2011 IEEE International Conference on Robotics and Automation
Shanghai International Conference Center
May 9-13, 2011, Shanghai, China

Optimal Complete Terrain Coverage using an Unmanned Aerial Vehicle

Anqi Xu, Chatavut Viriyasuthee, and Ioannis Rekleitis

Abstract— **We present the adaptation of an optimal terrain coverage algorithm for the aerial robotics domain. The general strategy involves computing a trajectory through a known environment with obstacles that ensures complete coverage of the terrain while minimizing path repetition. We introduce a system that applies and extends this generic algorithm to achieve automated terrain coverage using an aerial vehicle. Extensive experimental results in simulation validate the presented system, along with data from over 100 kilometers of successful coverage flights using a fixed-wing aircraft.**

I. Introduction

This paper presents a realization of an optimal terrain coverage algorithm for an Unmanned Aerial Vehicle (UAV). The problem formulation consists of generating a path that completely covers the area in a bounded environment, while:

- avoiding a set of obstacle regions with arbitrary shape
- preventing from flying over previously covered regions
- starting from and ending at the same location

The problem of coverage using an aerial vehicle has many applications, including: environmental inspection, search and rescue, surveillance, etc. Providing an optimal trajectory in terms of minimum travel distance is very important especially in time-critical scenarios such as search and rescue, while it also helps to minimize operational cost in general.

Our approach to terrain coverage is to decompose a 2-dimensional environment with obstacles into simple regions using the Boustrophedon Cellular Decomposition (BCD) algorithm [1]. This representation leads to a systematic motion pattern that covers each region individually and completely, by using a back and forth motion. The original Boustrophedon family of algorithms guarantees the complete coverage of an *unknown environment* with *no bounds on the distance travelled*. Mannadiar and Rekleitis [2] introduced a variant of the algorithm that ensures the complete coverage of an environment with *known obstacles*, that guarantees *optimality in terms of distance travelled*. It is important to note that these algorithms assume that the vehicle is either holonomic or redundant, meaning that it can either turn in place or move in all directions independently.

In contrast, in this paper we extend the optimal coverage algorithm [2] for the general class of non-holonomic robots. We compute a set of waypoints outlining the desired coverage path, which is then realized by a robot controller that accounts for the vehicular dynamics of the target robot. Because waypoint-based control can be less maneuverable than direct velocity-based steering, this coverage approach can no longer guarantee minimal travel distance for all robotic vehicles. Thus, we generalize the criterion of *coverage optimality* within a robot-independent context as *minimizing the distance travelled over previously covered regions*.

The authors are with the School of Computer Science, McGill University, [anqixu,pvirie,yiannis]@cim.mcgill.ca
The authors gratefully appreciate the financial support of the National Science and Engineering Research Council (NSERC) of Canada.

Fig. 1. Our fixed-wing UAV landing after a successful coverage session.

This paper's main contribution consists of addressing practical issues related to the deployment of our theoretical complete coverage algorithm to control a fixed-wing UAV shown in Fig. 1. Our field trials produced numerous complete aerial coverage sessions resulting in over 100 km of total flight distance. We also conducted extensive testing using a simulated version of the aircraft to determine the impact of various system parameters on the overall coverage quality.

II. Related Work

Choset and Pignon [1] first introduced the Boustrophedon Cellular Decomposition (BCD) family of algorithms for coverage-based path planning in an unknown environment. Their general coverage strategy extended the Seed Spreader algorithm [3] for producing back-and-forth sweeping motions through the free space. Acar et al. [4] further developed BCD-based coverage with experimental verification for a variety of control Morse functions. Mannadiar and Rekleitis [2] presented a variant of this technique which guaranteed optimal coverage for an arbitrary known environment. Previous algorithms proposed by Zheng et al. guaranteed a performance of at most eight times the optimal cost for known terrains [5], using additional constraints for the free space configuration. The reader is referred to [2], [6] for information on other general coverage strategies and for an extensive survey of coverage research.

The aerial robotics community has developed a number of systems that either directly achieve terrain coverage or use similar techniques to address other applications such as search and rescue. Gaudiano et al. [7] investigated swarm intelligence and UAV control, with extensions to collaborative aerial coverage. Agarwal et al. [8] proposed a multi-UAV

978-1-61284-380-3/11/$26.00 ©2011 IEEE

coverage solution for a rectilinear polygonal environment that focused on assigning partitions of the environment proportionally based on each vehicle's capabilities, although computing explicit trajectories was not within their scope. Maza and Ollero [9] presented a similar technique for distributed coverage of a polygonal environment with no obstacles. Their algorithm divided the free space into simple regions and focused on selecting a per-region coverage pattern that minimized the number of turns, which is an interesting solution to optimize performance.

Ahmadzadeh et al. conducted extensive research on coverage-style aerial surveillance [10], [11] with the goal of minimizing the amount of uncovered area given certain time constraints. Their work addressed explicit concerns for vehicle dynamics and sensor range, and explored a wide variety of different solutions to achieve surveillance.

Cheng et al. presented a solution to 3-D complete coverage by projecting the environment into non-planar surfaces [12]. Their approach focused on designing a time-optimal trajectory within this manifold, which is devoid of obstacles.

Several solutions to search-related problems within uniform distributions presented motion strategies that are readily extendable to achieve coverage [13]–[15]. These systems employed probabilistic models of the abstract problem to derive robust algorithms, although their use of approximation techniques could potentially impede the optimality of coverage. In addition, addressing the completeness of coverage using search-related strategies remains an open problem.

III. Aerial Coverage

The problem of covering the area of a bounded environment while avoiding a known set of arbitrarily shaped obstacles is solved using a two-stage hierarchical solution. First, an off-line analysis of the environment decomposes the free space into a set of simple regions, called *cells*, and determines an Eulerian circuit through all connected cells. During the on-line stage, back-and-forth sweeping motions are generated to cover individual cells while following the Eulerian circuit. Previous developments [2] primarily focused on the off-line components of the algorithm, and assumed that the target vehicle could move in all directions independently. This section begins with a review of the off-line analysis technique, and then elaborates on additional components required to ensure complete and optimal coverage for non-holonomic robots, such as a fixed-wing UAV.

The off-line algorithm breaks down free space using the Boustrophedon Cellular Decomposition (BCD) method. BCD-based coverage was originally proposed for the context of a terrestrial robot exploring through an unknown environment with obstacles. Cells of the BCD are populated as the robot sweeps back and forth through the free space. When the vehicle encounters a new obstacle in its path, it adopts a *locally optimal* strategy by choosing to cover the uncovered cell closest to its current location. Depending on the obstacle layout however, this strategy may force the robot to travel through previously covered areas in the process.

On the other hand, in aerial coverage one typically deals with known environments, since satellite maps of the terrain can be readily obtained. The definition of obstacles is also more flexible than in the terrestrial context. Assuming that the vehicle flies at a fixed altitude, certain terrains and structures such as mountains, high-rise buildings, and even restricted airspace still pose as (pseudo-) physical obstructions to the robot. More commonly however, there are application-specific areas where there is no need for coverage; for example, in a wilderness search operation, villages and large bodies of water do not need to be covered. The presented work deals only with the latter type of obstacles.

A. Off-line Analysis Algorithm

Off-line analysis of the environment consists of partitioning the free space into cells using the BCD method, and then generating an Eulerian circuit through all connected cells by solving a linear programming formulation.

The input to the system is a bitmap representation of the environment that differentiates between obstacles and free space. The BCD algorithm divides the free space into cells by sweeping a line through the bitmap and identifying locations, known as *critical points*, where the connectivity of the free space on the line changes by the presence of an obstacle. Without loss of generality, the direction of the sweep is assumed to move along the x-axis. The position of critical points and the boundaries of cells are stored respectively as vertices and edges of the Reeb graph structure $G = \langle V, E \rangle$.

The Reeb graph is used as input to the Chinese Postman Problem [16] to calculate an Eulerian circuit, which consists of a closed path traversing through every cell at least once. This is achieved by doubling selected edges of the Reeb graph, although no edge needs to be duplicated more than once [16]. The Eulerian circuit is the solution to the linear programming instance described in Eq. 1:

$$\text{Minimize } z = \sum_{e \in E} (c_e x_e) \tag{1}$$

subject to the following constraints:

$$\sum_{e \in E} (a_{ne} x_e) - 2w_n = k_n, \ \forall \ n \in V;$$

$$x_e \in \mathbb{Z}^+, \ \forall \ e \in E;$$

$$w_n \in \mathbb{Z}^+, \ \forall \ n \in V;$$

where:

$\sum_{e \in E} (a_{ne} x_e)$ is the number of added edges to node $n \in V$. For the result to be Eulerian, an odd number of edges must be added to nodes with odd degree and an even number of edges must be added to nodes with even degree;

a_{ne} is 1 if node n meets edge e, and 0 otherwise;

x_e is the number of added copies of edge e in the solution;

w_n is an integer variable that will force $\sum_{e \in E} (a_{ne} x_e)$ to be odd for odd nodes and even for even nodes;

k_n is 1 for nodes with odd degree, and 0 otherwise;

c_e is a real number representing the cost of edge e.

To prevent repeat coverage, cells corresponding to doubled Reeb graph edges are split into non-overlapping top and bottom sub-cells. One of the simplest methods for splitting cells is to divide along the mid-points between the top and

Fig. 2. Result of the off-line analysis phase for a 13 km $\times$ 10 km region. Darkened areas delineate obstacles; white lines depict cell boundaries, red lines and cell numbers indicate Eulerian circuit.

bottom boundaries. Unfortunately, because the mid-points at the left and right extremities of the cell do not necessarily correspond to the locations of critical points, the robot must sometimes backtrack through the covered area to reach the critical point and subsequently the next cell. We employed a more efficient cell-splitting scheme which guarantees that sub-cells will share the same critical points as their parent, by interpolating between the bounding critical points *and* between the top and bottom boundaries of the original cell.

At the end of the off-line analysis phase, the resulting Eulerian circuit outlines a cyclic path through all connected cells in the environment. Fig. 2 shows the decomposed cells and sub-cells for a sample environment, where the corresponding numbers illustrate the Eulerian circuit and consequently, the order of coverage.

B. Per-Cell Coverage Pattern

Given an Eulerian circuit through all cells of the free space, each cell can be covered by generating back-and-forth Seed Spreader motions, which is very well documented in the literature [1], [3], [17]. The resulting trajectory is represented as a set of waypoints forming parallel sweep lines through the cell. The *direction of coverage* is defined as the orientation of the coverage progression as the robot moves through the free space, which is orthogonal to all of the sweep lines.

A crucial parameter required by the Seed Spreader algorithm is the *coverage footprint*, which measures the spacing between consecutive parallel sweep lines in the back-and-forth motion path. Different factors contribute to the definition of the coverage footprint, depending on the intended application. For example, if the robot is to record atmospheric samples, then the footprint width would correspond to the sensor's range. In the context of visual coverage, where an UAV continuously records *"bird's-eye view"* pictures of the terrain, the footprint width depends on the camera's field of view, the UAV's altitude, and the required amount of image overlap. In this paper the focus is solely on visual coverage.

It is known that the footprint width of the Seed Spreader algorithm can affect the overall quality of coverage [2]. In particular, the footprint width determines whether the robot will arrive at the top or the bottom boundary of a completed cell. Depending on the location of the next cell in the Eulerian circuit however, the robot might need to backtrack through previously covered terrain. One method to prevent these disjunctions is to assume a *default* footprint width and find the best configuration out of all possible Eulerian circuits. Unfortunately, enumerating through all possible Eulerian circuits is a #P-complete problem [18], and the best configuration might still contain disjunctions. More importantly, in certain situations it is not possible to estimate the footprint width, for example if the vehicle's altitude during visual coverage is unknown prior to deployment.

(a) (b)

Fig. 3. Results of off-line analysis using (a) the default direction of coverage, and (b) after aligning with the dominant axis of the free space.

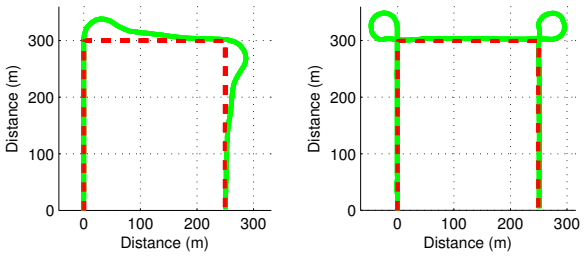
Our implementation used a more efficient method which eliminates disjunctions completely, by adding an extra sweep line in the back-and-forth motion pattern. This guarantees a continuous path across cell boundaries, and the process can be done during flight. Although this method does not improve the length of the flight path, in the case of visual coverage the increased overlapping regions in images might be beneficial to post-processing activities such as image stitching.

C. Direction of Coverage

The BCD algorithm sweeps through the bitmap in the direction of coverage. This direction impacts the shapes of cells in the Reeb graph and subsequently the length of the sweep lines in the Seed Spreader pattern. The quality of the coverage path can be improved by changing the direction of coverage, since fewer longer straight paths are preferred over many shorter ones. This is especially desirable when working with non-holonomic vehicles [19], since they cannot perform in-place turns. Fig. 3 illustrates that rotating the map prior to the construction of the BCD can reduce the presence of sharp cusps and make cells more elongated, thus minimizing the number of turns. We now propose three different strategies for selecting the direction of coverage, each of which rotates the map before running the BCD algorithm.

Given that boundaries are shared between obstacles and free space, one method is to set the direction of coverage orthogonal to the dominant edge orientation for obstacle boundaries, which would arguably produce longer sweep lines as a result.

An alternative strategy is to align the direction of coverage directly with the distribution of the free space, under the assumption that the length of sweep lines will be maximized along the dominant axis of the free space. Given the eigenspace decomposition for free space pixel coordinates, the direction of coverage is set orthogonally to the heading of the eigenvector with the largest eigenvalue. This ensures that cells produced by the BCD are mostly elongated, which



(a) (b)

Fig. 4. Two strategies for steering non-holonomic vehicles along the ideal path (shown in red): process the waypoints outlining the desired trajectory using a *greedy* path planner (a) or add *curlicue* orbits at corners (b).

in turn minimizes the number of turns in the coverage path.

Finally, certain environmental conditions can also affect the quality of coverage. For example, crosswind can constantly divert UAVs from their desired flight path. By aligning the direction of coverage to be perpendicular to the dominant wind heading, most of the crosswind force will be shifted to the lateral wind component.

D. Non-Holonomic Vehicle Control

The coverage algorithm generates desired coverage paths in a robot-independent manner by using a set of waypoints, which is then given as input to a robot motion controller. The completeness of coverage assumes that the robot travels through this path exactly; despite being trivial for robots that can turn in-place, additional logic is needed to steer a non-holonomic robot through the specified path efficiently.

Certain robots including fixed-wing aircrafts offer integrated motion planners that take waypoints as input. These planners typically employ a *greedy* strategy, by issuing a maximum turning rate until the vehicle aligns its heading to the destination. Fig. 4(a) illustrates this behavior when the vehicle turns around to transition between consecutive sweep lines. Processing waypoints corresponding to the coverage trajectory directly using a greedy path planner might violate the footprint width and compromise the completeness of coverage. In the illustrated example, although the UAV had a minimum turning radius of 45 m, the greedy planner caused a miss-distance of about 80 m at the beginning of the second sweep line. One way to resolve this issue is to adjust the footprint width to compensate for these deviations.

As an alternative, Fig. 4(b) depicts a corner-steering strategy that allows non-holonomic robots to remain on the designated coverage path. Rather than carrying out a turn directly, the robot is steered in a circular orbit to manually align it with the upcoming segment. The orbit's size is determined from the vehicle's minimum turning radius, and the orbit is positioned such that it intersects with both line segments. This *curlicue* strategy is well suited for most turning angles, although it is not required for mild turns where the vehicle can adjust its course without deviation. In contrast, if the turning angle is too steep, i.e. near 0°, then the resulting orbit placement will veer the vehicle away from the desired path. These rare occurrences can be addressed by relocating the orbit to be tangential to the cusp of the turn.

Fig. 5. Our robotic platform is a commercial fixed-wing UAV with an on-board autopilot microprocessor and a gimbal-mounted camera.

IV. Hardware Setup

The unmanned aerial vehicle shown in Fig. 5 is a rigid body fixed-wing plane commercially available from Procerus® Technologies. Its 1 meter wingspan is built using expanded polypropylene foam, which is useful for absorbing the impact upon touchdown. A brushless electric motor powered by batteries can drive the plane at average ground speeds of 14 m/s and for flight durations of up to 30 minutes.

This aircraft is controlled by an on-board autopilot microprocessor, which receives instructions from the ground control software wirelessly. The autopilot is connected to various sensors, including an accelerometer, a gyroscope, a pitot tube (for measuring pressure and wind speed), and a GPS unit. In addition, an on-board camera attached to a pan-tilt gimbal device transmits live analog video stream at 30 Hz via a separate radio frequency. This UAV can operate in many modes, ranging from joystick-based manual control to fully autonomous waypoint-based navigation.

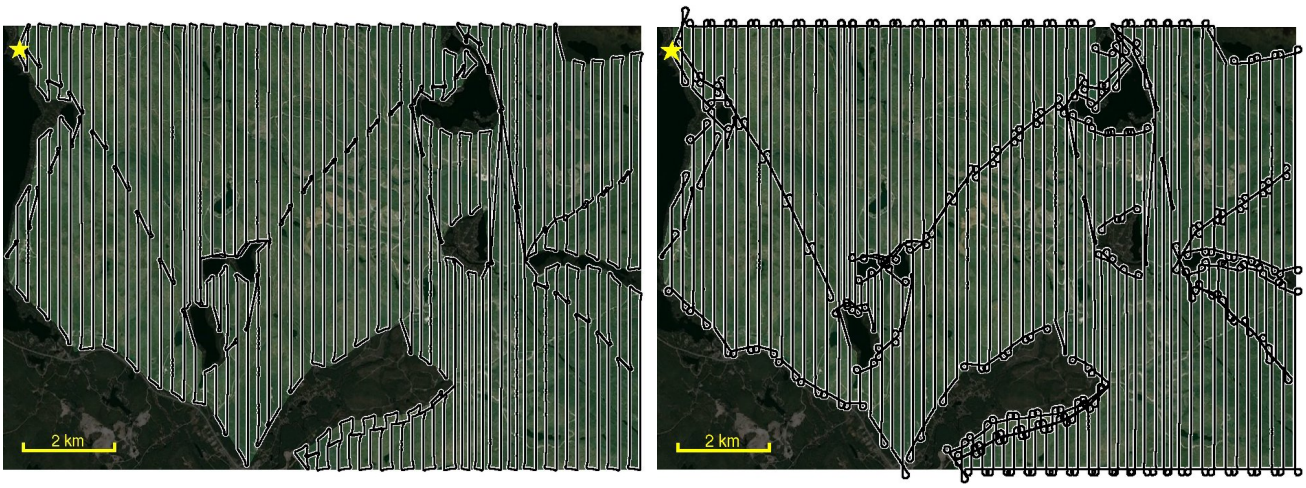
The presented coverage system is implemented in C++ under Linux, on-board a 1.66 GHz dual-core laptop, and communicates with a networked ground control computer.

All of the experiments presented in this paper were conducted at various fixed altitudes. During flight, the gimbal's orientation is continuously regulated to align it perpendicular to the ground plane. This removes the need to transform frames using projective geometry, and also establishes a uniform pixel density in the acquired images. The reader is referred to [20] for a discussion on the gimbal control algorithm as well as other information on this UAV.

V. Experimental Results

Extensive testing of the optimal coverage system was conducted using the Aviones [21] 6-DOF UAV simulation software, and also during several field trials. These sessions validated the optimality of the realized trajectory resulting from the implemented coverage system. Each experiment also investigated the impacts of a specific parameter or aspect of the system on the quality of coverage.

The performance of the system is quantified both by the total trajectory length and the elapsed time for each complete coverage session. Although these two metrics are strongly correlated, the total flight distance provides a somewhat isolated assessment of the coverage path; in contrast, the flight duration represents the quality of coverage for each



(a) (b)

Fig. 6. Simulated coverage paths for a 13 km × 10 km region at 300 m altitude with no wind. Star denotes start and end of flight. The UAV travelled (a) 590.9 km in 11 h 56 m using the greedy path planner, and (b) 740.7 km in 14 h 56 m using the curlicue strategy.

particular session, since it accounts for internal factors such as battery levels as well as external conditions such as wind.

A. Large Scale Simulation

Assessing the scalability of the algorithm, Fig. 6 depicts two complete aerial coverage sessions over a 13 km × 10 km environment containing multiple curved and polygonal obstacles These sessions were carried out in simulation both because the real UAV lacked sufficient battery to sustain such a flight, and more importantly because it allows the two control schemes, namely the greedy and curlicue controllers, to be evaluated under an optimal environment without wind. To remain faithful to the real UAV's capabilities, the coverage footprint was determined based on an operational altitude of 300 m and with a 46° field of view of the camera.

As mentioned previously in Sec. III-D, one method to compensate for path deviations resulting from using the greedy path planner is to decrease the *effective* coverage footprint width. Given these operational altitude and camera settings, and knowing that the UAV's minimum turning radius in the absence of wind is 50 m, the effective coverage footprint is about 70% of the original width. Unfortunately, simulation results using this setting nearly doubled the length of the original path, even in the absence of wind.

Rather than compensating for the deviations caused by the greedy path planner, the two control techniques are compared using the same footprint width to estimate the difference in performance between a non-holonomic vehicle, such as a fixed-wing aircraft, using curlicue patterns versus a holonomic robot, such as a helicopter, following the original waypoint path. Since the fixed-wing simulator was used in both cases however, the latter results can only approximate the performance of a holonomic vehicle.

The flight paths recorded from both experiments are provided in Fig. 6. Under ideal environmental conditions, The elapsed time and total flight distance both indicate a 25% increased penalty for using the curlicue strategy; this provides an estimate of the additional resources required to achieve complete coverage in a large scale environment.

TABLE I

SIMULATED COVERAGE RESULTS FOR A 1 KM × 0.6 KM REGION AT 150 M ALTITUDE UNDER DIFFERENT WIND CONDITIONS.

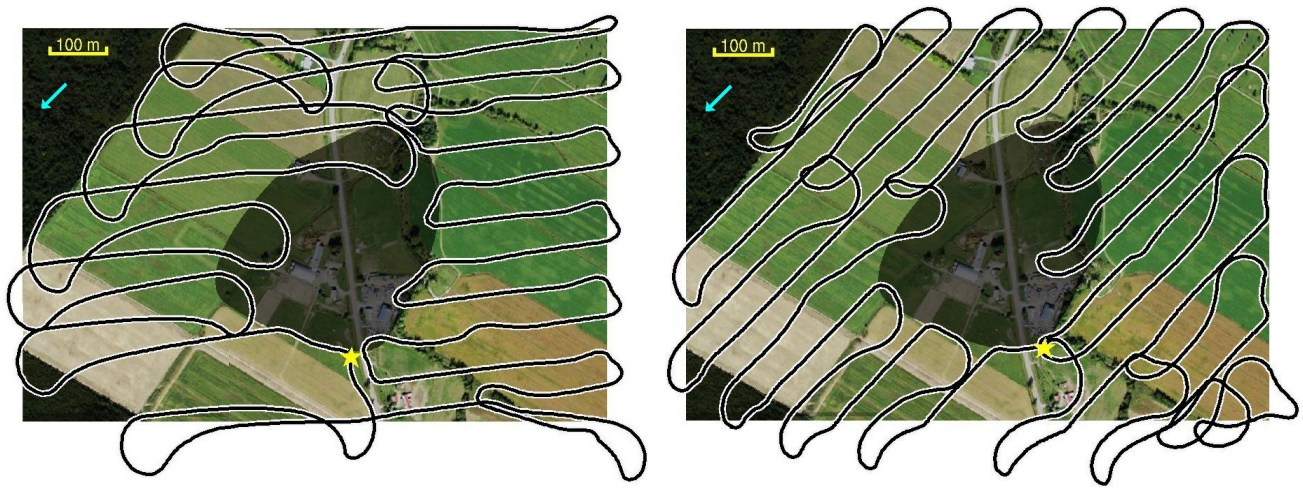
	Alignment of Coverage Direction			
	Default	Obs. Edges	Free Space	Wind
Time (No Wind)	21 m 54 s	21 m 59 s	21 m 26 s	–
Dist. (No Wind)	13.61 km	13.69 km	13.36 km	–
Time (7 m/s Wind)	28 m 57 s	25 m 23 s	27 m 13 s	27 m 39 s
Dist. (7 m/s Wind)	15.14 km	13.24 km	14.46 km	15.19 km

B. Alignment of Coverage Direction

The optimal direction of coverage, assuming that such a value exists, is strongly dependent on the obstacle layout for each particular environment. Nevertheless, several simulated coverage sessions were conducted on a terrain with one obstacle after aligning the direction of coverage using different strategies. Although the chosen environment is arguably overly simplistic, the coverage path generated using this terrain was well matched to the UAV's capabilities.

Table I enumerates the elapsed times and flight distances resulting from changing the direction of coverage using various alignment methods. In the first set of experiments, the UAV simulation was carried out in the absence of wind to determine the isolated effects of aligning the direction of coverage perpendicular either to the dominant obstacle edge orientation or to the dominant axis of the free space distribution. Unfortunately, the resulting paths exhibit nearly identical performance, arguably due to the simplicity of the chosen terrain. On the other hand, this suggests that the direction of coverage might not be an important factor for the scale of operations relevant to the specific UAV hardware.

The same three experiments were then repeated after introducing a static 7 m/s wind, where the wind direction was manually set to be perpendicular to each direction of coverage, to maximize deviations from the desired trajectory. The static wind setting was anecdotally observed to produce similar results compared to real flight sessions. In addition, a separate session was carried out after aligning the direction of coverage to be perpendicular with the wind heading. Looking at the second row of elapsed time results in Table I, the



(a) (b)

Fig. 7. Simulated coverage paths for a 1 km × 0.6 km region at 150 m altitude with 7 m/s wind (arrow shows direction). Star denotes start and end of flight. Our UAV travelled (a) 14.46 km in 27 m 13 s under alignment with free space, and (b) 15.19 km in 27 m 39 s under alignment with wind direction.

presence of wind appeared to have consistently increased the duration of coverage in all four sessions. Interestingly, in the final experiment where the rotation forced the wind to be perpendicular to the direction of coverage, the vehicle's velocity oscillated between 20 m/s along the wind and 10 m/s against the wind, which delayed the coverage overall. Thus, one should consider both the costs and benefits of the wind alignment technique, especially in time-critical scenarios.

Investigating the flight distances, one of the sessions (alignment with obstacle edges) actually resulted in a shorter coverage trajectory than previous results under no wind conditions. Upon further inspecting the data, the crosswind can be seen in Fig. 7(a) to have significantly warped the back-and-forth sweeping motions. In contrast, the flight trajectory resulting from alignment with the wind direction in Fig. 7(b) displayed minimal deviation from the original set of waypoints determined by the coverage algorithm.

As predicted previously, no single alignment method resulted in unilateral performance improvements, even for a simple one-obstacle terrain. For settings with little or no wind factor, the choice of alignment ultimately depends on whether the resulting cell shapes lead to a reduction in the number of turns. On the other hand, it was observed that aligning the direction of the sweep lines to be parallel to a strong wind force will extend the duration but will also ensure that the vehicle remains on course.

C. Global vs Local Optimality

This set of simulations compared the cell-selection technique for the coverage algorithm presented in this paper against the locally optimal approach of choosing the closest uncovered cell. Because the latter can lead to different paths depending on the first visited cell, seven locally-optimal coverage sessions were conducted on a 1 km^2 terrain with two obstacles to exhaustively test all possible scenarios. An additional session was carried out using the proposed algorithm in this paper, with identical configurations.

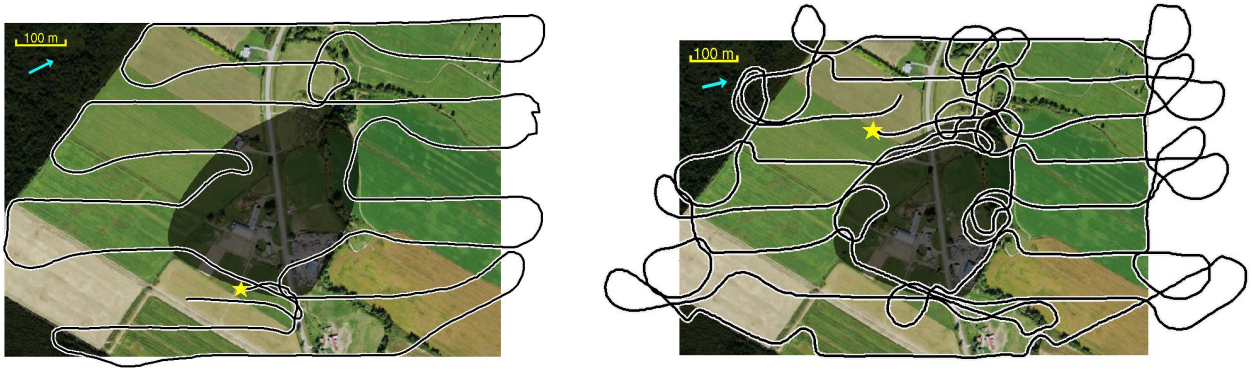
The proposed globally optimal algorithm completed coverage in 24 m 04 s with a total flight distance of 15.01 km, whereas the locally optimal sessions lasted between 25 m 46 s and 27 m 29 s with traveled distances ranging from 16.05 km to 17.13 km. These results indicate that the globally optimal cell traversal strategy produced an average time and distance improvement of 10%, even for the simple environment tested. Although more testing is needed to be conclusive, the gap in performance would arguably grow with the presence of more obstacles and would decrease when covering larger terrains.

D. UAV Field Trial

Extensive field experiments over farmland terrain have resulted in a total flight distance of 213.8 km during numerous coverage sessions, with 118.9 km accounting for completed coverage for various obstacle placements. All of the configuration parameters used in the simulated coverage sessions, including those in this paper, were obtained from real data collected during these extended field trials.

The final experiment revisits the comparison between the greedy robot controller and the curlicue technique, although this time the coverage sessions were performed using an actual fixed-wing UAV, above a farmland terrain affected by a 5-8 m/s wind. Observing the resulting trajectories and flight data in Fig. 8, the elapsed time and distance both indicate an 85% increased penalty for using the curlicue strategy. One of the main causes for the large performance gap can be seen in the curlicue path – rather than flying in circular orbits, the wind constantly pushed the UAV off course.

The collected aerial footage was used to estimate the completeness of coverage by analyzing the overlap percentage among video frames. The flight session using the greedy controller resulted in 66.6% single coverage, 28.3% repeated coverage, and 5.1% missed coverage, whereas the curlicue session resulted in 35.1% single coverage, 64.1% repeated coverage, and 0.8% missed coverage. The large amount of repeated coverage in the latter session is primarily caused by the circular orbits generated by the curlicue controller, as seen in Fig. 8(b). Also, the reduction of the footprint widths to accommodate an integer number of sweep lines per cell caused a certain amount of overlap in both sessions. Furthermore, because the wind deviated the UAV off course, the collected footage skipped over a small amount of terrain.



(a) (b)

Fig. 8. Actual coverage paths from an UAV field trial, for a 1 km × 0.7 km region at 150 m altitude with 5-8 m/s wind (arrow shows direction). Star denotes start and end of flight. Both directions of coverage were pre-aligned with obstacle edges. Our UAV travelled (a) 9.10 km in 12 m 48 s using the greedy path planner, and (b) 17.25 km in 23 m 42 s using the curlicue strategy.

The two coverage sessions were replicated in simulation using a fixed wind factor computed from field data. The UAV travelled 10.2 km in 16 m 19 s using the greedy path planner, and 19.3 km in 30 m 38 s using the curlicue controller. Comparing to the field trial results, the slightly increased distance and duration values are attributed to a combination of the inaccuracies in the replication of the setup and of the simulation. More importantly however, the performance gap between the two motion controllers remain at about 87%, which is near identical to the results obtained in field.

VI. Conclusion

In this paper we presented a realization of an optimal coverage algorithm on a fixed-wing unmanned aerial vehicle. We designed a hierarchical, robot-independent system that decomposes a bitmap representation of the environment into free space regions, computes an optimal circuit through these connected regions, and produces a motion path for complete coverage which accounts for the range of the deployed sensor. We propose two robot-specific controllers for carrying out the generated motion path while accounting for the dynamics of a non-holonomic vehicle.

Our experiments included over 100 km of successful visual coverage flights with a real fixed-wing vehicle, together with thousands of kilometers of flight in simulation. Extensive testing validated the robustness and efficiency of the proposed approach, and discovered the effects on the quality of coverage of different motion planners, of the direction of coverage, and of environmental factors such as wind.

We are currently investigating the effectiveness of alternative motion patterns such as sawtooth and spiral shapes for covering free space regions. We are also continuously expanding our experimental repertoire for a wider range of system parameters and obstacle configurations. Finally, ongoing efforts to integrate this system with various terrestrial and naval vehicles will ultimately lead to heterogeneous multi-robot collaborative coverage.

References

[1] H. Choset and P. Pignon, "Coverage Path Planning: The Boustrophedon Cellular Decomposition," in *In Proc. of the Int. Conf. on Field and Service Robotics*, Canberra, Australia, 1997.

[2] R. Mannadiar and I. Rekleitis, "Optimal Coverage of a Known Arbitrary Environment," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, Anchorage, AK, May 2010, pp. 5525–5530.

[3] V. J. Lumelsky, S. Mukhopadhyay, and K. Sun, "Dynamic path planning in sensor-based terrain acquisition," *IEEE Trans. on Robotics and Automation*, vol. 6, no. 4, pp. 462–472, Aug. 1990.

[4] H. Choset, "Coverage of Known Spaces: The Boustrophedon Cellular Decomposition," *Autonomous Robots*, vol. 9, pp. 247–253, 2000.

[5] X. Zheng, S. Jain, S. Koenig, and D. Kempe, "Multi-Robot Forest Coverage," in *In Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, Edmonton AB, Canada, Aug. 2005, pp. 3852–3857.

[6] H. Choset, "Coverage for robotics - A survey of recent results," *Annals of Mathematics and Artificial Intelligence*, vol. 31, pp. 113–126, 2001.

[7] P. G. et. al., "Swarm Intelligence: a New C2 Paradigm with an Application to Control of Swarms of UAVs," in *ICCRTS Command and Control Symposium*, 2003.

[8] A. Agarwal, L. Hiot, N. Nghia, and E. Joo, "Parallel region coverage using multiple UAVs," in *IEEE Aerospace Conference*, 2006, p. 8.

[9] I. Maza and A. Ollero, "Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms," *Distributed Autonomous Robotic Systems*, pp. 221–230, 2007.

[10] A. Ahmadzadeh, J. Keller, A. Jadbabaie, and V. Kumar, "An Optimization-based Approach to Time Critical Cooperative Surveillance and Coverage with Unmanned Aerial Vehicles," in *Int. Symposium on Experimental Robotics*. Citeseer, 2006.

[11] A. Ahmadzadeh, A. Jadbabaie, V. Kumar, and G. Pappas, "Multi-UAV cooperative surveillance with spatio-temporal specifications," in *45th IEEE Conf. on Decision and Control*, 2006, pp. 5293–5298.

[12] P. Cheng, J. Keller, and V. Kumar, "Time-optimal UAV trajectory planning for 3D urban structure coverage," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. IEEE, 2008, pp. 2750–2757.

[13] F. Bourgault, T. Furukawa, and H. Durrant-Whyte, "Optimal search for a lost target in a Bayesian world," in *Field and Service Robotics*. Springer, 2006, pp. 209–222.

[14] T. Furukawa, F. Bourgault, B. Lavis, and H. Durrant-Whyte, "Recursive Bayesian search-and-tracking using coordinated UAVs for lost targets," in *IEEE Int. Conf. on Robotics and Automation*, 2006.

[15] B. DasGupta, J. Hespanha, J. Riehl, and E. Sontag, "Honey-pot constrained searching with local sensory information," *Nonlinear Analysis*, vol. 65, no. 9, pp. 1773–1793, 2006.

[16] J. Edmonds and E. L. Johnson, "Matching, Euler tours and the Chinese postman," *Mathematical Programming*, vol. 5, pp. 88–124, 1973.

[17] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull, "Morse Decompositions for Coverage Tasks," *The Int. Journal of Robotics Research*, vol. 21, no. 4, pp. 331–344, April 2002.

[18] G. Brightwell and P. Winkler, "Note on Counting Eulerian Circuits," *CoRR*, vol. cs.CC/0405067, 2004.

[19] W. Huang, "Optimal line-sweep-based decompositions for coverage algorithms," in *In Proc. the IEEE Int. Conf. on Robotics and Automation*, vol. 1, 2001, pp. 27–32.

[20] A. Xu and G. Dudek, "A Vision-Based Boundary Following Framework for Aerial Vehicles," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Taipei, Taiwan, Oct. 2010.

[21] "Aviones: UAV Flight Simulator," http://aviones.sourceforge.net. Accessed: 02/05/11.