Energy-aware Coverage Path Planning of UAVs

Carmelo Di Franco, Giorgio Buttazzo
Scuola Superiore Sant'Anna, Pisa, Italy
E-mail: c.difranco@sssup.it, g.buttazzo@sssup.it

***Abstract*—Coverage path planning is the operation of finding a path that covers all the points of a specific area. Thanks to the recent advances of hardware technology, Unmanned Aerial Vehicles (UAVs) are starting to be used for photogrammetric sensing of large areas in several application domains, such as agriculture, rescuing, and surveillance. However, most of the research focused on finding the optimal path taking only geometrical constraints into account, without considering the peculiar features of the robot, like available energy, weight, maximum speed, sensor resolution, etc. This paper proposes an energy-aware path planning algorithm that minimizes energy consumption while satisfying a set of other requirements, such as coverage and resolution. The algorithm is based on an energy model derived from real measurements. Finally, the proposed approach is validated through a set of experiments.**

I. Introduction

Unmanned Aerial Vehicles (UAVs) are being used in many application domains for surveying and inspecting large areas to identify specific features of interests that otherwise should be detected at higher costs and times. For instance, UAVs are used in agriculture for detecting the state vegetation and plan for a timely intervention, [1], [2], [3], for surveying wooded area for fire prevention [4], inspecting industrial plants [5], and in several other situations, such as handling disasters and rescuing people in difficult to reach environments.

All these applications require the survey of a given area using specific sensors that are mission dependent. Example of sensors are video cameras, multispectral/thermal/3D cameras, temperature/humidity/pressure/ sensors, proximity sensors, and so on. UAVs can be divided in two main categories: fixed-wing and multi-rotors. Fixed-wing UAVs have the main advantages that can cover large distances and carry higher payloads, but cannot focus on a scene for long time. For this type of task, multirotor UAVs are more appropriate, since can be used as mobile cameras pointing desired details; however, they have the inconvenience of a lower battery duration and a lower payload.

For the particular problem of image reconstruction of geographical zones, a lot of research has been done for finding the optimal path that achieves a complete coverage of the area of interest. This problem, is known as coverage path planning (CPP). Most of the work on CPP, however, has taken only geometrical constraints into account without considering the peculiar features of the UAV, (e.g., the available energy, the weight, the maximum speed) and other mission requirements, (e.g., the spacial resolution of the acquired images). This paper proposes an energy-aware path planning algorithm that minimizes energy consumption while satisfying a set of other requirements, such as coverage and resolution. The algorithm is based on an energy model derived from real measurements. Finally, the proposed approach is validated through a set of experiments.

Contributions. This papers has two main contributions. First, an energy model is derived from real measurements to find the power consumption as a function of the UAV dynamic in different operating conditions. Second an energy-aware algorithm is presented for finding the path that reduces energy consumption while satisfying coverage and resolution requirements.

Paper organization. The rest of the paper is organized as follows. Section II presents the related work. Section III introduces system model and the relation among the states variables. Section IV describes how to find the speed that minimizes the energy of a given path. Section V presents the path planning algorithm. Section VI reports a set of experimental results carried out to validate the proposed approach. Section VII states our conclusions and future work.

II. Related Work

The problem of coverage path planning has been extensively studied by several authors. Among them, Galceran and Carreras investigated in detail the Coverage Path Planning (CPP) [6] showing pros and cons of several methods like cellular, grid-based, graph based, neural-network with online or off-line computation and for known or unknown areas. Maza et al. [7] proposed an algorithm to divide the whole area taking into account UAVs relative capabilities and initial locations. In particular, they used the maximum range of the UAV as a measure of their capabilities. After the area decomposition each UAV has to compute the sweep direction which minimizes the number of turns needed along a back-and-forth pattern. Öst in his thesis [8] provided an analysis of two different types of search methods (back-and forth and spiral) for convex-concave areas. He stated pros and cons of the two search methods combined with some area decomposition algorithms. Barrientos et al. [9] proposed a one-phase automatic task partitioning manager for team of UAVs based on negotiation among the vehicles, considering their state and capabilities. Santamaria et al. [10] proposed a path planning algorithm for multiple heterogeneous UAVs that also consider different sensors footprints. Bast and Hert [11] investigated the problem of partitioning an arbitrary polygon into a minimum number of convex pieces taking into account that the resulting subspaces need to be reasonable for robotic applications. They showed that partitioning an area with minimal cut length is NP-hard and proposed a polynomial algorithm that produces a non-optimal but reasonable partitioning. Huang [12] proposed an optimal line-sweep-based decomposition algorithm that minimizes the amount of time needed to cover an area with obstacles. He uses dynamic programming to find an optimal

area decomposition by assuming to know the area boundaries and obstacles inside.

A different approach has been taken by Lawrance and Sukkarieh [13], who formulated the atmospheric energy gain produced by the wind as a path planning problem and presented an algorithm capable of generating energy-gain trajectories using both static and dynamic soaring. Al-Sabban et al. [14] exploited wind energy to extend the flight duration of an UAV during the route from a starting point to another. Roberts et al. [15] estimated an energy model of a hover-capable flying robot and proposed an algorithm that mitigates the energy consumption in an indoor aerial exploration by using ceiling attachment as a means for preserving energy while maintaining the camera contact with the target.

Mei et al. [16] investigated the problem of the deployment of mobile wheel robots with energy and timing constraints. They proposed a speed-management method for optimizing the robot speeds and used energy constraints and area constraints, such as obstacles, to find the optimal number of robots to deploy. They also constrained the maximum amount of time available for performing the task. However, their only considered the energy consumption on a constant speed and did not investigate the energy consumption due to accelerations in general trajectories.

III. System model

This paper considers a quad rotor equipped with a video camera mounted on a stabilizer, which compensates for small rotation displacements that may be experienced by the UAV during navigation. The objective of the mission is to reconstruct the image of a given area with a spatial resolution R no lower than R_d, expressed in pixels/cm.

We assume that images are taken by a video camera having the following parameters:

- Angle of view (AOV), α (expressed in radians), that is, the angular extent of a given scene that is imaged by the camera;
- Image Resolution (I_x, I_y), expressed in pixels for both image sides;
- Aspect ratio ρ, that is, the ratio between the image width and image height ($\rho = I_x/I_y$);
- Minimum sampling period T_s^{min}, that is, the minimum interval between two consecutive shots.

An UAV flying at a given height h from the ground acquires an image corresponding to a specific portion of the area, called *projected area*. The size of the projected area depends on the height h and the angle of view (AOV) α, as illustrated in Figure 1. If the camera is exactly parallel to the ground (this assumption can be done considering the camera stabilizer), the size (L_x, L_y) of the projected area can be computed as

$$\begin{cases} L_x = 2h \cdot \tan\left(\frac{\alpha}{2}\right) \\ L_y = L_x/\rho \end{cases} \tag{1}$$

Hence, the spacial resolution R obtained by taking a picture

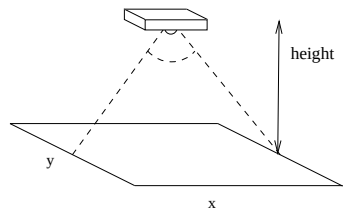

Fig. 1. Projected area of a camera with AOV α placed at height h.

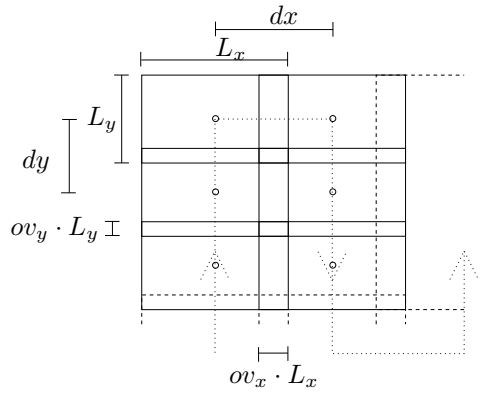

Fig. 2. Projected areas overlapped both on the horizontal and vertical side. The center of the rectangles are the waypoints computed during a back-and-forth pattern.

at height h is

$$R = \frac{I_x}{L_x} = \frac{I_x}{2h \cdot \tan\left(\frac{\alpha}{2}\right)}. \tag{2}$$

By substituting equation (2) in the inequality imposed by the mission requirement ($R \geq R_d$) we have that

$$h \leq \frac{I_x}{2R_d \cdot \tan\left(\frac{\alpha}{2}\right)}. \tag{3}$$

Hence, the mission requirement imposes a constraint on the maximum height the UAV can fly, which is

$$h_{max} = \frac{I_x}{2R_d \cdot \tan\left(\frac{\alpha}{2}\right)}. \tag{4}$$

To perform image reconstruction by photogrammetry, the area of interests has to be decomposed into rectangles of size (L_x, L_y) and the UAV has to be programmed on a trajectory that passes through their centers. The entire path is stored as a list of coordinates, called waypoints, and the UAV moves from a waypoint to the next until the end of the list. The complete path includes other parameters, as the coordinates in GPS, the height, the delays between waypoints, the speed, etc.

It is important to observe that, to correctly merge the acquired images, each projected area has to be overlapped as shown in Figure 2. The amount of overlap can be chosen by the user and can be different on each side. The horizontal and vertical overlaps are denoted as ov_x and ov_y, respectively, and are expressed as a number in [0,1], where 0 means no overlap, and 1 full overlap. The distance between the centers of two adjacent areas becomes $d_x = L_x(1 - ov_x)$ in the horizontal direction, and as $d_y = L_y(1 - ov_x)$.

Note that the sampling period T_s of the camera (i.e. the interval between two consecutive shots) imposes another constraint on the maximum speed of the UAV. In particular, the space covered by the UAV between two consecutive image acquisitions must satisfy the minimum specified overlap, that is $v \cdot T_s \leq L_y(1 - ov_y)$. Therefore,

$$v_{max} = \frac{L_y(1 - ov_y)}{T_s}. \tag{5}$$

For the sake of clarity, Table I summarizes the main parameters used in this section.

Variable	Name
α	angle of view (rad)
f	focal length (mm)
I_x	image width (pixels)
I_y	image height (pixels)
ρ	aspect ratio
T_s	camera sampling period
R	spacial resolution (pixels/cm)
ov_h	horizontal overlap (pixels)
ov_v	vertical overlap (pixels)
h	height (m)

TABLE I. LIST OF THE MAIN SYSTEM PARAMETERS.

A. Energy model

Given the large variety of drones, each with specific physical characteristics, like weight, type of power supply, propellers, etc., deriving a general parametric energy model that can be used to predict the energy consumption in different operating conditions is a hard task. In this work, we propose a method that can be used to model and analyze the energy consumption of a specific drone as a function of its speed and operating conditions.

To derive an energy model suited for the analysis, we performed a set of experiments aimed at understanding how the energy consumption is affected by the different operating conditions, such as speed, horizontal and vertical accelerations. The drone used for the experiment is an IRIS quadrotor with a GoPro camera mounted on a Gimbal stabilizer, controlled by a PX4 autopilot board. The Iris weighs about 1.3 Kg, and is equipped with four 850 Kv motors and powered with a 3S lipo battery 11.1 V 5.5 Ah. The stabilizer uses two brushless motors to ensure that the camera always points with high precision to the specified direction. The GoPro camera can be used at three different resolutions (5-7-12 Megapixel) and two different AOV ($94, 4$-122.6 deg). The PX4 board runs the open-source APM ardupilot.

In a first experiment, we programmed the drone to run at the maximum acceleration and deceleration, monitoring the speed from the onboard GPS and the absorbed current from the control board. The consumed power was then derived by multiplying the absorbed current by the supply voltage, for each speed. Figure 3 shows the speed and the power acquired under maximum acceleration, along with the corresponding fitted curves. Similarly, Figure 4 shows the same quantities acquired under maximum deceleration.

Thus, the energy consumed to vary the speed from v_1 to v_2 with a given acceleration a can be computed as

$$E_a = \int_{t_1:v=v_1}^{t_2:v=v_2} P_a(t)\, dt. \tag{6}$$

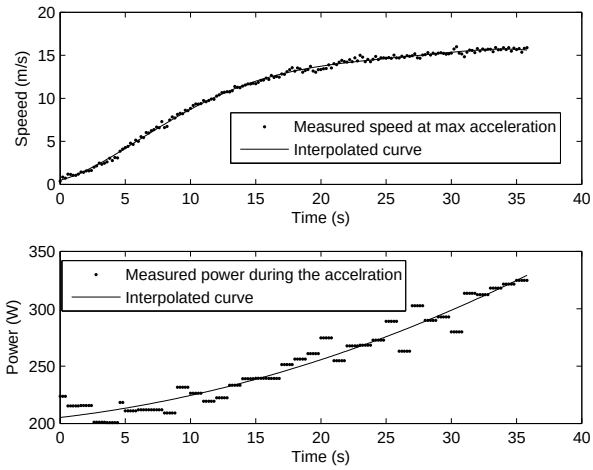

Fig. 3. Speed and power consumption acquired during maximum acceleration.

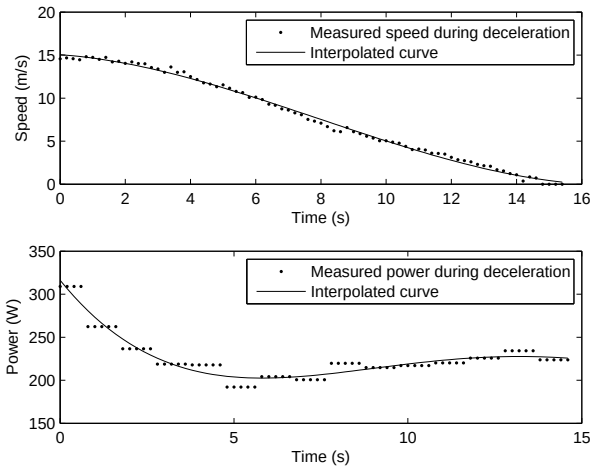

Fig. 4. Speed and power consumption acquired during maximum deceleration.

In a second experiment we derived the power consumption as a function of the speed in different flight conditions, such as horizontal flight, climbing, descending, and hovering. The results of this experiment are reported in Figure 5, which also shows the fitted curve. Note that, since the climbing and descending operation is always performed at a constant speed, the corresponding power consumption is plotted as a single point in the graph corresponding to that speed. Also note that the power consumed during hovering corresponds to the the point in the graph for $v = 0$. The energy consumed in a straight flight to cover a distance d at a constant speed v can be computed as

$$E_v = \int_0^{d/v} P(v)\, dt = P(v)\frac{d}{v}. \tag{7}$$

The energy consumed during climbing and descending to cover

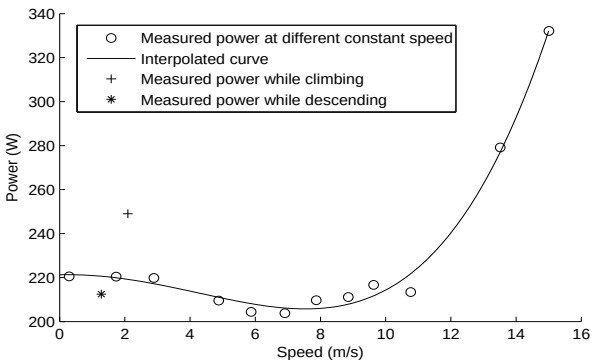

Fig. 5. The fitted curve of the power consumption as a function of the speed during a straight flight. The figure also shows the measured power while the UAV climbs, descends, and hovers. These values are plotted as single points since the speed during these operation is constant

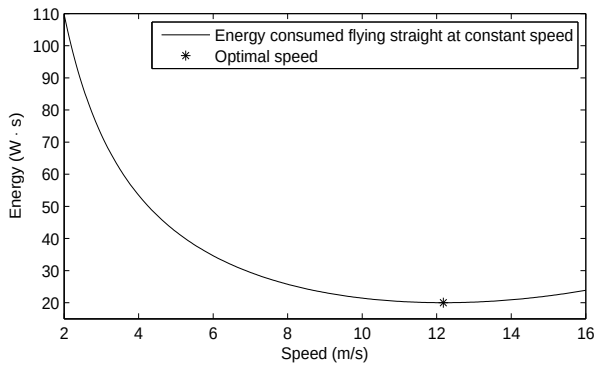

Fig. 6. Energy consumed by the UAV flying straight at constant speed v for a distance d.

a height displacement Δh can be computed as

$$E_{climb} = \int_{h_1/\hat{v}_{climb}}^{h_2/\hat{v}_{climb}} P_{climb}\, dt = P_{climb} \frac{\Delta h}{v_{climb}} \tag{8}$$

$$E_{desc} = \int_{h_2/\hat{v}_{desc}}^{h_1/\hat{v}_{desc}} P_{desc}\, dt = P_{desc} \frac{\Delta h}{v_{desc}} \tag{9}$$

$$\tag{10}$$

Finally, the energy consumed during hovering in an interval $[t_1, t_2]$ can be computed as

$$E_{hover} = \int_{t_1}^{t_2} P_{hover}\, dt = P_{hover}(t_2 - t_1). \tag{11}$$

A third experiment was carried out to measure the time and the power needed during rotations. Since both the angular rotation speed $\omega_{turn}(2.1 rad/s)$ and the power $P_{turn}(225 W/s)$ consumed during rotations can be considered constant, the energy required to cover an angle $\Delta\theta$ can be computed as

$$E_{turn} = P_{turn} \frac{\Delta\theta}{\omega_{turn}}. \tag{12}$$

IV. Finding the optimal speed

The curve reported in Figure 5 shows the power consumption as a function of the linear speed of the UAV. The objective of this section is to compute the speed value that minimizes the energy required to cover a given straight path of length d. The solution to this problem is first shown in a simple case of constant speed, and then extended to the more general case of variable speed, taking into account the acceleration profile recorded during the experiments.

A. Constant speed

The energy consumed by the UAV during a straight flight at constant speed v to cover a distance d is expressed by Equation (7). Figure 6 shows how this energy changes with the speed.

As evident from Figure 6 the curve has a minimum for a speed $v^* \simeq 12$ m/s, which can be computed as

$$v^* = \min_v P(v) \frac{d}{v}. \tag{13}$$

B. Variable speed

To solve the energy minimization problem in the presence of accelerations, we consider trajectories consisting of three phases: an acceleration phase, an intermediate phase in which the UAV has a constant speed, and a final deceleration phase. The total space covered after these three phases is must be equal to a give distance d. Hence, the optimal speed v^* can be found by minimizing the following function:

$$E(d) = \int_0^{t_1} P_{acc}(t)\, dt + \int_{t_1}^{t_2} P(v)\, dt + \int_{t_2}^{t_3} P_{dec}(t)\, dt \tag{14}$$

where P_{acc}, $P(v)$ and P_{dec} are the functions derived from the experiments reported in Section III-A by interpolating data with fifth order polynomials, and

$$t_1 : v_{acc}(t_1) = v \tag{15}$$

$$t_2 = \frac{d - d_{acc} - d_{dec}}{v} = \frac{d - \int v_{acc}(t_1)\, dt - \int v_{dec}(t_1)\, dt}{v} \tag{16}$$

$$t_3 : v_{dec}(t_3) = 0 \tag{17}$$

Figure 7 shows the energy consumed as a function of the maximum speed for different given distances d whereas Figure 8 reports the optimal speed as a function of the covered distance d. Note that for long distances ($d > 1$ Km) v^* tends to the optimal value computed assuming constant speed. This result allows us, to compute the optimal speed that minimizes the energy consumption for each straight line of length d during the path.

V. Path planning

The search area considered in this work is defined by an ordered set of v vertices. Each vertex v_i has coordinates $v_i = (v_x, v_y)$. The corresponding inner angle on each vertex is called γ_i and the bounding lines between each vertex have length $l_{i,j} = ||v_i - v_j||$, as illustrated in Figure 9. The area can be either convex or concave. An area is convex if $\forall i, \gamma_i < \pi$, concave otherwise. In this work no complex shapes are considered, that is, the bounding lines do not intersect each others.

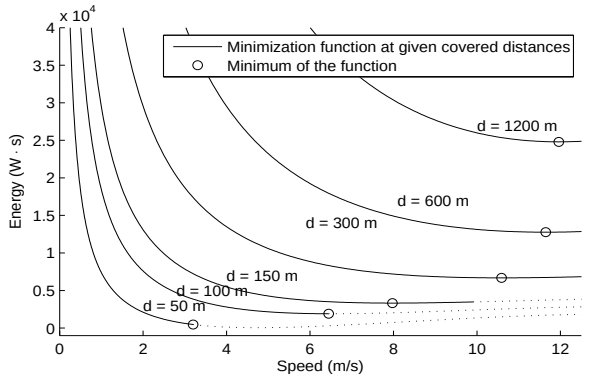

Fig. 7. Energy consumed as a function of the maximum speed for different given distances d.

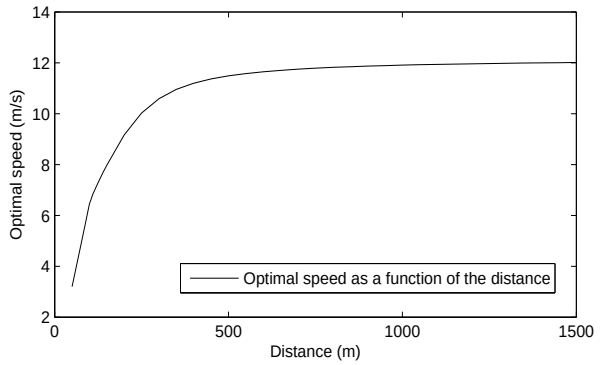

Fig. 8. Optimal speed as a function of the covered distance d.

A. Optimized Back-and-forth Algorithm for convex and concave areas

The back-and-forth algorithm creates a set of waypoints that scan the entire area back and forth along one direction. The number of turns during the survey impacts the time needed to accomplish the entire path since the robot needs to decelerate, turn and accelerate. In the literature, a lot of work has been done to find the optimal scan direction that minimizes the number of turns [6], [12]. Figure 10 shows an example that highlights that changing the scan direction produces a reduction in the number of turns.

If the area has a high number of concavities, finding the optimal direction can be difficult. However, in UAV applications the shape of the area is not a real concern because of

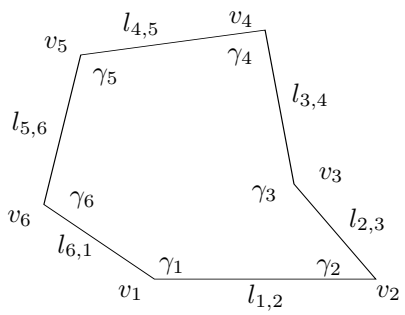

Fig. 9. An example of survey area considered in this paper.

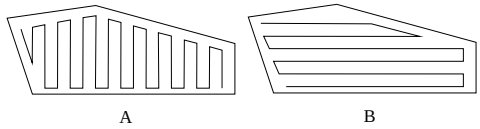

Fig. 10. The number of turns in path A is higher than in path B.

two reasons: a) the survey area is often a quadrilater and even if the area is convex or concave the number of vertex is low, and b) we can assume that there are no obstacles on the sky (after a certain height). For this reason we propose a simple but effective algorithm that optimizes the path by reducing the number of turns by setting the scan direction parallel to the longest bounding line. Is not proved that this choice will lead to the minimum number of turns, but in practice it will be close to it. Another improvement can be done if we consider that normally the final and the starting points can be far from each others and the distance to come back is wasted. To address this issue, the proposed algorithm exploits the return trajectory as a final part of the planned path.

The proposed path planning algorithm decomposes the entire path into three main parts:

1) A list of waypoints that starting from the first vertex go to the longest bounding line through the borders.
2) A list of waypoints that cover the area with the back-and-forth pattern along a scan direction parallel to the longest bounding line.
3) A list of waypoints that go from the final point to the starting point through the borders, if they are not coincident.

Note that, in order to minimize the total energy consumption, for every straight distance the speed is set at the optimal value.

Algorithm Part 1: Given an area $A = \{v_1, \ldots, v_p\}$ and a starting point coincident with one of its vertices, e.g. start_vertex_index $= 1$,

1) Find the longest bounding line $l_{max} = l_{k,k+1}$ being $k, k+1$ the indexes of the vertices determining l_{max};
2) set the scan direction to be parallel to l_{max} by finding the slope of the longest line;
3) set $i =$start_vertex_index;
4) while ($i \neq k$)
   a) compute the distance $d = ||v_i - v_{i+1}||$ from the current vertex to the next vertex plus an additional distance to cover the piece of region created if the $\gamma_{i+1} > \pi/2$. In this case the distance is computed as $d = ||v_i - v_{i+1}|| + \cot(\pi - \gamma_{i+1}) \cdot L_x$ as shown in Figure 11;
   b) place the first waypoint at a distance $(L_y/2, L_x/2)$ from the border and the $m-1$ waypoints at a distance $\Delta_y = d_y$ from each other, along the current scanning direction;
   c) increment i.

The number of waypoints is computed as $m = \lceil d/\Delta_y \rceil$ but since m is rounded up to the next nearest integer, the projected area of the mth waypoint will also include an area that is out of interest (Figure 11) and will produce an additional path to

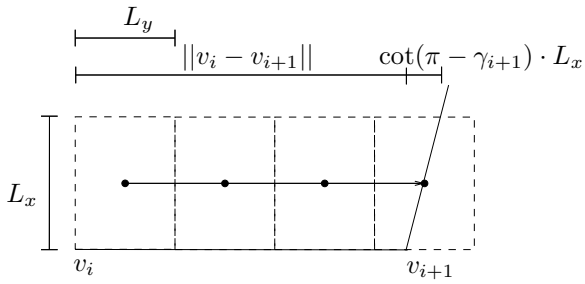

Fig. 11. If the inner angle $\gamma_{i+1} > pi/2$ the distance d also includes an additional part. The number of waypoints can be then computed as $m = \lceil d/\Delta_y \rceil$. However, the projected area created by the last waypoints also covers an area out of interest.

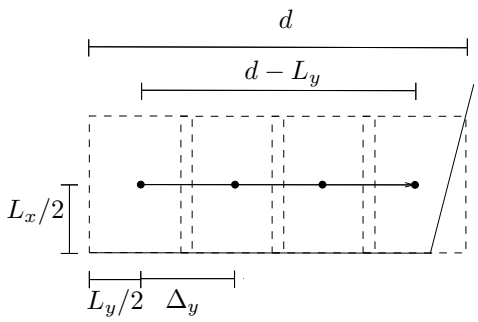

Fig. 12. The first waypoint is placed at a distance $(L_y/2, L_x/2)$. The other waypoints are equally distant each other to perfectly fit d.

reach that waypoint. To overcome this problem, in Part 1.4b is possible to compute $\Delta_y = (d - L_y)/(m - 1)$. This will avoid the need to go any further to cover a region out of interest (Figure 12) and at the same time it will lead to an increase of the overlap.

Algorithm Part 2: In this phase, the back-and-forth pattern takes into account that the area will be reduced by the portion scanned in Part 1 and considers an area for the returning path that will be computed in Part 3.

1) To compute the number of stripes n_s, we need to calculate the maximum distance d_{max} between the longest side $l_{k,k+1}$ and the farthest vertex of the area. Then, the number of stripes is computed as $n_s = \lceil d_{max}/d_x \rceil$;
2) $if\,(n_s \mod 2 \neq 0)$, n_s is incremented by one to make the value even. Note that incrementing the number of stripes by one does not increase the total distance, because otherwise there would be a useless path from the final point to the starting point (Figure 13). Another benefit is that the horizontal overlap also slightly increased since n_s is incremented;
3) the distance between two stripes Δ_x needs to be computed taking into account that the first and the last strip do not need an overlap on the border, thus $\Delta_x = (d_{max} - L_x)(n_s - 1)$ as showed in Figure 13.B.
4) for each strip compute the length d of the strip, and create m waypoints as in Part 1 of the algorithm.

Algorithm Part 3: At the end of phase 2 the last waypoint created will be close to the farthest vertex used to compute d_{max}. If this vertex is not coincident with the starting point we need to come back to the starting point along the borders:

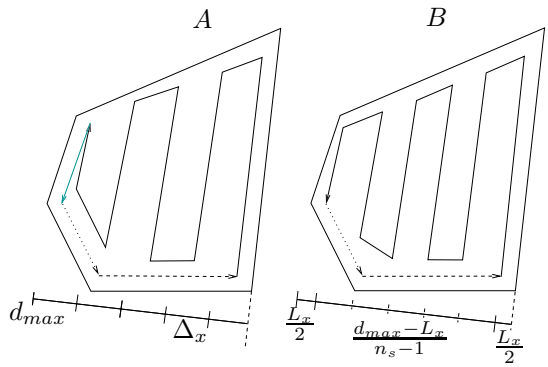

Fig. 13. The number of stripes is computed as $n_s = \lceil d_{max}/d_x \rceil$. Figure A shows that, when n_s is odd, the final strip may be to the opposite direction with respect to the starting point. Figure B shows that if n_s is incremented to an even number the final strip direction will point at the starting point.

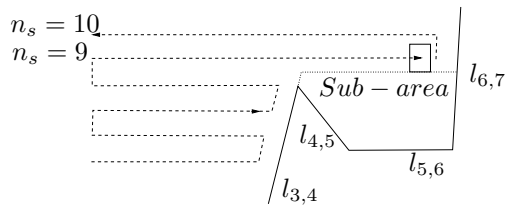

Fig. 14. During the back-and-forth scanning, if it is present a concave sub-area, the algorithm, before continuing the scan, runs recursively the whole procedure on the sub-area.

1) Set $i =$ farthest_vertex_index ;
2) while ($i \neq$start_vertex_index)
   a) place the first waypoint at a distance $(L_y/2, L_x/2)$ from the border and the $m-1$ waypoints at distance Δ_y from each other computed as in part 1, along the current scanning direction;
   b) increment i.

Handling concavities: In order to take into account all the concavities, the algorithm, during Part 2, checks the sides involved during the current scan. If the next strip will involve a side that is not contiguous, it performs the scan but before going to the next strip it recursively runs the whole algorithm on the region just created. Figure 14 clarifies the procedure by an example: when the next strip is $n_s = 9$, the involved side will be $l_{6,7}$ that is not contiguous to $l_{3,4}$. The algorithm will scan strip $n_s = 9$ but before going to the next strip it will run the algorithm to the sub-area just created.

VI. Energy-aware coverage path planning

Given a desired spacial resolution R_d, the maximum altitude h_{max} is computed by Equation 4. Then, with the algorithm described in Section V, we can compute the waypoints of a specific area. A feasibility test can be used to check whether the computed path is suitable for a UAV with specific characteristics and energy. In fact, the total energy consumption of the entire path can be computed as follows:

$$E_{tot} = E_{climb}(0, h_{max}) + E_{desc}(h_{max}, 0) + n_t E_{turn} + \sum_i (E_{acc}(0, v_i^*) + E_v(d_i, v_i^*) + E_{dec}(d_i, v_i^*)) \quad (18)$$

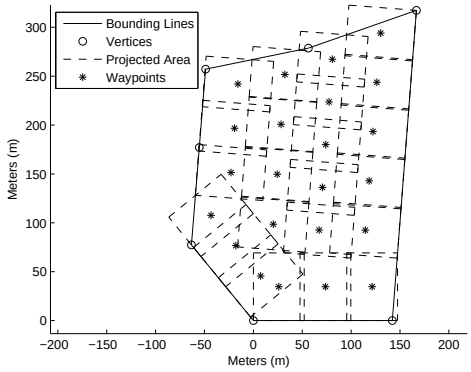

Fig. 15. Coverage path planning computed with the algorithm

where d_i is the distance between two waypoints that compose a straight line denoted with the index i, v_i^* is the optimal speed to travel d_i, and n_t is the total number of turns in the path. Given the amount of available energy E_a, the feasibility test can simply be done by checking if $E_{tot} < E_a$. If the feasibility test is passed, the remaining energy ($E_a - E_{tot}$) can be used to increase the spatial resolution of the acquired images. This can be done by iteratively reducing the flight altitude, recomputing the path, and re-running the feasibility test, until an altitude h is found such that $E_{tot} = E_a - \epsilon$, where ϵ is a given tolerance. If the feasibility test is not passed, the path has to be redesigned considering multiple flights or multiple UAVs. Note that energy can be traded with the spatial resolution, and the same iterative procedure can be applied to reduce the total required energy by increasing the altitude. However, finding the optimal altitude that minimizes E_{tot} is highly complex due to the non linearity of the problem: increasing/decreasing the height will change the energy consumption not monotonically.

A. Experimental Validation

To validate the energy model, we created a set of waypoints over a random area at a height of $h = 20$ (Figure 15). By actually flying the area with the IRIS quadcopter, the flight time to cover the area was $t \simeq 200$ s and the measured energy was $E_{tot} = 4.53 \cdot 10^4$ J. The estimated energy, computed according to our energy model was $\hat{E}_{tot} = 4.62 \cdot 10^4$ J, and the estimated flight time was $\hat{t} = 210.09$ s. These results show that the estimated values were close to the actual values, with a relative error on the flight time estimate $e_t = (\hat{t} - t)/t = 0.05$ and on the total energy $e_E = (\hat{E}_{tot} - E_{tot})/E_{tot} = 0.02$.

VII. Conclusions

This paper presented an energy-aware path planning algorithm that computes a path for achieving the full coverage of a given survey area, taking into account other constraints, namely the available energy, the minimum spacial resolution for the pictures, and the maximum camera sampling period. The paper described a method for deriving an energy model of a specific UAV starting from real measurements. Using such a model, the proposed algorithm computes the speed that minimizes energy consumption along a given distance. Once the full path is generated, it is possible to derive the speed that minimizes the energy consumption for each segment in the path. Then, a feasibility test is performed to verify whether the energy available on the UAV is sufficient to scan the entire area.

Note that, since energy is minimized along individually path segments, the resulting solution is suboptimal. However, the proposed approach has the advantage of combing together different mission constraints, such as area coverage, energy consumption, and image resolution.

References

[1] G. Grenzdörffer, A. Engel, and B. Teichert, "The photogrammetric potential of low-cost uavs in forestry and agriculture," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 31, no. B3, pp. 1207–1214, 2008.

[2] P. J. Zarco-Tejada, J. A. Berni, L. Suárez, and E. Fereres, "A new era in remote sensing of crops with unmanned robots," *SPIE Newsroom*, pp. 2–4, 2008.

[3] W. Kazmi, M. Bisgaard, F. Garcia-Ruiz, K. D. Hansen, and A. la Cour-Harbo, "Adaptive surveying and early treatment of crops with a team of autonomous vehicles," in *European Conference on Mobile Robots*, 2011, pp. 253–258.

[4] D. W. Casbeer, D. B. Kingston, R. W. Beard, and T. W. McLain, "Cooperative forest fire surveillance using a team of small unmanned air vehicles," *International Journal of Systems Science*, vol. 37, no. 6, pp. 351–360, 2006.

[5] J. Nikolic, M. Burri, J. Rehder, S. Leutenegger, C. Huerzeler, and R. Siegwart, "A uav system for inspection of industrial facilities," in *Aerospace Conference, 2013 IEEE*. IEEE, 2013, pp. 1–8.

[6] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

[7] I. Maza and A. Ollero, "Multiple uav cooperative searching operation using polygon area decomposition and efficient coverage algorithms," in *Distributed Autonomous Robotic Systems 6*. Springer, 2007, pp. 221–230.

[8] G. Öst, "Search path generation with uav applications using approximate convex decomposition," Master's thesis, Linköpings universitet, Sweden, 2012.

[9] A. Barrientos, J. Colorado, J. d. Cerro, A. Martinez, C. Rossi, D. Sanz, and J. Valente, "Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots," *Journal of Field Robotics*, vol. 28, no. 5, pp. 667–689, 2011.

[10] E. Santamaria, F. Segor, I. Tchouchenkov, and R. Schoenbein, "Rapid aerial mapping with multiple heterogeneous unmanned vehicles," *International Journal On Advances in Systems and Measurements*, vol. 6, no. 3 and 4, pp. 384–393, 2013.

[11] H. Bast and S. Hert, "The area partitioning problem," in *12th Canadian Conference on Computational Geometry*, 2000.

[12] W. H. Huang, "Optimal line-sweep-based decompositions for coverage algorithms," in *International Conference on Robotics and Automation (ICRA)*, vol. 1. IEEE, 2001, pp. 27–32.

[13] N. Lawrance and S. Sukkarieh, "Wind energy based path planning for a small gliding unmanned aerial vehicle," in *AIAA Guidance, Navigation and Controls Conference, American Institute of Aeronautics and Astronautics*, 2009.

[14] W. H. Al-Sabban, L. F. Gonzalez, and R. N. Smith, "Wind-energy based path planning for unmanned aerial vehicles using markov decision processes," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2013, pp. 784–789.

[15] J. F. Roberts, J.-C. Zufferey, and D. Floreano, "Energy management for indoor hovering robots," in *International conference on Intelligent Robots and Systems (IROS)*. IEEE, 2008, pp. 1242–1247.

[16] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. G. Lee, "Deployment of mobile robots with energy and timing constraints," *Robotics, IEEE Transactions on*, vol. 22, no. 3, pp. 507–522, 2006.