path taken by fixed-wing UAV performing coverage.

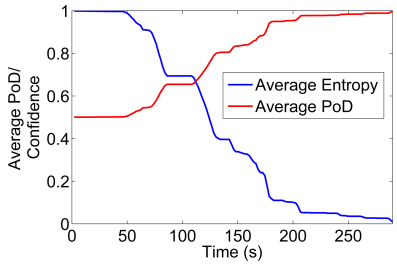

Fig. 17. Convergence of the average PoD and average entropy over the course of the mission.

B. Real World Experiments

We have field tested our vehicle a number of times in Lincoln, New Brunswick. The UAV was able to cover an area of approximately 300 m by 200 m in the limited operating area specified by our Special Flight Operations Certificate (SFOC). The UAV was able to cover the area to an average PoD of 99.8% based on the actual vehicle trajectory and the projective camera model as shown in Fig. 16. In this case the mission lasted 290s (first $\approx$ 50s is spent on ground) and has a total path length of 2535m. In Fig. 17, the average PoD is shown to converge to 1 and the average entropy is shown to converge to 0.

It should be explicitly stated that this sensor-driven approach makes no claim on optimality compared with classical offline planning methods. Referring to the results it can be seen that the paths cannot be considered to be optimal since there is sensor overlap. However, the major benefit of this is increased autonomy and robustness.

VIII. Discussion

In this section we make a qualitative comparison of our method with a standard lawn mower type approach [17] simulated in Fig. 12.

1) Coverage overlap: It is acknowledged that the proposed method does yield some overlap in coverage when paths crossed. However, preplanning a lawn mower pattern to achieve coverage notwithstanding disturbances requires placing tracks conservatively close together, which will also result in high overlap.
2) Online versus offline: Planning in the proposed method is entirely online.
3) Convergence guarantee: The proposed method is guaranteed to converge to complete coverage due to the branch entropy behavior which can find areas left to be covered. A lawn mower path can guarantee coverage at the planning stage but not necessarily in the field.
4) Computational requirements: Relatively high but low enough to run at 4 Hz on the GCS. Lawn mower paths are planned offline and therefore require no computation once mission is underway.

In the absence of any independent control (either stabilization or optimization) of the camera gimbal angles, performance of the algorithm proposed here would likely degrade. In that case, an alternative approach that explicitly incorporates vehicle dynamics or searches over vehicle configurations would be preferable. Possible examples could include an approach similar to [21] but adapted to the Chinese postman problem. However, solving these kino-dynamic planning problems optimally results in a search space that grows exponentially with the horizon length [49].

IX. Conclusion

We have presented a sensor-driven approach to UAV area coverage planning. Information theory has been used to provide a planner that is capable of efficiently planning paths online to achieve coverage. The outer loop heading reference is generated by trading off greedy objectives, quantified with information gain, and global objectives, quantified through branch entropy. In addition, the angles of the camera gimbal are optimized in real-time to maximize the rate of coverage.

Realistic simulations, done with a hardware-in-the-loop simulator, and actual field trials have demonstrated the effectiveness of the approach.

In summary, we have implemented and demonstrated a fully autonomous fixed-wing UAV system capable of performing area coverage missions. The essential benefit of the approach is that sensor data is used during flight to model the coverage and make control decisions making the system much more robust to disturbances.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable feedback.

References

[1] H. Choset, "Coverage of known spaces: The boustrophedon cellular decomposition," *Autonom. Robots*, vol. 9, no. 3, pp. 247–253, 2000.
[2] Y. Li, H. Chen, M. J. Er, and X. Wang, "Coverage path planning for UAVs based on enhanced exact cellular decomposition method," *Mechatron.*, vol. 21, no. 5, pp. 876–885, 2011.
[3] J. Tisdale, Z. Kim, and J. Hedrick, "Autonomous UAV path planning and estimation," *IEEE Robot. Autom. Mag.*, vol. 16, no. 2, pp. 35–42, Jun. 2009.
[4] L. Paull, S. Saeedi, M. Seto, and H. Li, "Sensor-driven online coverage planning for autonomous underwater vehicles," *IEEE/ASME Trans. Mechatron.*, to be published.
[5] H. Choset, W. Burgard, S. Hutchinson, G. Kantor, L. E. Kavraki, K. Lynch, *et al.*, *Principles of Robot Motion: Theory, Algorithms, and Implementation*. Boston, MA, USA: MIT Press, 2005.
[6] S. LaValle, "Motion planning," *IEEE Robot. Autom. Mag.*, vol. 18, no. 1, pp. 79–89, Mar. 2011.
[7] J.-C. Latombe, *Robot Motion Planning*. Norwell, MA, USA: Kluwer Academic, 1991.

[8] H. Choset, "Coverage for robotics—A survery of recent results," *Ann. Math. Artif. Intell.*, vol. 31, nos. 1–4, pp. 113–126, 2001.
[9] M. Bosse, N. Nourani-Vatani, and J. Roberts, "Coverage algorithms for an under-actuated car-like vehicle in an uncertain environment," in *Proc. IEEE ICRA*, 2007, pp. 698–703.
[10] T. Lee, S. Baek, Y. Choi, and S. Oh, "Smooth coverage path planning and control of mobile robots based on high-resolution grid map representation," *Robot. Autonom. Syst.*, vol. 59, no. 10, pp. 801–812, 2011.
[11] E. Acar and H. Choset, "Sensor-based coverage of unknown environments: Incremental construction of Morse decompositions," *Int. J. Robot. Res.*, vol. 21, no. 4, pp. 345–367, 2002.
[12] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull, "Morse decompositions for coverage tasks," *Int. J. Robot. Res.*, vol. 21, no. 4, pp. 331–344, 2002.
[13] E. Acar, H. Choset, and J. Lee, "Sensor-based coverage with extended range detectors," *IEEE Trans. Robot.*, vol. 22, no. 1, pp. 189–198, Feb. 2006.
[14] Y. Gabriely and E. Rimon, "Spanning-tree based coverage of continuous areas by a mobile robot," *Ann. Math. Artif. Intell.*, vol. 31, pp. 77–98, 2001.
[15] T. Oksanen and A. Visala, "Coverage path planning algorithms for agricultural field machines," *J. Field Robot.*, vol. 26, no. 8, pp. 651–668, 2009.
[16] J. Jin and L. Tang, "Coverage path planning on three-dimensional terrain for arable farming," *J. Field Robot.*, vol. 28, no. 3, pp. 424–440, 2011.
[17] A. Xu, C. Viriyasuthee, and I. Rekleitis, "Optimal complete terrain coverage using an unmanned aerial vehicle," in *Proc. IEEE ICRA*, 2011, pp. 2513–2519.
[18] A. Tsourdos, B. White, and M. Shanmugavel, "Cooperative path planning of unmanned aerial vehicles." New York, NY, USA: Wiley, 2011.
[19] L. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *Amer. J. Math.*, vol. 79, no. 3, pp. 497–516, Jul. 1957.
[20] C. Hanson, J. Richardson, and A. Girard, "Path planning of a Dubins vehicle for sequential target observation with ranged sensors," in *Proc. ACC*, 2011, pp. 1698–1703.
[21] K. Savla, E. Frazzoli, and F. Bullo, "Traveling salesperson problems for the dubins vehicle," *IEEE Trans. Autom. Contr.*, vol. 53, no. 6, pp. 1378–1391, Jul. 2008.
[22] M. Quigley, M. Goodrich, S. Griffiths, A. Eldredge, and R. Beard, "Target acquisition, localization, and surveillance using a fixed-wing mini-UAV and Gimbaled camera," in *Proc. IEEE ICRA*, 2005, pp. 2600–2605.
[23] A. Xu and G. Dudek, "A vision-based boundary following framework for aerial vehicles," in *Proc. IEEE/RSJ Int. Conf. IROS*, 2010, pp. 81–86.
[24] S. C. Spangelo, E. G. Gilbert, A. T. K. P. T. Kabamba, and A. R. Girard, "Periodic energy-optimal path planning for solar-powered aircraft," in *Proc. AIAA Guidance, Navigat. Control Conf.*, 2009, pp. 1–15.
[25] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948.
[26] A. Cameron and H. Durrant-Whyte, "A Bayesian approach to optimal sensor placement," *Int. J. Robot. Res.*, vol. 9, no. 5, pp. 70–88, 1990.
[27] G. Hoffmann and C. Tomlin, "Mobile sensor network control using mutual information methods and particle filters," *IEEE Trans. Autom. Control*, vol. 55, no. 1, pp. 32–47, Jan. 2010.
[28] A. Ryan and J. K. Hedrick, "Particle filter based information-theoretic sensing," *Robot. Autom. Syst.*, vol. 58, pp. 574–584, Jan. 2010.
[29] A. Klesh, P. Kabamba, and A. Girard, "Path planning for cooperative time-optimal information collection," in *Proc. Amer. Control Conf.*, 2008, pp. 1991–1996.
[30] P. Skoglar, J. Nygards, and M. Ulvklo, "Concurrent path and sensor planning for a UAV—Towards an information based approach incorporating models of environment and sensor," in *Proc. IEEE/RSJ Int. Conf. IROS*, 2006, pp. 2436–2442.
[31] M. Faied, P. Kabamba, B. Hyun, and A. Girard, "Path planning for optimal classification," in *Proc. IEEE 51st Annu. Conf. CDC*, 2012, pp. 520–527.
[32] B. Hyun, P. T. Kabamba, and A. R. Girard, "Optimally-informative path planning for dynamic Bayesian classification," *Optimiz. Lett.*, vol. 6, no. 8, pp. 1627–1642, 2012.
[33] R. Mannadiar and I. Rekleitis, "Optimal coverage of a known arbitrary environment," in *Proc. IEEE ICRA*, 2010, pp. 5525–5530.
[34] F. Bourque and B. Nguyen, "Optimal sensor configurations for rectangular target detection," in *Proc. ICRA*, 2011, pp. 455–460.
[35] A. Cassandra, L. Kaelbling, and J. Kurien, "Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation," in *Proc. IEEE/RSJ Int. Conf. IROS*, vol. 2, 1996, pp. 963–972.
[36] D. R. Thompson, D. Wettergreen, and F. J. Peralta, "Autonomous science during large-scale robotic survey," *J. Field Robot.*, vol. 28, no. 4, pp. 542–564, 2011.
[37] J. Le Ny and G. Pappas, "On trajectory optimization for active sensing in Gaussian process models," in *Proc. IEEE Conf. Decision Control*, Dec. 2009, pp. 6286–6292.
[38] A. Klesh, P. Kabamba, and A. Girard, "Optimal path planning for uncertain exploration," in *Proc. ACC*, 2009, pp. 2421–2426.
[39] S. Ghosh, J. Burdick, A. Bhattacharya, and S. Sarkar, "Online algorithms with discrete visibility—Exploring unknown polygonal environments," *IEEE Robot. Autom. Mag.*, vol. 15, no. 2, pp. 67–76, Jun. 2008.
[40] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.
[41] C. Cai and S. Ferrari, "Information-driven sensor path planning by approximate cell decomposition," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 3, pp. 672–689, Jun. 2009.
[42] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *Int. J. Robot. Res.*, vol. 30, no. 1, pp. 56–79, Jan. 2011.
[43] Z. W. Kim and R. Sengupta, "Target detection and position likelihood using an aerial image sensor," in *Proc. IEEE ICRA*, 2008, pp. 59–64.
[44] M. R. Benjamin, H. Schmidt, P. M. Newman, and J. J. Leonard, "Nested autonomy for unmanned marine vehicles with MOOS-IvP," *J. Field Robot.*, vol. 27, no. 6, pp. 834–875, Nov.–Dec. 2010.
[45] M. Benjamin, J. Curcio, and P. Newman, "Navigation of unmanned marine vehicles in accordance with the rules of the road," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2006, pp. 3581–3587.
[46] Opencv. (2012). *Open source computer vision* [Online]. Available: http://opencv.willowgarage.com
[47] B. Grocholsky, "Information-theoretic control of multiple sensor platforms," Ph.D. dissertation, Australian Centre Field Robot., Univ. Sydney, Sydney, Australia, 2002.
[48] Gumstix. (2012) [Online]. Available: http://www.gumstix.com
[49] E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," *AIAA J. Guidance, Control, Dynam.*, vol. 25, pp. 116–129, 2000.

Liam Paull received the B.Eng. degree in computer engineering from McGill University, Montreal, QC, Canada, in 2004, and the Ph.D. degree in electrical and computer engineering from the University of New Brunswick, Fredericton, NB, Canada, in 2013, where his research focussed on seabed coverage with autonomous underwater vehicles.

He is currently a Post-Doctoral Associate at the Computer Science And Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA. His current research interests include autonomous vehicle planning and navigation, cooperative localization, multiagent systems, and simultaneous localization and mapping.

Carl Thibault received the B.Sc. degree in mechanical engineering from the University of New Brunswick, Fredericton, NB, Canada, in 2009, specializing in mechatronics and numerical control of machines. He is currently pursuing the master's degree with the Electrical And Computer Engineering Department at the same university.

His current research interests include unmanned vehicle systems, vertical take-off and landing aircraft, control and applications of robotics.

Amr Nagaty received the B.Sc. degree in mechatronics engineering from the German University in Cairo, Cairo, Egypt, in 2011. He is currently pursuing the M.Sc. degree in electrical engineering at the University of New Brunswick, Fredericton, NB, Canada.

His current research interests include control theory, state estimation, and vision and unmanned systems.

Mae Seto received the B.A.Sc. degree in engineering physics—electrical, and the Ph.D. degree in mechanical engineering from the University of British Columbia, Vancouver, BC, Canada, in 1987 and 1996, respectively.

She is currently a Senior Defence Scientist at Defence Research and Development Canada, Dartmouth, NS, Canada, and Leader of the Mine and Harbour Defence Group. She is also an Adjunct Professor of Mechanical Engineering and Computer Science at Dalhousie University, Halifax, NS, Canada, and the University of New Brunswick, Fredericton, NB, Canada. Previously, she was an NSERC Industrial Post-Doctoral Fellow doing research on autonomous underwater vehicles and unmanned surface vehicles. She has published numerous papers in her research areas. Her current research interests include intelligent autonomy for marine autonomous vehicles and systems, underwater vehicle and tow body dynamics, multiagent systems, and underwater acoustics.

Howard Li received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, ON, Canada.

Currently, he is an Associate Professor in the Department of Electrical and Computer Engineering, University of New Brunswick, Fredericton, NB, Canada. He was also associated with Atlantis Systems International, Canada, Defence Research and Development Canada, and Applied AI Systems Inc., Canada, to develop unmanned ground vehicles, unmanned aerial vehicles, autonomous underwater vehicles, and mobile robots for both domestic and military applications. He is a registered Professional Engineer in the Province of Ontario, Canada. His current research interests include linear control, non-linear control, intelligent control, distributed control, unmanned vehicles, mechatronics, robotics, multiagent systems, artificial intelligence, motion planning, and simultaneous localization and mapping.