2019 International Conference on Unmanned Aircraft Systems (ICUAS)
Atlanta, GA, USA, June 11-14, 2019

Grid-based Coverage Path Planning with Minimum Energy over Irregular-shaped Areas with UAVs

Tauã M. Cabreira and Paulo R. Ferreira Jr.
Programa de Pós-Graduação em Computação (PPGC)
Universidade Federal de Pelotas (UFPel)
Pelotas/RS, Brasil
Email: {tmcabreira, paulo}@inf.ufpel.edu.br

Carmelo Di Franco and Giorgio C. Buttazzo
Real-Time Systems Lab (ReTiS Lab)
Scuola Superiore Sant'Anna
Pisa, Italy
Email: {c.difranco, g.buttazzo}@santannapisa.it

***Abstract*—Grid-based methods have been proposed to solve the Coverage Path Planning problem using Unmanned Aerial Vehicles in irregular-shaped areas since simple geometric flight patterns, such as the back-and-forth, are inefficient in this type of scenario. However, the grid-based methods usually apply simplistic cost functions and demand high computational time leading to inefficient and expensive paths, making them not usable in real-world scenarios. This paper introduces an energy-aware grid-based approach aimed at minimizing energy consumption during mapping missions over irregular-shaped areas. Our work was built upon a previously proposed grid-based approach. Here we introduce an energy-aware cost function based on an accurate energy model. The proposed approach was able to save up to 17% of energy in real flight experiments, proving that the original cost function was not capable of finding the optimal solution in terms of real energy measurements. Additional simulation experiments were also performed to state the energy savings in different irregular-shaped scenarios. As a further contribution, we also applied two pruning techniques to the original approach dropping the computation time up to 99%.**

1. Introduction

Unmanned Aerial Vehicles (UAVs) consist of aircrafts with no pilots onboard, usually remotely controlled by a pilot on the ground, by a program with a flight plan, or by intelligent systems. These vehicles have increasingly been employed in several application domains, such as surveillance [1], smart farming [2], wildfire tracking [3], cloud monitoring [4], and power line inspection [5].

Many UAVs applications are related to the Coverage Path Planning (CPP) problem, which consists of determining a path that guarantees that an agent will pass over a given zone usually to fully reconstruct the area by photogrammetry [6]. Nowadays most UAVs engage in missions based on CPP using simple geometric flight patterns [7]. The main one employed in real-world scenarios is the back-and-forth, also referred in literature as the lawnmower pattern, adopted by the most popular flight-control software [8] to allow flights based on an offline programmed plan. Recent efforts present

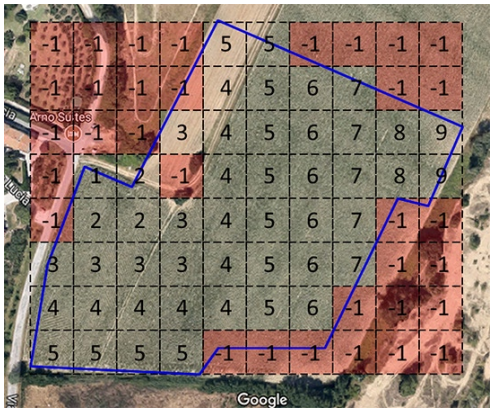

Figure 1. Irregular-shaped area discretized into a regular grid with the starting position marked with number 1 and the surrounding neighbors with number 2, and so on. Obstacles and no-fly zones are marked with -1.

novel energy-aware CPP solutions considering the back-and-forth [9] and the spiral [10] patterns for regular-shaped scenarios. These approaches employ an energy model [9] based on real measurements to optimize the speed in order to minimize the energy during the flights.

However, these flight patterns may be inefficient depending on the complexity of the shape of the area. In order to deal with these areas in CPP missions, Valente et al. [11] proposed a more complex algorithm using a Deep-limited search with a backtracking procedure over a scenario discretized into a regular grid through approximate cellular decomposition [6], as shown in Fig. 1. The irregular-shaped area consists of a concave/convex polygon, which may contain obstacles and no-fly zones inside or outside. These regions are marked with a -1. The decomposed area is converted to a regular graph numerically labeled by the Wavefront algorithm, which is a flooding algorithm that marks the neighborhood adjacency of cells. The starting cell is marked with the number 1 and all of its neighbor cells are marked with the number 2, and so on. The farther the cell is from the initial position, the greater it is value. Using this approach, the search length can be limited to the number of vertices, and consequently, the search neither goes around

in infinite cycles nor visits a node twice [11].

Despite being the state-of-the-art in CPP missions over irregular-shaped areas, the algorithm proposed by Valente et al. [11] presents high computational time during the offline search phase due to the complexity of the area. The approach may need several hours to find a path on a scenario with a reasonable number of cells, e.g. 50. This is impractical for real-world applications considering the amount of time spent for a single run. Besides, the quality of the solution is based on an unreliable cost function that relies exclusively on the number of turns, which may provide more expansive paths in terms of real measured energy, while discarding promising solutions during the minimization process. In fact, several authors consider the minimization of the turning maneuvers as the main performance metric for CPP problems aiming at indirectly minimizing energy consumption [12, 13, 14, 15, 16]. We intend to show that it is possible to use an energy estimation to generate more economic paths in terms of energy instead of minimizing the turns.

This paper proposes an energy-aware grid-based coverage path planning algorithm aimed at minimizing the energy consumption during mapping missions over irregular-shaped areas using UAVs, specifically multi-rotors. This work improves the state-of-the-art grid-based approach proposed by Valente et al. [11] by replacing its original cost function based on the sum of the angles to a novel energy-cost function. The function computes the cost of the path by considering the dynamics of the vehicle and exploits the energy model described by Di Franco and Buttazzo [9] and improved by Cabreira et al. [10] to account not only for the energy required for every turn but also for the energy needed when accelerating/decelerating and flying at a constant speed. Real flights have been performed to validate the proposed approach and verify the accuracy of the energy model. In particular, the energy-cost function allowed saving 17% of the energy with respect to paths generated by the previous cost function. Additional simulation experiments were also performed in different irregular-shaped scenarios varying size and complexity, stating once more the effectiveness of the proposed approach over the original one.

As a further contribution of this work, we also apply two pruning techniques to the original algorithm drastically reducing the computation time up to 99%. This improvement allows us to generate complete coverage paths for all the different starting positions in the workspace. Once the paths are computed offline in a reasonable amount of time (in order of seconds), we are able to indicate the ideal location to start the mission, saving even more energy and making this approach feasible for real-world applications.

The remain of the paper is organized as follows: Section 2 presents the related work. Section 3 describes the proposed approach using the energy-aware cost function and the two pruning techniques applied to reduce the computational time of the original algorithm. Section 4 reports a set of experimental results carried out to validate the proposed approach. Section 5 draws the conclusions and presents some future work.

2. Related Work

The problem of Coverage Path Planning (CPP) using UAVs has been addressed by several authors in the literature. A survey on CPP was recently published by Cabreira et al. [17], where the authors exclusively review approaches related to UAVs. The authors consider the classic taxonomy defined by Choset [6] to classify the existing approaches according to the cellular decomposition technique adopted.

Some approaches explore only rectangular areas as presented by Andersen [7], where the author compares five types of flight patterns from the *US National Search and Rescue Manual*, including back-and-forth (parallel and creeping line), square, sector search, and barrier patrol. More complex areas containing obstacle-regions with arbitrary shapes are explored by Xu et al. [18]. The authors decompose the area into a simple set of cells using Boustrophedon Cellular Decomposition (BCD) [6] and build an adjacency graph with the vertices representing the cells and the edges connecting the adjacent cells. Each cell is explored using a back-and-forth pattern and the order between the cells follows a Eulerian circuit with start and end at the same vertex. This coverage algorithm avoids to fly over previously explored regions, but it is applied only for fixed-wing UAVs.

Concave polygonal areas are decomposed into convex subregions by Li et al. [13] using a minimum width sum algorithm based on a greedy recursive method [19]. The algorithm finds the optimal line sweep direction and performs back-and-forth motions perpendicular to this line aiming at minimizing the number of turning maneuvers. Different sweep directions for each convex subregion are employed in order to optimize the complete coverage path. Another recent CPP algorithm for convex and non-convex areas is proposed by Torres et al. [16]. A back-and-forth pattern perpendicular to the optimal line sweep direction performs the coverage in convex polygons. However, in concave shapes it is necessary to verify if there is an interruption in the stripes of the pattern. If so, the area is decomposed into concave and convex subregions. The authors also explore four back-and-forth alternatives, varying the direction and the orientation, aiming at minimizing the distance between the subregions.

A spiral CPP algorithm for missions in coastal regions using multiple UAVs is explored by Balampanis et al. in several works. The authors discretize the workspace using a Constrained Delaunay Triangulation (CDT) [20, 21], stating that grid decomposition creates regular square cells that are partially over no-fly zones or outside the workspace. Thus, the CDT provides triangle cells within the area of interest matching almost the exact shape of the area. In order to generate more uniform triangles, they applied the Lloyd optimization. Then, a spiral algorithm previously proposed by Balampanis et al. [22], and improved by Balampanis et al. [23] by introducing a smoothing parameter, is used to generate the coverage paths in the resulting sub-areas.

Most of the approaches in the literature seek the minimization of the number of turning maneuvers to indirectly reduce the mission execution time and the energy

consumption. Recent studies present energy-aware solutions exploring the dynamics and the behavior of the UAVs to save energy. Di Franco and Buttazzo [9] propose an energy-aware back-and-forth algorithm for photogrammetry with energy and resolution constraints imposed by the mission. The algorithm minimizes the energy spent during the mission not only minimizing the number of turning maneuvers, but also flying at the maximum altitude and using optimal speeds in each straight segment of the path. An energy-aware spiral algorithm is proposed by Cabreira et al. [10]. The algorithm performs turning maneuvers with wider angles and does not need to reduce the speed to zero on every turn, which decreases the acceleration and deceleration periods. This behavior keeps the optimal speed for longer periods, providing an even more effective energy saving than the one proposed by Di Franco and Buttazzo [9]. Both approaches adopt an energy model derived from real measurements.

Despite the energy outcomes in regular-shaped areas, back-and-forth and spiral flight patterns still generate inefficient trajectories considering complex areas with irregular shape and no-fly zones. Thus, some grid-based solutions have been proposed both for single [11] and multiple-UAVs [2, 24]. In particular, Valente et al. [11] proposed an algorithm for image mosaicing in precision agriculture with irregular-shaped fields. The area of interest is discretized into a regular grid using the approximate cellular decomposition and converted into a regular graph. A Deep-limited search is used to build a tree with all possible coverage paths in order to find a complete coverage path that passes through all nodes in the adjacency graph only once. A backtracking procedure is also employed to solve issues, such as the choice among neighbors with the same potential weight. Another similar approach is discussed by Nam et al. [25], where the authors use the same Wavefront algorithm to obtain the coverage path, but also apply a cubic interpolation algorithm for smoothing the turning maneuvers.

The algorithm proposed by Valente et al. [11] can be considered the state-of-the-art for coverage missions over irregular-shaped areas. However, it has an exponential complexity and uses a simplistic cost function aiming at minimizing the number of turning maneuvers to save energy. Important elements such as acceleration, deceleration, and optimal speed that impact the energy cost are missing and will be considered in our proposed energy-aware grid-based approach.

3. Proposed Approach

Our proposed approach exploits the energy model described by Di Franco and Buttazzo [9] and improved by Cabreira et al. [10] to reformulate the cost function adopted by Valente et al. [11]. Furthermore, our algorithm includes two pruning techniques drastically reducing the computational time (in the order of seconds) and saving even more energy in a real-world coverage path. It is important to notice that the planning phase is executed offline and the resulting coverage path generated by the algorithm is loaded into the UAV in a waypoint list format.

3.1. Energy-aware Cost Function

The algorithm proposed by Valente et al. [11] presents a cost function based on the sum of angles to find the minimum-cost path to perform a complete coverage, as shown in Eq. (1):

$$\Gamma = \sum_{i=1}^{m} \gamma_k^{\{i\}}, k \in \{135°, 90°, 45°, 0°\} \tag{1}$$

where m represents the number of waypoints of the path $\{i_1, i_2, ..., i_m\}$ and γ represents the angle of the i-th waypoint compressed by the k set. In literature, the path is usually evaluated using the number of turns. The authors often correlate the turns with the power consumption trying to minimize them to save energy. However, such cost functions are simplistic and do not explore important details as acceleration and deceleration phases during the turns.

The energy model proposed by Di Franco and Buttazzo [9] splits the path into a set of straight segments and rotations to predict the energy cost of a path. Following the same idea, the cost function has been extended in order to consider the traveled distance between waypoints. Thus, it is possible to evaluate the path and estimate the total energy (and time) using the Eq. (**??**) as follows:

$$\begin{aligned}\Gamma_E = &\sum_{i=1}^{m} \Big(\int_0^{v_i} P_{acc}\,\mathrm{d}v + P_{v_i}\Delta T_i + \int_{v_i}^{0} P_{dec}\,\mathrm{d}v \Big) \\ &+ \sum_{i=1}^{m} E_{turn}(\gamma^{\{i\}})\end{aligned} \tag{2}$$

where the first summation computes the energy consumed during a set of straight line i by splitting it into three phases (acceleration, deceleration, and constant speed) and the second summation considers all the rotations of the path. P_{acc}, P_{dec}, and P_v define the power consumed when accelerating, decelerating, and flying at constant speed. $E_{turn}(\gamma^{\{i\}})$ is the energy to rotate an angle γ at the i-th waypoint (computed as the power consumed when turning P_{turn} multiplied by the duration of the rotation). ΔT_i is the time when flying the portion of path at constant speed and it is computed considering the total distance and constant acceleration and deceleration. The terms in Eq. (2) are polynomial functions obtained through real measurements and allow reaching high accuracy in the energy prediction of a given trajectory [9].

3.2. Algorithm Optimization

The proposed Energy-aware Grid-based CPP approach (EG-CPP) is presented in Algorithm 1. The green lines represent the additions made during the optimization process and the red lines represent the modifications in the original algorithm.

Algorithm 1 Energy-aware Grid-based CPP Algorithm

```
1:  grid ← convertAreaToGrid(area)
2:  matrix ← floodingMatrix(grid)
3:  firstPoint ← getStartingPosition()
4:  prevCost ← 0
5:  path ← recursiveFunc(firstPoint, matrix, prevCost)
6:  loop
7:     cell ← getLastCell(path)
8:     tempMatrix ← computeTempMatrix(matrix, cell)
9:     neighbors ← computeNeighbors(cell, tempMatrix)
10:    if There is no neighbors then
11:       return
12:    end if
13:    for each neighbors(i) do
14:       path ← path + neighbors(i)
15:       cost ← computeCost(path)
16:       cost ← cost + prevCost
17:       if cost > minimumCost then
18:          return
19:       end if
20:       path ← recursiveFunc(path, tempMatrix, cost)
21:       ~~FiFo ← FiFo + path~~
22:       path ← path + firstPoint
23:       cost ← computeCost(path)
24:       if cost < minimumCost then
25:          minimumCost ← cost
26:          minimumPath ← path
27:       end if
28:    end for
29: end loop
30: ~~minimumPath ← min(ComputeCost(FiFo))~~
31: return minimumPath
```

3.2.1. Pruning technique 1. In the approach proposed by Valente et al. [11], the nearest neighbor cell with the highest value is chosen to find a complete coverage path. If two or more cells have the same highest value, all the cells must be explored. Every complete coverage path passes through all nodes in the adjacency graph only once. A Deep-limited search is used to build a tree with all possible coverage paths.

During the recursive search (lines 6-29), the current cell is obtained from the last added neighbor (line 7) and marked as visited in the temporary matrix (line 8). Then, the neighbor cells with the highest values are selected (line 9). If there is no available neighbor, the search in the current path ends and it returns to the previously recursive call (line 10), where the path is added to the FIFO data structure (line 21). Otherwise, the search continues in all selected neighbors (lines 13-28) with a new recursive call for every one of them (line 20).

The approach proposed by Valente et al. [11] computes the entire cost of all possible paths at the end of the algorithm (line 30), including those paths whose cost are much higher than the minimum value. This repetitive operation is massive and can be prevented. Thus, we modified the algorithm to store the minimum-cost and the path associated with it during the search (lines 25-26). Then, it is checked if the current path has a cost that is already higher than the cost of the stored path at each iteration (line 17). By adopting this technique, it is possible to drastically reduce the number of unnecessary recursive calls, pruning a huge number of partial paths (and all the sub-trees starting from that current cell) that already present higher costs than the complete current minimum path.

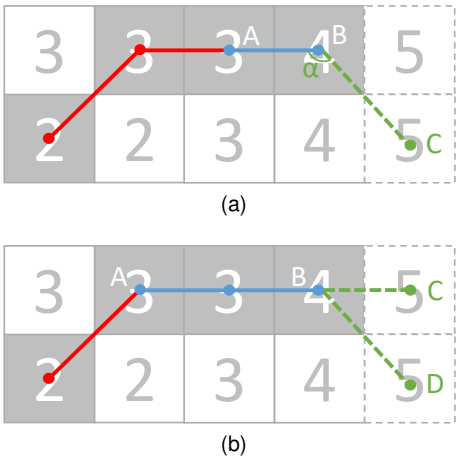

Figure 2. Example of the algorithm adopting the pruning technique 2, where only the cost of a partial path is computed. (a) sum of angles cost function (b) energy-aware cost function

3.2.2. Pruning technique 2. It is not necessary to compute the cost of the path from the initial cell to the current one at each iteration, but only the cost introduced by choosing the next neighbor (line 15).

By applying the pruning technique to the algorithm with the original cost function, as shown in Fig. 2a, it is necessary to consider three cells A, B, and C. In the mentioned example, the cost for travelling the path from the initial cell to A (red path) was already computed. The current cell is B and the next chosen neighbor is the cell C. Thus, the updated cost to reach C is the sum of the previous cost and the new angle $\widehat{ABC}$.

On the other hand, in order to compute the partial cost of the path using the algorithm with the energy-aware cost function, as illustrated by Fig. 2b, it is only necessary to consider the current straight line and the last performed turn. In the example, the current position is the cell B and there are two potential neighbor cells C and D to be explored. By choosing the cell D, the energy of the partial path can be computed by Eq. (3) as follows:

$$E_{partial} = E_{partial} + E_{turn}(\widehat{ABD}) + E_{line}(\overline{BD}) \quad (3)$$

where $E_{partial}$ is the previously computed energy to travel from the starting position to the current point B, $E_{turn}(\widehat{ABD})$ is the energy needed to perform a turn at B, and $E_{line}(\overline{BD})$ corresponds to the energy necessary to travel the straight distance from B to D, i.e., the first term of Eq. (2) considering a single line. However, when choosing

the cell C, the straight distance increases and the energy of the partial path should be computed by Eq. (4) as follows:

$$E_{partial} = E_{partial} - E_{line}(\overline{AB}) + E_{line}(\overline{AC}) \quad (4)$$

where $E_{line}(\overline{AB})$ is the previously computed energy to travel from A to B (blue line) and $E_{line}(\overline{AC})$ is the energy needed to travel from A to C. In this case, since we are going forward to the cell C, it is necessary to remove the energy computed considering the old distance ($\overline{AB}$) and add the energy of the new straight line ($\overline{AC}$).

The two proposed pruning techniques can be applied to the algorithm using the original and the energy-aware cost function. The techniques drastically reduce the computational time of the algorithm during the offline planning phase, making it time-affordable for real-world applications. Measurements on such reduction are reported in Section 4.

4. Experimental Results

This section reports a set of experiments aimed at evaluating the performance improvements of our Energy-aware Grid-based CPP Algorithm (EG-CPP) with respect to the one proposed by Valente et al. [11]. The benefits of the energy-aware cost function are validated by performing real energy measurements carried out during real flights. A set of simulations exploring irregular-shaped areas were also performed to state the energy savings in different scenarios. We can trust in the simulation results once the energy model proved to be quite accurate regarding the real energy measured and the estimated one. Further simulation experiments were also executed to analyze the computational time reduction provided by the algorithm optimization using the pruning techniques.

An IRIS quadrotor with a GoPro camera mounted on a Gimbal stabilizer was used to perform the experiments. The quadrotor weighs about 1.3 Kg, carrying a LIPO 3S battery, and the autopilot is an Arducopter 3.2 on top of a PixHawk board. For each flight, the autopilot saves a log with all the useful information to analyze the experiment (GPS, speed, altitude, voltage and current, etc.). The algorithms have been implemented on MATLAB® both with the original and the energy-aware cost function. All files and logs from the experiments are available at GitHub in https://github.com/tauacabreira/EGCPP.

4.1. Cost Function Analysis

The area of interest of the real flights consists of a concave polygon with an internal no-fly zone. The area was discretized into a grid of 8 x 10 cells according to our on-board camera characteristics. Since we are interested in coverage paths for a photogrammetry sensing application, we consider elements such as flight altitude (10m), camera resolution (2386 x 2386 pixels), field of view (100 degrees), and overlapping rate (10%) to determine the size of the cells in the area. Using the flooding algorithm, we marked 47 valid cells to be considered during the offline search phase of the coverage path. Cells outside of the area or within the no-fly zone are not included.

We run the algorithm with the original (O-F) and the energy-aware (E-F) cost function - see Eq. (1) and Eq. (2) - considering every valid cell of the area as a potential starting position for the path. Note that, this test was possible only thanks to the modified version of the algorithm that includes the two pruning techniques which sped up the execution time up to 99%, otherwise just a single run would take about 3 hours making the following analysis difficult to achieve. Fig. 3a and Fig. 3b present the *colormap* with the minimum-cost paths starting in each one of the 47 cells of the scenario. As it can be seen, there is a wide range of values and the ideal solution depends on the initial point.

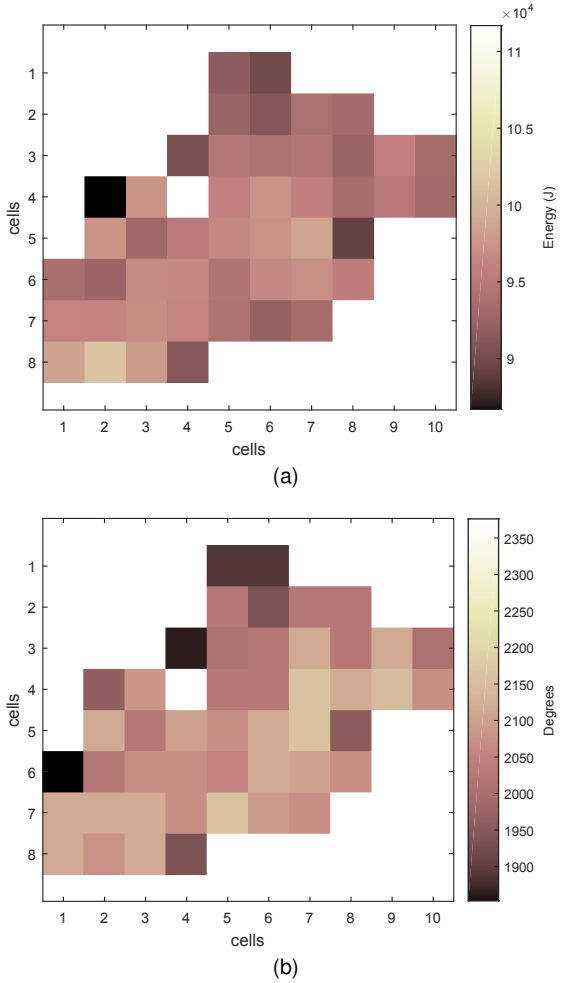

Figure 3. Colormaps with the minimum-cost paths starting in each one of the 47 valid cells of the workspace. (a) E-F with the ideal starting position in cell (4,2), (b) O-F with the ideal starting position in cell (6,1).

The minimum-cost path generated by the E-F (Fig. 3a) starts at the cell (4,2), while the minimum-cost path generated by the O-F (Fig. 3b) starts at the cell (6,1). Both cells are painted in black in the *colormap*. Despite the fact that the two cost functions produce distinct behaviors, it is possible

Figure 4. Four real flights performed with two different starting positions according to the colormap. (a) Path generated with E-F and ideal starting position at Cell (4,2), (b) Path generated with E-F and normal starting position at Cell (6,1), (c) Path generated with O-F and normal starting position at Cell (4,2), (d) Path generated with O-F and ideal starting position at Cell (6,1).

to identify some similarities in the results presented in the two figures. In both *colormaps*, the ideal starting positions are at one of the corners of the irregular-shaped area, not in the center of the scenario. The O-F computes the minimum-cost path based on the sum of the turning angles, while the E-F explores directly the energy consumption estimation. Both generate different coverage paths with different ideal starting positions. In this way, it is necessary to compare them in order to point out the most accurate cost function. Even if cost functions optimize different metrics, we believe that this is a fair comparison considering that the main goal of the O-F is indirectly minimizing the energy.

4.2. Flight Results

Four real flights were performed to evaluate the energy consumption of the minimum-cost paths generated using O-F and E-F. All flights were executed in a short period of time (less than an hour), with a speed of $8m/s$ and in a pleasant weather condition with almost zero wind. Fig. 4a and Fig. 4b show the paths generated with E-F, while Fig. 4c and Fig. 4d present the paths generated with O-F, both starting from the cells (4,2) and (6,1), respectively. The area of interest and the no-fly zone are marked by the red line, the planned path by the blue line, the performed path during the real flights by the thin white line, the starting position by the green "x", and the final position by the red "x".

The experimental results can be seen in Fig. 5. The green bars represent the flight results, while the yellow bars illustrate the estimated values for the energy consumption obtained through the energy model previously proposed. We can observe that our proposed E-F obtains the best results, showing an energy saving of 17% in real flights. The path generated with the E-F and starting from the ideal position at the cell (4,2) overcomes the path generated by the O-F using its ideal initial point at the cell (6,1) - the fourth bar vs. first bar in Fig. 5. Even not considering the best choice for the starting position - E-F with cell (6,1), our solution overcomes the original approach - third bar vs. first and second bars.

Table 1 presents the detailed information about energy and time results comparing original and energy-aware approaches using different starting positions obtained from the

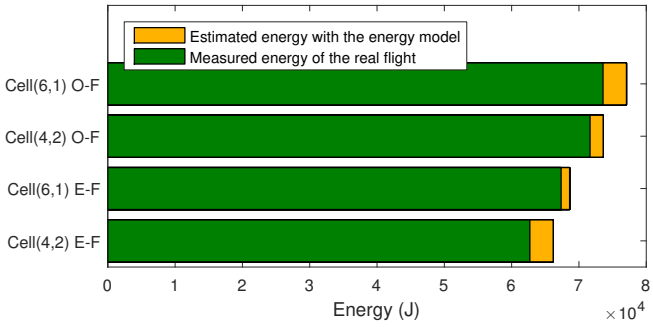

Figure 5. Energy consumption measured during real flights (in green) and predicted in simulation (yellow) with the O-F and E-F considering two starting positions. The optimum found by O-F (cell (6,1)) actually behaves the worst and also consumes more than other discarded paths (cell (4,2)) showing that using the number of turns is not accurate enough. The E-F, instead, not only produces better paths given the same starting position (cell (6,1)) but also finds the best path with a resulting energy saving of 17%.

TABLE 1. ENERGY CONSUMPTION AND MISSION EXECUTION TIME OBTAINED IN ESTIMATION AND REAL FLIGHTS WITH THE ORIGINAL AND THE ENERGY APPROACHES

Path	O-F	E-F	Real Flight	Accuracy
Cell (6,1) O-F	N/A	359.15s	348.60s	97.06%
Cell (4,2) O-F	N/A	342.91s	340.00s	99.15%
Cell (6,1) E-F	N/A	319.97s	317.00s	99.07%
Cell (4,2) E-F	N/A	308.75s	306.00s	99.10%
Cell (6,1) O-F	1890^o	$7.7053 \times 10^4 J$	$7.3583 \times 10^4 J$	95.49%
Cell (4,2) O-F	2205^o	$7.3593 \times 10^4 J$	$7.1655 \times 10^4 J$	97.36%
Cell (6,1) E-F	1980^o	$6.8607 \times 10^4 J$	$6.7354 \times 10^4 J$	98.17%
Cell (4,2) E-F	2025^o	$6.6165 \times 10^4 J$	$6.2710 \times 10^4 J$	94.77%

colormaps. The O-F is not able to estimate the time needed to perform a flight, while our E-F can estimate energy and time. The O-F tries to indirectly save energy by minimizing the total sum of angles. However, while the O-F points out that the path starting from the cell (6,1) presents the minor cost value (1890^o), the E-F states that actually this path spends the greater amount of energy. The real measurements confirm this assumption and validate the proposed approach. The last column of Table 1 presents the accuracy rate of the E-F in estimating energy and time spent during the flights.

The logs store information such as velocity, time, current, and voltage. The last two were used to compute real energy.

The energy-aware cost function (E-F) of the EG-CPP is based on the energy model proposed by Di Franco and Buttazzo [9] and improved by Cabreira et al. [10]. It is able to correctly estimate the energy necessary to perform a mission. The energy consumption accuracy varies from 94% to 98%, approximately. Using the Eq. (2), it is possible to correctly estimate the energy needed to perform a given path splitting it into a set of straight lines composed by three phases (acceleration, flying at the constant speed, and deceleration), and a set of turning maneuvers.

The energy model was built upon the real measurements and considers external forces such as the drag of the vehicle, which explains the estimation precision during the experiments. The results lead to interesting conclusions and possibilities. First, it is worth to notice that the *colormap* illustrated in Fig. 3a is validated by the results obtained from the energy model. Second, it is possible to apply the offline feasibility test proposed by Di Franco and Buttazzo [9] to know in advance if the energy stored in the battery is sufficient to perform a flight. An online feasibility test was also proposed by Di Franco and Buttazzo [9] to constantly check if the remaining energy in the battery is sufficient to bring the UAV back to the starting position. In this way, the proposed solution is able to avoid crashes during flights due to battery-exhaustion.

4.3. Simulation with Different Scenarios

Additional simulation experiments were performed with different irregular-shaped scenarios. We explored farm regions near the city of Pisa in Italy, importing such scenarios to MATLAB® to compare the O-F and the E-F. As stated in Section 4.2 and Section **??**, the E-F overcame the O-F in a real scenario considering the energy spent during the flight. Furthermore, the E-F presents a high accuracy regarding the estimated energy and the real measured one. In this way, we can trust in these additional simulations to state the effectiveness of the proposed approach.

Four irregular-shaped scenarios with different size and complexity were employed in the experiments: (a) Scenario A with 37 cells; (b) Scenario B with 45 cells; (c) Scenario C with 47 cells containing a no-fly zone at the Cell (4,5) and (d) Scenario D with 50 cells containing a no-fly zone at the Cell (5,5). Fig. 6 presents the simulation experiment results considering the flight time and the energy consumption of the paths generated by the O-F and the E-F and Fig. 7 illustrates the four scenarios and the corresponding complete coverage paths generated by the cost functions. Table 2 presents the cost values of O-F and E-F for each scenario and the percentage of energy saving of our proposed approach over the original one.

The paths generated by the original approach are based on the O-F, which is the minimization of the total sum of the angles. For the Scenario A containing 37 cells, the minimum-cost path using O-F is the path with 1764°. On the other hand, the paths generated by our proposed approach are based on the E-F, which computes the energy consumption necessary to perform the path. In this case, the minimum-cost path using E-F in the Scenario A consumes $3.5810 \times 10^4 J$. We can use the O-F to compute the sum of the angles of the path generated by E-F (1871°), as well as we can use the E-F to estimate the energy spent by the path generated by O-F ($3.9114 \times 10^4 J$). In some cases, as in the Scenario A and the Scenario C, the lower value for the sum of the angles does not mean the minimum energy consumption, which states that this is not an appropriate metric for generating energy-aware complete coverage paths for UAVs. According to the results presented in Table 2, the four paths generated by E-F consume less energy than the paths generated by O-F with energy savings varying from 2.5% to 8.45%.

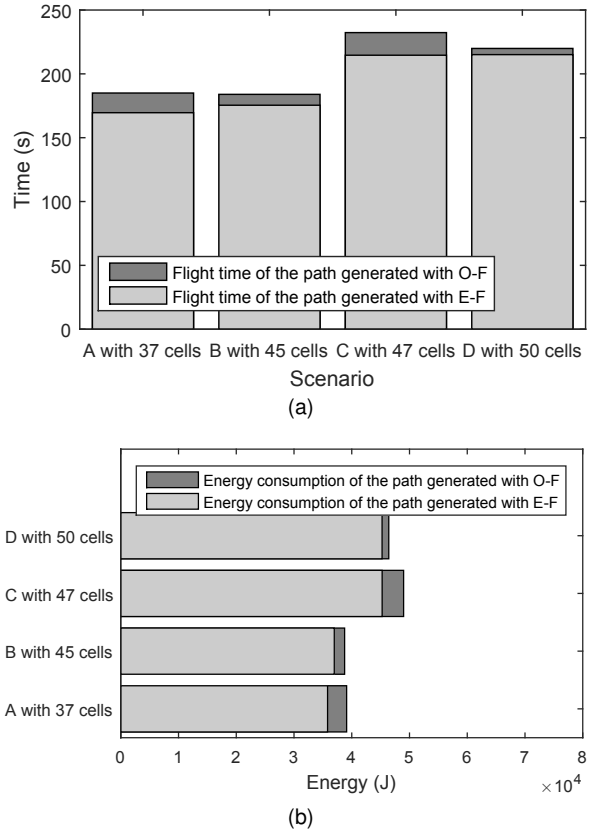

Figure 6. Simulation experiment results with four different scenarios. (a) Flight time of the path generated with O-F (in dark grey) and E-F (in light grey), (b) Energy consumption of the path generated with O-F (in dark grey) and E-F (in light grey)

TABLE 2. ENERGY CONSUMPTION ESTIMATION WITH THE ORIGINAL AND THE ENERGY APPROACHES

Scenario	Sum of Angles		Energy ($\times 10^4 J$)		Energy saving
	O-F	E-F	O-F	E-F	
A (37 cells)	1764°	1871°	3.9114	3.5810	8.45%
B (45 cells)	1617°	1532°	3.8764	3.6972	4.62%
C (47 cells)	2114°	2205°	4.8980	4.5261	7.59%
D (50 cells)	2009°	1926°	4.6411	4.5249	2.50%

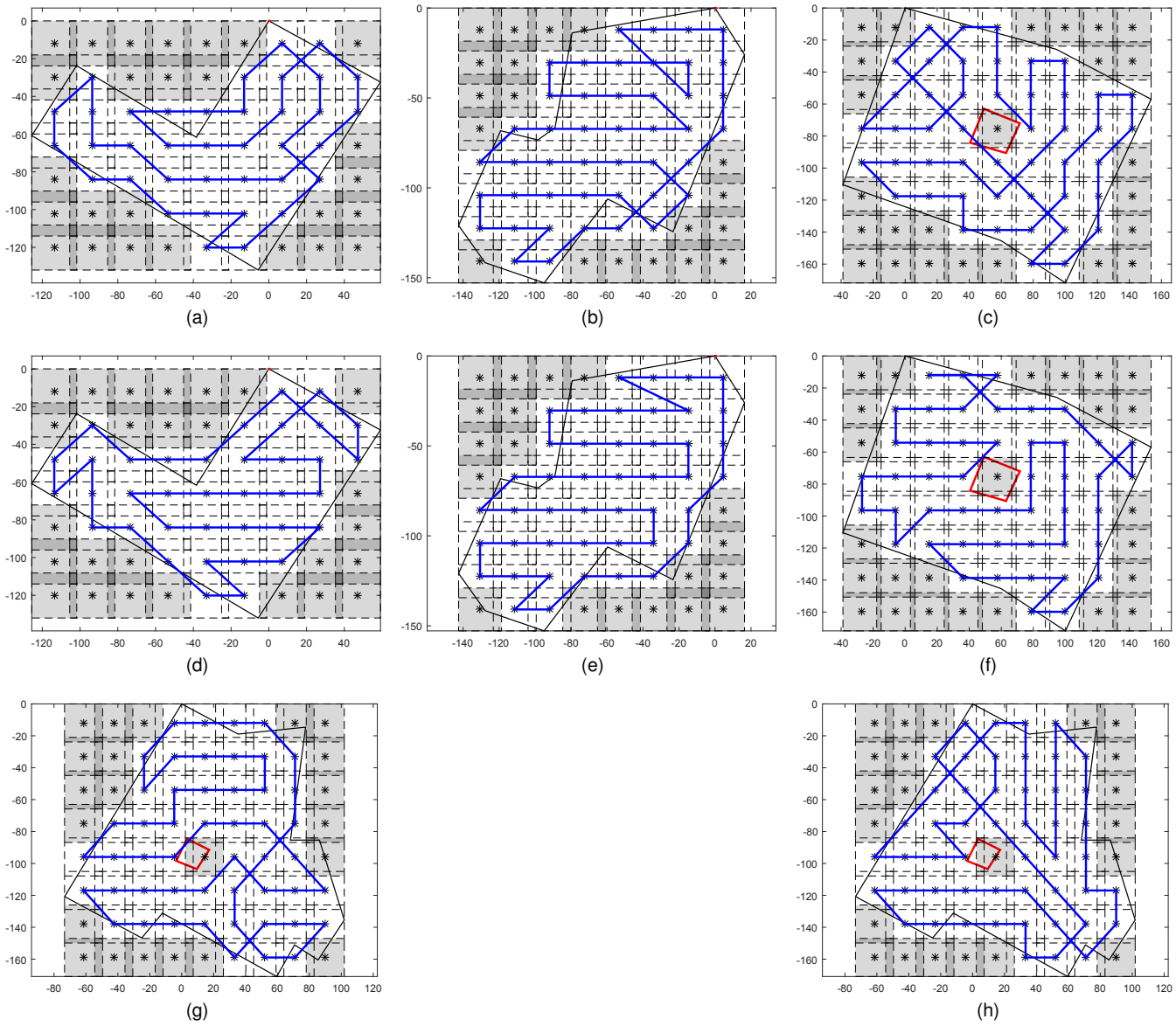

Figure 7. Four different irregular-shaped scenarios based on farm regions near the city of Pisa in Italy. (a-d) Scenario A with 37 cells and paths generated by O-F and E-F, (b-e) Scenario B with 45 cells and paths generated by O-F and E-F, (c-f) Scenario C with 47 cells containing a no-fly zone at the Cell (4,5) and paths generated by O-F and E-F, (g-h) Scenario D with 50 cells containing a no-fly zone at the Cell (5,5) and paths generated by O-F and E-F.

4.4. Computational Time Analysis

The impact of the two pruning techniques in the computational time of the algorithms was evaluated in a series of simulation experiments. The complete coverage paths are generated during an offline planning phase. Thus, we are interested in reducing the computational time in order to speed up the total execution time of the mission.

A generic rectangular area composed by $n_c \times 4$ cells, where n_c is the number of columns of the grid, was adopted to analyze the algorithms. The overall number of cells can be increased by incrementing n_c. We are interested in measuring the computation time for a generic number of cells. Note that, we decided to use a rectangular area without obstacles because complex shaped areas, even with the same number of cells, may lead to a high difference in the computational cost.

For the sake of clarity, the original grid-based approach proposed by Valente et al. [11] is denoted as Alg. A, the grid-based with the original cost function, but modified by the pruning techniques as Alg. B, and the Energy-aware Grid-based CPP (EG-CPP) as Alg. C - with the pruning and the energy cost function. The computational time of the coverage paths may change according to the adopted starting position. For this reason, we performed each experiment for all starting positions. Fig. 8 reports the average of the

computational time of Alg. A, Alg. B, and Alg. C in an area with a number of cells varying from 16 to 48 with a step of 4.

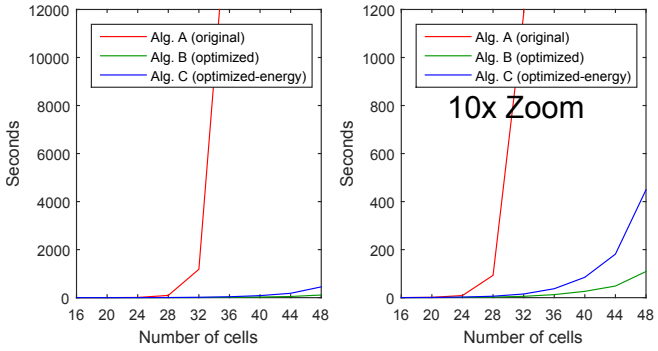

Figure 8. Plot of the computational time from 16 to 48 cells comparing Alg. A (original), Alg. B (optimized), and Alg. C (optimized-energy). The 10x zoom ease to visualize the small difference between the Alg. B and the Alg. C, which is in order of seconds.

Experimental results show that Alg. A has a significant computational time during the offline planning phase. For instance, in a scenario with 36 cells the algorithm takes more than 3 $hours$ to run. However, the optimization proposed in Alg. B drastically reduces the computational time spent by the Alg. A, reaching a percentage of improvement that varies from 44% to 99,9% as the number of cell increases. In fact, results show that the computational time is in the order of seconds instead of hours. Alg. C, that includes the energy-aware cost function, reaches a percentage of optimization over the Alg. A that varies from 31% to 99%. The Alg. C is slightly more expensive than Alg. B but still takes only $35s$ to be executed on 36 cells, stating the effectiveness of the proposed approach. This was expected since the original cost function only considers the sum of angles of the path, while the proposed one considers more complex elements, such as acceleration, deceleration, and constant-speed phases.

Finally, we measured the difference in the total number of executions of the recursive function considering the original and the modified approaches, which explains the improvement in the computation time of the algorithm. Fig. 9 illustrates the histogram of the cost for all possible paths searched with Alg. A and B in a scenario with 24 cells.

The original approach explores the entire sample field presenting a wide range of values with a large number of executions for each one them. Several different paths present the same or similar costs. For example, more than 450 paths have the cost of 1400, which is the sum of angles in the original cost function. The optimized approach computes the cost of the paths during the search phase and discards incomplete high-cost solutions, taking advantage of the pruning technique. In the example shown in Fig. 9, the first path computed with the pruning technique had a cost of 1400. All the successive paths (complete or incomplete) higher than this value were discarded. New paths with a lower value will further reduce the search cost. This procedure avoids

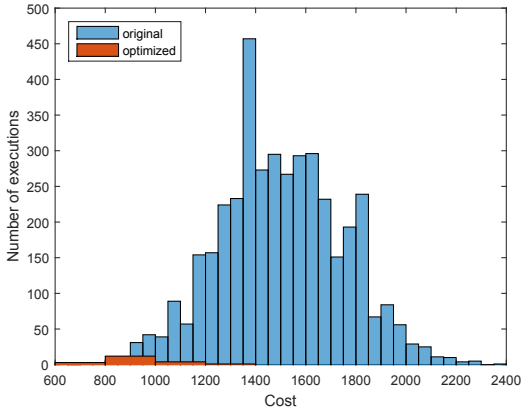

Figure 9. Histogram of the cost of all possible paths searched with the Alg. A (original approach) and the Alg. B (optimized approach) in a workspace containing 24 cells. Using the optimized approach, the cost of the first complete path was 1400 and all the remaining searched paths, whose costs were higher than this value, were discarded without being fully explored.

an excessive number of executions of the recursive function, which is responsible for a considerable time in the algorithm execution.

5. Conclusion

In this paper, we proposed the Energy-aware Grid-based Coverage Path Planning Algorithm (EG-CPP) that aims at minimizing the energy consumption during mapping missions over irregular-shaped areas. We compared the EG-CPP with respect to the original state-of-the-art approach. Measurements on the energy spent by a quadrotor during real flights proved an energy saving of 17%, stating the effectiveness of the proposed approach. Additional simulation experiments in different irregular-shaped scenarios also support this statement. By adopting the energy-aware cost function, we were able to consider not only turning maneuvers, but also complex elements, such as acceleration, deceleration and optimal speed in the coverage path.

Thanks to the energy model, we were able to correctly estimate the energy required to perform a mission with high accuracy (97%). This allowed us to know in advance the amount of energy required to perform a flight. As a notable result, we drastically reduced the computational time of the algorithm up to 99%. For instance, it is possible to compute a minimum-cost path to cover an area divided in 36 cells in $35s$, instead of more than 3 $hours$. Moreover, since the computational time has been drastically reduced, it is possible to run the algorithm on all the starting positions to find the global minimum-cost path in a reasonable amount of time, saving even more energy.

As a future work, we intend to explore the coverage algorithms for regular and irregular-shaped areas in order to obtain a global solution regardless of the workspace. Furthermore, we are exploring dynamic programming to speed up even more the proposed algorithm.

References

[1] Nicola Basilico and Stefano Carpin. Deploying teams of heterogeneous uavs in cooperative two-level surveillance missions. In Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, pages 610–615. IEEE, 2015.

[2] Antonio Barrientos, Julian Colorado, Jaime del Cerro, Alexander Martinez, Claudio Rossi, David Sanz, and João Valente. Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots. J. Field Robot., 28(5): 667–689, 9 2011. ISSN 1556-4959.

[3] Huy X Pham, Hung M La, David Feil-Seifer, and Matthew Deans. A distributed control framework for a team of unmanned aerial vehicles for dynamic wildfire tracking. In Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on, pages 6648–6653. IEEE, 2017.

[4] Alessandro Renzaglia, Christophe Reymann, and Simon Lacroix. Monitoring the evolution of clouds with uavs. In Robotics and Automation (ICRA), 2016 IEEE International Conference on, pages 278–283. IEEE, 2016.

[5] Wenkai Chang, Guodong Yang, Junzhi Yu, Zize Liang, Long Cheng, and Chao Zhou. Development of a power line inspection robot with hybrid operation modes. In Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on, pages 973–978. IEEE, 2017.

[6] Howie Choset. Coverage for robotics – a survey of recent results. Annals of Mathematics and Artificial Intelligence, 31(1):113–126, 2001. ISSN 1573-7470.

[7] Håvard Lægreid Andersen. Path planning for search and rescue mission using multicopters. Master's thesis, Institutt for teknisk kybernetikk, Norway, 2014.

[8] Michael Oborne. Mission planner - ground station. Available online: http://planner.ardupilot.com, 2017. (accessed on 15 June 2017).

[9] Carmelo Di Franco and Giorgio Buttazzo. Coverage path planning for uavs photogrammetry with energy and resolution constraints. Journal of Intelligent & Robotic Systems, pages 1–18, 2016.

[10] Tauã Milech Cabreira, Carmelo Di Franco, P. R. Ferreira Jr., and G. C. Buttazzo. Energy-aware spiral coverage path planning for uav photogrammetric applications. IEEE Robotics and Automation Letters, 3 (4):3662–3668, oct 2018. ISSN 2377-3766.

[11] João Valente, David Sanz, Jaime Del Cerro, Antonio Barrientos, and Miguel Ángel de Frutos. Near-optimal coverage trajectories for image mosaicing using a mini quad-rotor over irregular-shaped fields. Precision Agriculture, 14(1):115–132, 2013. ISSN 1573-1618.

[12] Ivan Maza and Anibal Ollero. Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms. In Distributed Autonomous Robotic Systems 6, pages 221–230. Springer, 2007.

[13] Yan Li, Hai Chen, Meng Joo Er, and Xinmin Wang. Coverage path planning for UAVs based on enhanced exact cellular decomposition method. Mechatron., (5): 876 – 885, 2011. ISSN 0957-4158. Spec. Issue Dev. Auton. Unmanned Aer. Veh.

[14] JF Araujo, PB Sujit, and João B Sousa. Multiple UAV area decomposition and coverage. In 2013 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), pages 30–37. IEEE, 2013.

[15] Yu-Song Jiao, Xin-Min Wang, Hai Chen, and Yan Li. Research on the coverage path planning of UAVs for polygon areas. In 2010 5th IEEE Conference on Industrial Electronics and Applications, pages 1467–1472. IEEE, 2010.

[16] Marina Torres, David A. Pelta, José L. Verdegay, and Juan C. Torres. Coverage path planning with unmanned aerial vehicles for 3d terrain reconstruction. Expert Systems with Applications, 55:441 – 451, 2016. ISSN 0957-4174.

[17] Tauã Cabreira, Lisane Brisolara, and Paulo R Ferreira. Survey on coverage path planning with unmanned aerial vehicles. Drones, 3(1):4, 2019.

[18] Anqi Xu, Chatavut Viriyasuthee, and Ioannis Rekleitis. Efficient complete coverage of a known arbitrary environment with applications to aerial operations. Autonomous Robots, 36(4):365–381, 2014.

[19] Christos Levcopoulos and Drago Krznaric. Quasi-greedy triangulations approximating the minimum weight triangulation. Journal of Algorithms, 27(2): 303–338, 1998.

[20] Fotios Balampanis, Ivan Maza, and Anibal Ollero. Area decomposition, partition and coverage with multiple remotely piloted aircraft systems operating in coastal regions. In Unmanned Aircraft Systems (ICUAS), 2016 International Conference on, pages 275–283. IEEE, 2016.

[21] Fotios Balampanis, Ivan Maza, and Anibal Ollero. Area partition for coastal regions with multiple UAS. J. Intell. & Robot. Syst., 88(2-4):751–766, 2017.

[22] Fotios Balampanis, Iván Maza, and Aníbal Ollero. Coastal areas division and coverage with multiple UAVs for remote sensing. Sensors, 17(4):808, 2017.

[23] Fotios Balampanis, Ivan Maza, and Anibal Ollero. Spiral-like coverage path planning for multiple heterogeneous UAS operating in coastal regions. In Unmanned Aircraft Systems (ICUAS), 2017 International Conference on, pages 617–624. IEEE, 2017.

[24] João Valente, Jaime Del Cerro, Antonio Barrientos, and David Sanz. Aerial coverage optimization in precision agriculture management: A musical harmony inspired approach. Computers and Electronics in Agriculture, 99:153 – 159, 2013. ISSN 0168-1699.

[25] LH Nam, L Huang, XJ Li, and JF Xu. An approach for coverage path planning for uavs. In Advanced Motion Control (AMC), 2016 IEEE 14th International Workshop on, pages 411–416. IEEE, 2016.