Robotica (2018) volume 36, pp. 1144–1166. © Cambridge University Press 2018
doi:10.1017/S0263574718000292

Multi-robot coverage path planning using hexagonal segmentation for geophysical surveys

Héctor Azpúrua†‡*, Gustavo M. Freitas‡, Douglas G. Macharet† and Mario F. M. Campos†

†*Department of Computer Science, Universidade Federal de Minas Gerais, Belo Horizonte, MG 31270-901, Brazil. E-mails: doug@dcc.ufmg.br, mario@dcc.ufmg.br*
‡*Instituto Tecnológico Vale, Ouro Preto, MG 35400-000, Brazil. E-mail: gustavo.medeiros.freitas@itv.org*

(Accepted March 24, 2018. First published online: April 15, 2018)

SUMMARY
The field of robotics has received significant attention in our society due to the extensive use of robotic manipulators; however, recent advances in the research on autonomous vehicles have demonstrated a broader range of applications, such as exploration, surveillance, and environmental monitoring. In this sense, the problem of efficiently building a model of the environment using cooperative mobile robots is critical. Finding routes that are either length or time-optimized is essential for real-world applications of small autonomous robots. This paper addresses the problem of multi-robot area coverage path planning for geophysical surveys. Such surveys have many applications in mineral exploration, geology, archeology, and oceanography, among other fields. We propose a methodology that segments the environment into hexagonal cells and allocates groups of robots to different clusters of non-obstructed cells to acquire data. Cells can be covered by lawnmower, square or centroid patterns with specific configurations to address the constraints of magneto-metric surveys. Several trials were executed in a simulated environment, and a statistical investigation of the results is provided. We also report the results of experiments that were performed with real Unmanned Aerial Vehicles in an outdoor setting.

KEYWORDS: Multi-robot systems; Area coverage; Path planning; Geophysical surveys.

1. Introduction

Interest in and research on Unmanned Aerial Vehicles (UAVs) have grown steadily, especially due to the decreases in the cost, weight, size, and performance of actuators, sensors, and processors. UAVs have a niche that covers a broad set of applications that cannot be fulfilled by other types of mobile robots.

In robotics, autonomous mapping can be defined as the process of acquiring a model of the environment using autonomous mobile robots.[1] There are still several open challenges when using mobile agents for mapping with broad area coverage, and good algorithmic solutions are still needed. Geophysical maps, including magnetic, gravitational and textured digital elevation maps, can be used in different situations. Some practical applications that benefit from such maps are: (i) in geology, robot cooperation, and area coverage enable the detection of minerals of commercial interest beneath the ground; (ii) the military can use these maps to find unexploded ordnance (UXOs), sunken ships and submarines at sea; (iii) in telecommunications, these maps can be used to measure the fingerprints of wireless signals; (iv) in archeology, they are used to create maps of sub-surface buried artifacts.

Aerial mapping is one of the most versatile mapping methods. Aerial surveys can use a wide variety of sensors, such as cameras, lasers, radio receptors, barometers, and magnetometers. Currently, most aerial surveys are created using manned aircrafts with expensive and special equipment, making

* Corresponding author. E-mail: hector.azpurua@itv.org

them high-cost and time-consuming tasks, as they generally must be performed over large areas. Thus, autonomous aerial robots can provide many benefits to workers and companies. Mapping with autonomous robots can reduce the risk to human lives and the economic costs of the task.

Here, we consider the use of teams of small autonomous aerial vehicles that can be easily replaced if necessary and have lower costs than their manned counterparts. Multi-robot systems provide several advantages over single-robot operations, including increases in efficiency, flexibility, and robustness. For example, available robots can be reallocated to perform tasks previously allocated to robots that become unresponsive or damaged. Additionally, several simultaneous robots can be configured to survey using different resolutions, heights, and coverage patterns thereby, improving the quality of the resulting map.

In this work, we consider the area coverage problem, which can be defined as the maximization of the total area covered by the robot sensors.[2] Despite being a classical problem tackled in robotics, there are still open challenges to overcome, especially when considering multi-robot systems and cooperative strategies. Some of the open challenges remain open such as task allocation, complete collaborative coverage, failure tolerance, decentralized cooperation, etc. In particular, this work focuses on overcoming the limitations of previous works on a fast method for multi-robot coverage that can be used on geophysical surveys. The proposed method is mainly developed to perform magnetometric acquisitions, and to such, it supports different coverage angles, regions with different height, obstacle avoidance and a mixture of sensors and coverage patterns.

This paper proposes an area coverage methodology using multiple aerial vehicles that minimize survey time and is easily customized for different types of aerial surveys, vehicles, and resolutions. The technique consists of segregating the environment into hexagonal cells, which are then grouped and allocated to different teams of robots. A path-planning algorithm has also been developed to generate efficient coverage routes within each cell. The principal use cases explored here include a magnetic survey and a photometric mapping application using small autonomous quadrotors.[3] The approach has been thoroughly evaluated in different simulated and real-world experiments, thus, validating the efficiency and reliability of the technique.

The remainder of this paper is structured as follows: A discussion of related works is presented in the next section, followed by Section 3, in which we describe the proposed methodology for cooperative area coverage. The experiments and results are presented in Section 4, and finally, in Section 5, we draw our conclusions and discuss paths for future investigation.

2. Related Work

Path planning is a fundamental task for autonomous mobile robots. The quality and efficiency of a mapping task (e.g., resolution, energy consumption, and execution time) are highly dependent on efficient area coverage. Designing efficient coverage routes is required or at least highly desirable in many scenarios, including aerial surveys, landmine deployment or clearance, recognition of changes in terrain, and learning tasks, among others.

2.1. Area coverage

The coverage problem is a classical problem in robotics. This problem is related to the Covering Salesman Problem (CSP), which can be defined as a variant of the classical Traveling Salesman Problem (TSP),[4] where instead of visiting each city, the salesman has to visit a neighborhood in every city such that the total distance is minimized.[5] The main difference between the CSP and the coverage problem is that in the case of area coverage, the agent must go through all of the points that belong to the environment instead of visiting all of the neighborhoods. In the literature, different approaches to the coverage problem have been developed, including comparisons of this problem to the Art Gallery Problem[6] and the Watchman Route Problem.[7] Some of the best-known approaches to area coverage in robotics are distance transforms cell decomposition, spanning trees, the use of template models, and neural networks.[8]

There are a few requirements to perform a proper coverage of a region with a robot: the robot needs to be able to reach all the points that describe the area, needs to traverse the region using optimal and simple paths without overlaps or repeats, and should avoid any obstacles.[9] Additionally, there are other constraints related to the characteristics of specific robots and environments, such as autonomy optimization and the availability of prior knowledge about the region. Execution time and vehicle

https://doi.org/10.1017/S0263574718000292

autonomy can be critical in defining the type of trajectory that will be generated as small unmanned robots may have very limited autonomy.

Coverage techniques that adhere to these requirements can be mainly classified as either online or offline. Online coverage relies on the robot sensors, and the coverage path is updated in real time given the information acquired by those sensors. Offline coverage relies on previous knowledge of the scene, which is used to compute a path that the robot will follow. Other coverage classification techniques consider the type of algorithm (exact, heuristic, and probabilistic approaches), how the environment is evaluated (cellular decomposition, topological, grid, and graph-based methods), the extent of the coverage (complete or incomplete), and information regarding the vehicles used (one vehicle or multiple vehicles).[10,11] In robotics, complete coverage can be defined as the guarantee that if there is a robot available, the region of interest will be completely covered in a given amount of time.

2.2. Single-vehicle area coverage

In the literature, several coverage patterns have been evaluated, and the lawnmower pattern has been determined to be one of the most efficient methods.[12–14] This pattern is simple to implement, as it is composed of several parallel straight lines separated by a space. Cell decomposition can also be used in single-vehicle coverage production to allow for the avoidance of obstacles, in some cases reducing the coverage time by generating paths that allow the robot to maintain their maximum speed for a longer period.

The lawnmower pattern can be used in conjunction with cellular decomposition to increase the coverage speed using holonomic robots, such as quadrotors. In Huang *et al.*,[15] the area to be covered was sub-divided into different regions, and on every decomposed cell, the coverage trajectory can change angles to allow the robot to maintain its maximum cruise velocity for more time, thus, minimizing time-consuming turns.

Some studies have proposed the use of genetic algorithms for planning coverage strategies in structured environments. In Jimenez *et al.*,[16] the authors proposed a trapezoidal decomposition of the environment and a subsequent optimization of the paths between cells to generate the shortest path. Jan *et al.*[17] proposed the use of vertical cellular decomposition with a convex hull to achieve area coverage in real time. The proposed approach uses a Minimum Spanning Tree to find the cell visit order and uses the lawnmower pattern for achieving coverage within the cells.

Other works have adapted ground coverage solutions for aerial coverage. Xu *et al.*[18] proposed the use of Boustrophedon cell decomposition[19] on the region of interest to generate a Euler circuit for all the connected cells, after which a lawnmower path is used to comb the cells. The authors proposed a wavefront planner that uses square cell decomposition with a simple breadth-first search on the resulting graph to generate the routes.

Lin *et al.*[20] proposed a coverage algorithm using Mixture Gaussian Models. The authors demonstrated a method to prioritize sub-regions in which the robot has a higher probability of finding a human. The use of a spiral pattern is proposed to visit these regions, and a difficulty map its updated with the probabilities of finding humans given the characteristics of the terrain.

Patents that address the cooperative coverage have been issued. Meuth *et al.*[21] proposed a technique for cellular decomposition using different approaches, including square, hexagonal, and Voronoi cells. The system optimizes regions for heterogeneous robots while decreasing the coverage time. This particular approach considers changing environmental conditions and robot capabilities and updates the map in real time to account for such changes.

2.3. Multi-robot area coverage

Cooperation is a very important factor when dealing with large areas to be covered and small robots. Robotic cooperation can be defined as a group of robots working together to perform one or several tasks to increase the efficiency and robustness of the tasks.

Carneiro *et al.*[22] proposed a complete coverage technique using space-filling curves with Worm and Hilbert patterns.[23] The authors divide routes among a group of robots, and the robots communicate and share information when they are closer to other robots than to the edge of the region. In Rekleitis *et al.*,[24] the region of interest is divided using the Boustrophedon algorithm, and the technique is extended for multiple robots. The algorithm considers robots with multiple behaviors (cover and explore) to cover a cell, while an auction selects the most appropriate robots to engage tasks.

https://doi.org/10.1017/S0263574718000292

The use of Spanning Trees is a typical approach in cooperative coverage generation. Gabriely *et al.*[25] proposed sub-dividing a region of interest into cells and then allocating those cells using a Spanning Tree. Hazon *et al.*[26] proposed the use of Multiple Spanning Trees to allocate the paths between the robots, a method that is robust to robot failure. Agmon *et al.*[27] used Spanning Trees with optimal structures and improved the coverage time. Breitenmoser *et al.*[28] proposed the use of decentralized Voronoi cell segmentation for non-convex environments to produce area coverage.

Some studies have implemented biologically inspired techniques to tackle the cooperation problem in coverage, such as neural networks or pheromone communication.[29,30]

2.4. Aerial coverage

Aerial cooperative coverage has been an area of great research interest over the last few years due to the advances in and miniaturization of components and sensors.[31] In this context, the work of Maza *et al.*[32] is one of the first to propose a cooperative coverage approach using such vehicles. This study employed a polygonal area decomposition with heuristics to assign routes among robots considering their autonomy limitations. Next, a collection of straight lines with the maximum possible length is generated, with the assumption that robots do not waste time on curves and remain at full speed most of the time.

Santamaria *et al.*[33] developed a path-planning algorithm for multiple UAVs based on cell decomposition. The allocation of the cells depends on the sensor footprint of each aircraft. Jiao *et al.*[34] sub-divided the region of interest with a greedy algorithm that minimizes the number of turns inside every sub-area. The authors also proposed the generation of sweep lines with the longest length possible, after which the visit order of the cells is calculated using an undirected graph.

Avellar *et al.*[35] proposed an algorithm that does not segment the environment. Initially, a collection of waypoints is assigned to the boundary of the region of interest, and then a graph composed of these waypoints is created. Finally, the original problem is formulated as the Vehicle Routing Problem (VRP).[36] The main contribution of this study is the incorporation of specific features that are relevant in a real-world deployment, such as a setup time. However, the study does not consider cases where the number of vehicles or their autonomy is not sufficient given the size of the area to be covered.

Franco *et al.*[37] presented a method for deriving an energy model of an aerial quadrotor to calculate the velocity that minimizes energy consumption during flight on a custom lawnmower coverage path. This model also introduced fail-safe mechanisms that trigger with greater accuracy compared to traditional mechanisms based on voltage thresholds.

2.5. Summary and comparison

The methodology proposed in the present work can be classified as a complete coverage approach using a customized cell decomposition algorithm based on regular hexagons. The hexagons are clustered and allocated to teams of robots. Our approach optimizes the inner and outer visiting points of the sub-areas by modeling them as a graph problem. The main differences to other similar works can be observed in Table I.

3. Methodology

Developing specific procedures to efficiently cover regions of interest is necessary due to the limitations of small aerial vehicles, including low levels of autonomy and small payloads. The features of the region of interest, for example, very large regions or regions with non-flyable zones, also have direct impacts on the routes generated. Our main focus in this work is the development of efficient and feasible routes given the particular constraints related to aerial magnetic surveys, such as flight altitude, coverage angle, and speed.[42] We propose this method, based on hexagonal segmentation and different coverage patterns, as an extension of the magnetic surveying approach from a previous study[3] examining multi-robot and cooperative scenarios.

A high-level diagram of the methodology steps is presented in Fig. 1. Each topic is further detailed in the following sections.

3.1. Region definition

The first step in planning area coverage is the definition of a region of interest. This region is defined by the list of its vertexes $V = \{v_1, v_2, v_3, \ldots, v_n\}$. A closed list of vertexes generates a polygon

https://doi.org/10.1017/S0263574718000292

Table I. Related work comparison.

Work	Method	Type	Cooperative	Complete	Aerial
[20]	Heuristic	Online			•
[16]	Genetic algorithm	Offline		•	
[38]	Cellular decomp.	Offline			
[15]	Cellular decomp.	Offline		•	
[21]	Cellular decomp.	Offline	•	•	•
[18]	Cellular decomp.	Offline		•	•
[17]	Cellular decomp.	Offline		•	
[24]	Cellular decomp.	Offline	•	•	•
[32]	Cellular decomp.	Offline	•	•	•
[33]	Cellular decomp.	Offline	•	•	•
[22]	Space filling curves	Offline	•	•	
[39]	Graph based	Offline	•	•	
[40]	Graph based	Offline	•		•
[35]	Graph based	Offline	•		•
[25]	Spanning tree	Offline	•	•	
[26]	Multiple spanning tree	Online	•	•	
[41]	MRFC	Online	•	•	•
Ours	Hexagonal cellular decomp.	Offline	•	•	•

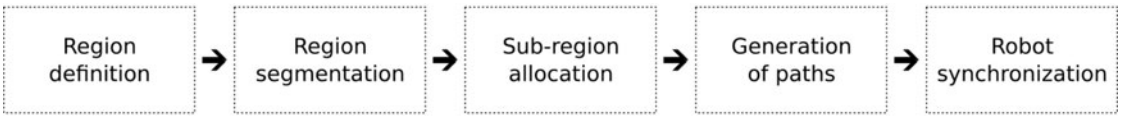


Fig. 1. High-level diagram of the methodology.

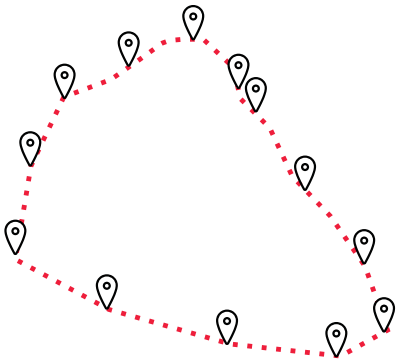


Fig. 2. A list of vertexes is used to define a polygon P.

P, which can be convex or non-convex. This polygon is generally defined manually by an experienced user depending on the type of survey required. Figure 2 illustrates the vertexes that enclose the region of interest.

3.2. Region segmentation

We propose the use of regular hexagons for decomposing the region of interest. This approach confers several advantages compared to the other segmentation techniques mentioned previously, especially in tasks involving surveys with height and orientation constraints like aerial magnetometric surveys.

The region segmentation process is described in Algorithm 1. First, a hexagonal grid is generated by finding the minimum bounding rectangle that covers the convex hull of the desired region of interest denoted by the P polygon. Then, the bounding rectangle is rotated to be parallel to the horizontal plane. The hexagon tessellation begins from the upper left corner of the bounding box with a spacing equal to the diameter of the desired hexagon size ($2 * r$ where r is the desired hexagon radius). With

https://doi.org/10.1017/S0263574718000292

the hexagon list, the intersections of the hexagons on the grid with P are then calculated. That hexagon sub-group (H') is used as input for the next steps of the method.

Algorithm 1 Region segmentation(r, P)

1: $P_{\text{convex}} \leftarrow getConvexHull(P)$
2: $BBox \leftarrow getBoundingBox(P_{\text{convex}})$
3: $H \leftarrow generateHexagonGrid(BBox)$
4: $H' \leftarrow getIntersections(H, P)$
5: $C \leftarrow getCentroids(H')$
6: **return** H', C,

Generating cooperative coverage using robots with limited communication range and battery autonomy requires efficient coverage patterns that improve communication and reduce the coverage time. Regular structures, such as regular hexagons, make it easier to determine such routes since the route only needs to be planned once for each set of hexagons. Some assumptions are taken into consideration as real battery consumption depends on several factors such as wind, battery state, sensors consumption, and environment influence. We consider that the autonomy of the robot is directly related to the paths performed at one fixed altitude and there are no external environment influences that affect this measurement.

A hexagonal grid best suits tasks that aim to cover a region with circles;[43] in our case, the circles represent the communication range of the robots. In a field of wireless networks, it has been shown that in comparison with other structures, such as squares, the use of hexagons requires a fewer polygons to cover the same area.[44] In the present work, the hexagonal shape maximizes the level of communication between the robots inside every coverage sub-region.

Hexagons are used in the generation of geodesic grids over squares. Square cells have the disadvantage that their diagonal neighbors have different distances to their centroids than their other direct neighbors.[45] Hexagons have the advantage of having a fixed distance to the centroids of their six neighbors. Other types of grids, such as triangular grids, have the disadvantages in the orientation of the polygons, which can add more complexity to the calculations. If required, the resolution of the map using hexagons can be regulated using multiple resolution techniques, such as hierarchical grids.[46]

The selection of regular hexagons as the cellular shape for sub-dividing the region of interest is supported by the following factors:

- For the coverage, we wish plane tiling with convex regular polygons—so that the centroid distance is the same, no empty spaces, no need for rotation, one-time calculations.
- There are only three such polygons (hexagons, squares, and triangles). Out of the three of them, hexagon shape is the most similar to the robot's communication range (the error between tile and communication area is minimized for the hexagons).

Cellular segmentation can also allow some interesting surveying approaches that meet many constraints in geophysical applications:

- Allows for the avoidance of flights over certain cells. This can be used to avoid obstacles or to prevent flying over hazards (Fig. 3(c)).
- Allows for the easy configuration of different heights, increasing the surveying resolution if needed in specific regions, as illustrated in Fig. 3(a). It also allows for the generation of paths that follow the inclination of the terrain, as shown in Fig. 3(b).
- Since it is possible to directly estimate the battery consumption of a robot covering a hexagonal sub-region, the size of the hexagonal cells can be defined to better accommodate the flight autonomy of the robots.
- When conducting cooperative flights, the cells allocated to robots that become inoperative can be reallocated to the remaining available robots, thus, guaranteeing complete coverage when at least one additional robot is in operation.

https://doi.org/10.1017/S0263574718000292

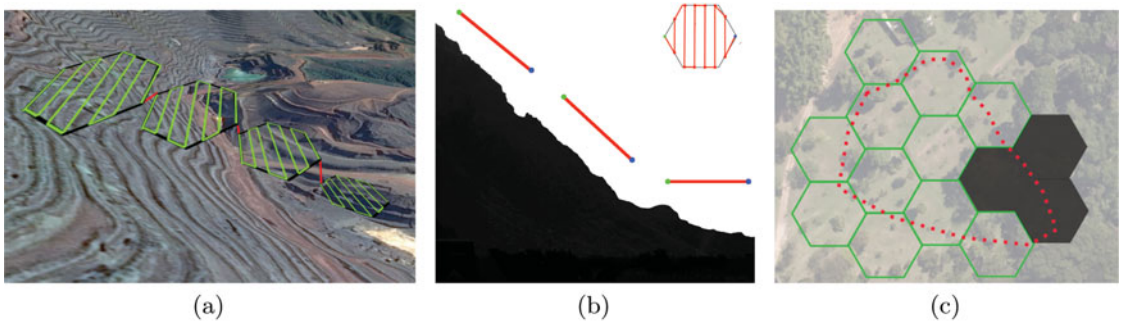


Fig. 3. (a) Different heights can be configured for every cell, e.g., following the descent of a hill. (b) Sub-regions can be configured to have different angles. (c) Certain regions can be configured as no-fly zones (black cells).

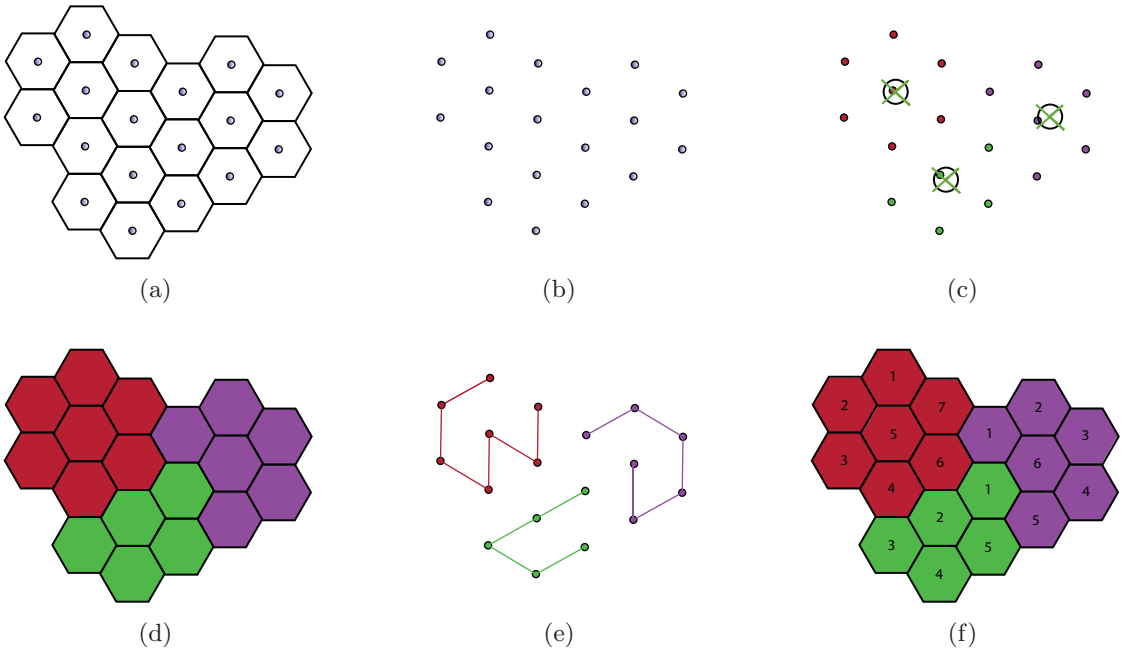


Fig. 4. Clustering steps. (a) The C centroids of the hexagons in H'. (b) The C centroids are used as input for the *K-means* algorithm. (c) The clustering algorithm returns k clusters. (d) The classification is used to label the hexagons in H'. (e) The centroids are used as visit points for a TSP solver. (f) The hexagons are labeled according to their visiting sequence.

3.3. Sub-region allocation

To equally divide the hexagons between the k robot teams, we use the *K-means*[47] clustering algorithm. This algorithm groups or labels the elements closer to the k seeds $S = \{s_1, s_2, \ldots, s_k\}$, updating their positions at every iteration until the seeds are stabilized.

The proposed method uses the C centroids (Fig. 4a) as an observation group in a two-dimensional (2D) matrix (Fig. 4b). The *K-means* algorithm is applied to these data 100 times, and the most homogeneous solution is chosen (Figs. 4c and d). For each cluster, a graph is modeled as an instance of the TSP. The solution to the TSP is the minimum Euler path that visits all points, and this is used as input in the next step.

3.4. Generation of paths

In many different geophysical applications, the use of multiple, and possible heterogeneous robots, allows the possibility for the execution of different coverage patterns with different sensors. Those sensors could vary in accuracy, size, weight, and type, thus, allowing a more valuable data acquisition. In this work, we consider the use of a team of one, two, or three robots. Multiple robots per team

https://doi.org/10.1017/S0263574718000292

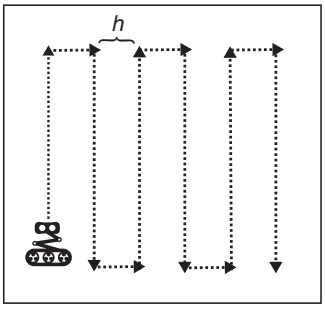


Fig. 5. Example of a lawnmower pattern. The h distance must be defined for every lawnmower generation.

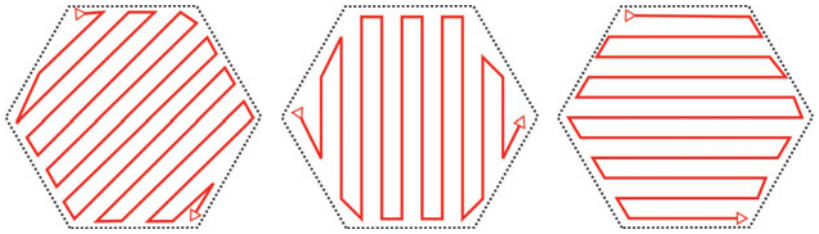


Fig. 6. Example of hexagon rotations. (a) 90°. (b) 50°. (c) 180°.

allows a better sensor fusion during the map generation phase, reducing noise, and improving the quality of the survey.

Initially, we discuss the simpler case in which the team is composed of a single robot, after which we consider teams of multiple robots.

3.4.1. Single-robot teams. When the T teams of robots are composed of one robot ($\gamma = 1$), the lawnmower pattern is chosen as the basic coverage pattern (Fig. 5). This pattern can completely cover a region and has an efficient coverage time.[12–14,32] In this work, we present a version of the lawnmower pattern generation that represents an improvement over that in Perez-imaz *et al.*[48]

The first step is to define the width of the parallel coverage lines (the h value). This value depends on the type of the survey and must be chosen by an expert. This choice must account for aspects such as sample velocity, flight altitude, and sensor coverage. For example, in aerial photogrammetry, the angle of view and height dictate an acceptable value of h. In aerial magnetometry, the size of the expected magnetic anomaly will dictate the h value: for small anomalies, such as landmines, a very small value of h, such as 1–2 m, will be necessary. When surveying large areas to find ferrous minerals, larger sizes of h are recommended.[42]

After selecting h, the next step is the hexagon rotation angle calculation. For time constrained surveys, we propose a method to automatically generate the set of angles (one angle for every hexagon) that minimize the coverage path. This angle optimization allows the robot to reduce the time spent when changing cells in broader regions of interest. The optimization is more relevant when the size of the hexagons is small; this usually means that there are a significant number of cells to visit. In this work, we consider a hexagon rotation as a rotation of the hexagon coverage path among its centroid by a multiple of 10° (Fig. 6).

Those different angle rotation configurations are desirable depending on the survey restrictions. For example, for magnetic surveys, following a specific North–South or East–West orientation when surveying is recommended,[42] as the influence of the magnetic field of the Earth changes depending on the hour and the latitude/longitude of the region.

To generate the lawnmower patterns at a fixed angle the Algorithm 2 is executed. r is the hexagon radius, α the desired rotation angle for all the hexagons, and h is the lawnmower distance between parallel lines.

The algorithm starts by calculating the point of a regular hexagon by a given radius (*generateHexagon* function). Then, this hexagon is rotated by α degrees to simplify the lawnmower

https://doi.org/10.1017/S0263574718000292

Algorithm 2 Generate lawnmower at fixed angle(r, α, h)

```
P ← generateHexagon(r)
P ← rotate(P, α)
x_max, x_min, y_max, y_min ←getMaxMinBoundaries(P)
y_diff ← y_max − y_min
segments ← y_diff/h
points ← getLineIntersections(x_min, h, segments)
R ← {}
i ← 0
while i < |segments| do
    p ← points_i
    if i + 1 < |segments| then
        p ← P ∪ points_{i+1}
        i+ = 2
    else
        i+ = 1
    for j ∈ {1, ..., |p|} do
        intersect ← getIntersection(P, x_min, x_max)
        if |intersect| > 0 then
            ipoints ← getInternalPoints(intersect, h)
            if (i + j)mod2 == 0 then
                ipoints ← reverse(ipoints)
                R ← R ∪ ipoints
R ← rotate(R, −α)
return R
```

path generation. Then, the hexagon is divided into small segments of h distance, and for every segment, the maximum upper and lower bounds that lie inside the hexagon are calculated. Every segment is sub-divided in h smaller sections using the Bresenham algorithm. Those smaller portions help the drone to know where to acquire the magnetic information. In our implementation, the drone must capture sensor data every new waypoint or segment start.

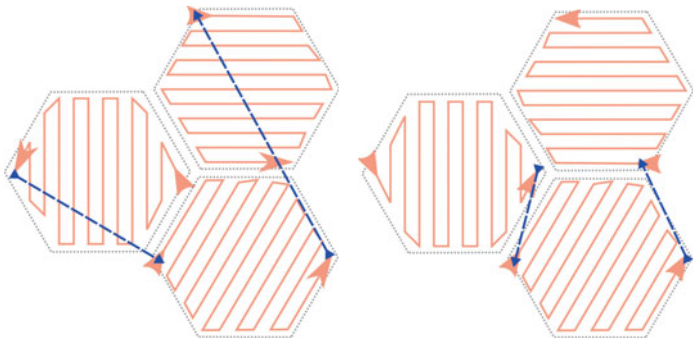
As stated before, if the survey is time constrained, we can evaluate the possible rotation of every lawnmower path inside the hexagons to determine the configuration that has the shortest path between every pair of contiguous cells in L_i. This problem was modeled as a graph $G = (W, N)$, in which all the possible rotation configurations of every hexagon in $L_i \mid i \in \{1, \ldots, k\}$ represent the vertexes V, and the edges between two vertexes (u, v) are the Euclidean distances between the exit point in u and the entry point in v after their respective rotations, as seen in Fig. 7. The entry and exit points are the start and end of the lawnmower path of a hexagon. The possible rotation angles are defined in the list $\Theta = \{o_1, o_2, \ldots, o_z\}$, where every element o_i is a different rotation angle and z is the total number of rotation angles.

Once each hexagon is labeled with a coverage angle, Algorithm 2 is then used to acquire the lawnmower path.

To verify whether the path can be effectively executed by the robot, the use of Algorithm 3 is proposed. This algorithm uses the list of cells of a robot ($Cell$), the capacity of a robot $Autonomy$, the battery consumption to visit a hexagon ρ, and the location home position $Home$ as inputs. The algorithm iteratively tests if the robot can perform a total coverage of the cell list. If at one point, the robot cannot cover a cell completely, it returns home to change batteries. If the robot autonomy is not sufficient to go back and return to the coverage, then that path cannot be performed. The algorithm returns a boolean value that identifies whether the robot can complete the entire path or not and a list of the positions where the robot needs to go back to the home position and recharge batteries. The *CalcDist* function calculates the Euclidean distance between the two parameters.

3.4.2. Multi-robot teams. Some types of surveys, including aero magnetometric surveys, require modification to comply with the limitations of small autonomous robots. This work proposes the use of different coverage patterns with the insight that the integration of all acquired data improves

https://doi.org/10.1017/S0263574718000292



(a) (b)

Fig. 7. Example of angle optimization for smaller total paths. (a) A non-optimized set of angles. (b) The same hexagon topology with the angles rotated to minimize inter-cell distance.

Algorithm 3 Route verification($Cell$, $Autonomy$, ρ, $Home$)

$i = 0$
$j = 0$
$remaining = Autonomy$
$return\_points \leftarrow \{\}$
while $i <$ size(Cell) **do**
    $d_{\text{cell}} \leftarrow$ calculate the distance from home to complete cell i
    $d_{\text{home}} \leftarrow$ calculate distance from the end of cell i and home location
    if $d1 + d2 > remaining$ **then**
        return $False, return\_points$
    else:
        remaining -= d_{cell}
        $j = i + 1$
        while $j <$ size(Cell) **do**
            d_{next} = distance from end of hexagon i to complete next hexagon j
            $d_{\text{next home}}$ = distance from end of hexagon j to home
            if $remaining < d_{\text{next}} + d_{\text{next home}}$ **then**
                $return\_points.append(i)$
                break
            else:
                remaining -= $(d_{\text{next}} + \rho)$
                $j++$
                $i = j$
    remaining $\leftarrow$ battery
return $True, return\_points$

mapping quality. This quality improvement is achieved by minimizing the noise impact through the fusion of different readings, allowing for a differential measuring scheme that also reduces influences from the environment, especially in tasks considering the Earth's magnetic field. The use of several coverage patterns helps to overcome payload and autonomy limitations since routes could adapt to the used sensors.

When the T teams of robots have more than one robot, we propose the use of different coverage patterns simultaneously. The coverage patterns are depicted in Fig. 8.

Two new types of patterns were added, (i) a coverage pattern following a square shape (ii) and a fixed visit to the center of the hexagon. The visit to the centroids serves as a reference to the map (the base value). This robot will stay at the center C of every hexagon, capturing data until the other robots have finished their internal paths in that hexagon. The square coverage pattern is inspired by the control lines for magnetic surveying,[42] over which the robot will perform a slower square path, while another robot performs a faster lawnmower coverage pattern. In this work, we assume that a

https://doi.org/10.1017/S0263574718000292

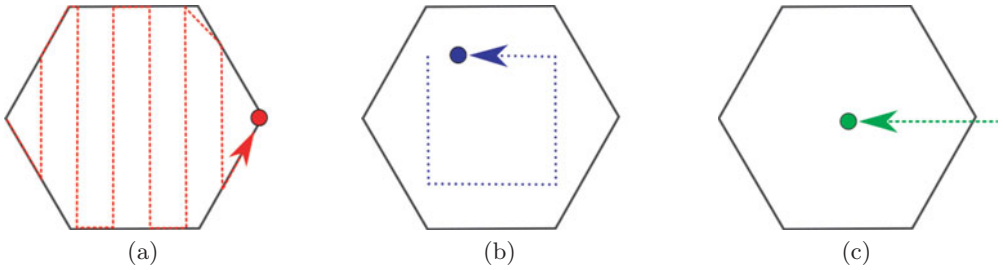


Fig. 8. Coverage patterns: (a) Lawnmower, (b) Square pattern, and (c) Hexagon centroid.

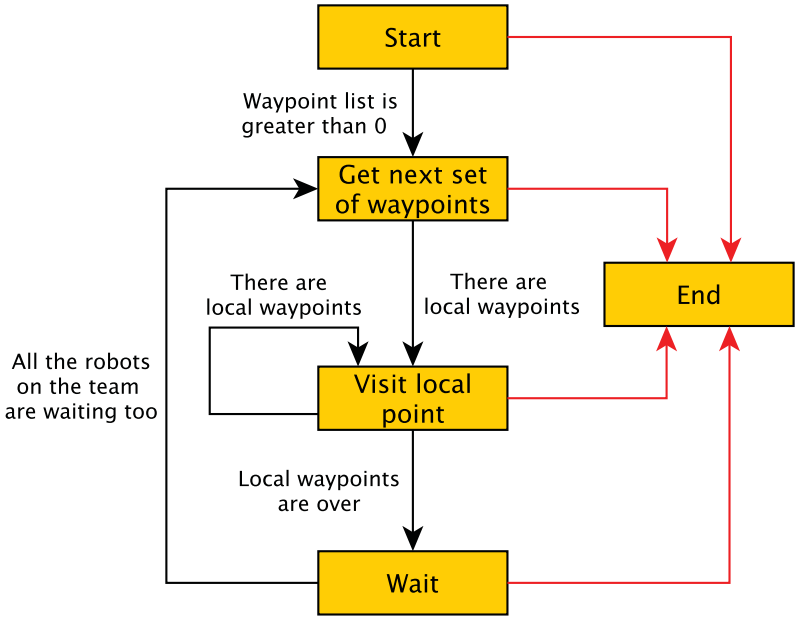


Fig. 9. Finite state machine allowing the synchronization of a team with multiple robots. The red lines are the transitions when there are no more waypoints on the list.

route other than the default lawnmower coverage path will add more sensing points over the region and will thus improve the map quality when interpolating the data.

Therefore, the entire set of paths can be defined depending on the team size:

- 1 robot: Only the lawnmower pattern is used.
- 2 robots: The lawnmower and square patterns are used.
- 3 robots: All the patterns are used simultaneously.

3.5. Robot synchronization

The synchronization of the start of cell coverage between robots is mandatory for aeromagnetic surveys as small variations in the magnetic field of the Earth can introduce unexpected noise in the measurements. Coverage in the same time window will reduce these consistency problems in the data.

We propose a finite state machine (FSM) to coordinate the simultaneous coverage of the hexagons. The proposed FSM can be seen in Fig. 9.

This FSM is applied to every robot. Each team member shares its actual state with its teammates. The FSM begins in the "start" state, and if the robots have been assigned visit points, they separate and prepare to visit in the "visit point" state. If the robot has finished visiting the different points, it will wait for its teammates to finish their visits in the "wait" state, and when all robots have finished, this process is repeated for the next hexagon. When the robots have finished, they proceed to the "finish" state, and the FSM ends.

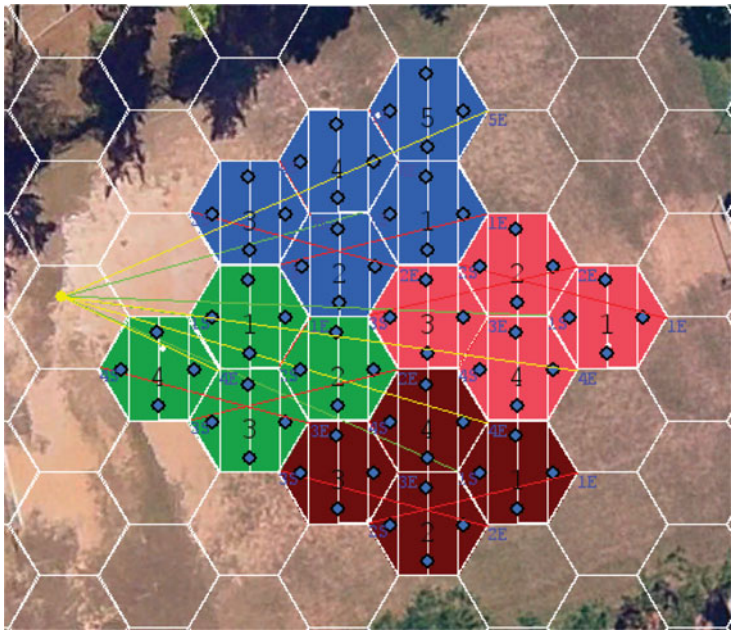


Fig. 10. Example from the simulation framework. In this example, we consider four single-robot teams, each with predefined coverage angles (North to South).

4. Experiments

In this section, we evaluate our methodology as applied in two groups of experiments: simulated and real experiments. The simulated experiments analyze how each of the parameters of the algorithm affects the coverage time. The real experiments analyze the behavior of the methodology using real quadrotors in an outdoor environment.

4.1. Simulated experiments

For the initial phase of the experiments, a simulation framework was developed to facilitate a quantitative analysis. This framework allows for the modification of all the available parameters in the method and the evaluation of the coverage time and behavior of the cooperative teams in real time. The set of simulated experiments will be described in the following sub-sections.

To better visualize the experiments, we initially present an example considering four teams, each comprising a single robot. As a constraint, the robots cover each sub-region following a North–South path. Figure 10 shows a screenshot of the result obtained in the simulated environment.[(1)]

4.1.1. Distance between parallel coverage lines. The distance between the coverage lines inside the hexagons has a direct influence on the total traveled path length. In this experiment, h is modified, and the time for a complete coverage is measured. This test was done using $k = 1$ robots and a size of $r = 25(pixel)$ for every hexagon, with no battery restriction ($b = \infty$). As previously mentioned, on a real task such value may be defined by a specialist.

As shown in Fig. 11, the coverage time is inversely proportional to the size of h: when h is smaller the coverage time increases. The increase in time when h is lower is due to the risen number of turns and lines.

4.1.2. Number of robots. A suitable assumption is that coverage time will decrease with the use of multiple robots. The main objective of this experiment is to verify whether this statement remains true regardless of the number of robots. The experiment was conducted using areas with different sizes: one area was covered by 43 hexagons (dotted blue line), and the other was covered by 110 hexagons (solid red line). Each team is formed by a single robot, and the lawnmower path was used with angle optimization between the hexagons with no battery restrictions ($b = \infty$).

[(1)]A video of the execution of the simulation is available at: `https://youtu.be/BM9Qe4XjJ0k`

https://doi.org/10.1017/S0263574718000292

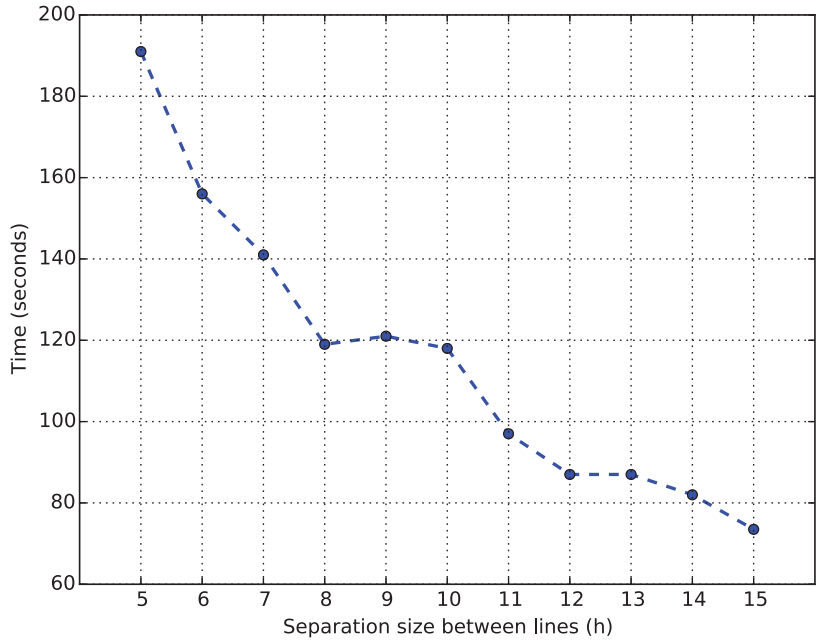


Fig. 11. Results of the experiment of the variation of h vs. coverage time.

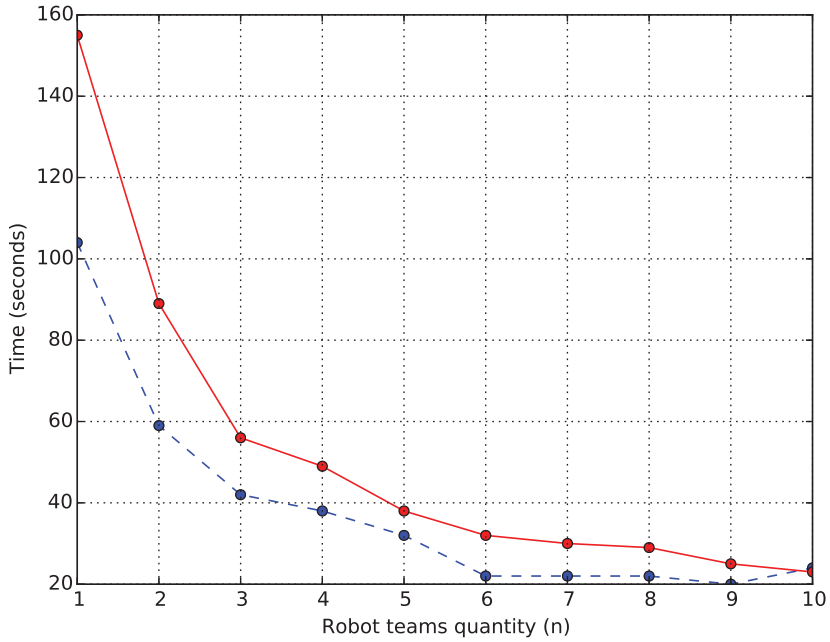


Fig. 12. Experiment varying the number of robot teams. The dotted blue line represents an area with 43 hexagons, and the solid red line represents an area with 110 hexagons.

When the number of robots increases, the coverage time decreases, as shown in Fig. 12. However, there is a clear reduction in improvement when the area to map is small or there is a large number of robots. There is a non-linear relation between the hexagon quantity and the area covered. Since the number of hexagons was modified by reducing and augmenting the radius of the hexagons over the same region of interest, on the experiment with fewer hexagons, cells have a bigger size than the ones of the test with more hexagons.

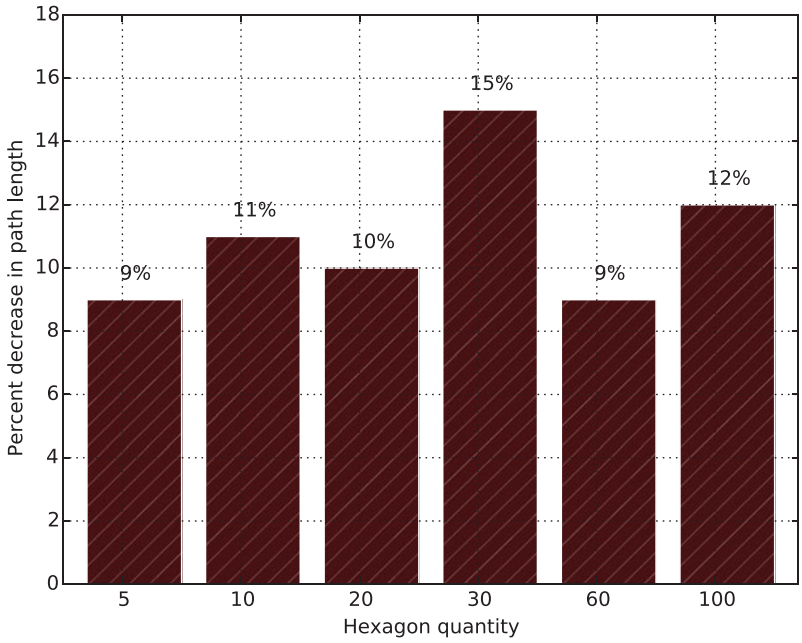


Fig. 13. Experiment comparing the optimization of the coverage angles with random selection of angles.

4.1.3. Coverage angle optimization. The angle optimization between hexagons tries to minimize the path length between the ending point of a hexagon and the start point of the next one by rotating the polygons until the minimum path is found between them.

This experiment compares the reduction gain over a set of randomly selected angles when the optimization is used. The optimization algorithm was executed 30 times for every instance of the test and the mean value is calculated.

As shown in Fig. 13, in a mean of 30 experiments, there is a gain of 11% over the paths that do not use optimization and use a random allocation of rotation angles. Despite the number of hexagons was greatly modified for this test (between 5 and 100 hexagons), the gains are maintained between 9% and 15%. This could be because, with the random selection of rotation angles, the choices will not be inefficient all the time.

4.1.4. Hexagon size and battery. The size of the hexagons and the aircraft autonomy have a direct impact on the total coverage time when the robot needs to come back to the original point to charge its batteries. This experiment uses the list of return points calculated with the Algorithm 3 to allow the robot to decide when to go back to the starting point and charge batteries. This method allows the robots to cover a larger area than when robots are limited to a single path without a battery charge. The total coverage time for each run is compared to the robot teams, varying the battery autonomy and the size of the hexagons.

As presented in Fig. 14, hexagons with a bigger radius will also require robots with an increased battery autonomy. If the hexagons are too big, the robot will not be able to cover one of the hexagons completely with a single battery charge, and thus a valid path cannot be created for the robot. The coverage time also decreases with larger hexagons since the distances between the cells will also decrease (having fewer hexagons decreases inter-total hexagon distances). So, it is paramount to find an equilibrium in hexagon radius and battery autonomy to complete the coverage without wasting resources.

Similar to the results of the previous experiment, selecting the correct parameters, such as the level of autonomy (selection of an appropriate robot or batteries) and the size of hexagons, is fundamental to obtaining the best performance. Furthermore, these parameters have a direct impact on the total coverage time as well as the overall feasibility of the mission.

4.1.5. Route generation with non-flyable zones. In this experiment, we evaluate how the robots performed with obstacles or non-flyable zones defined by the operator manually. We generated

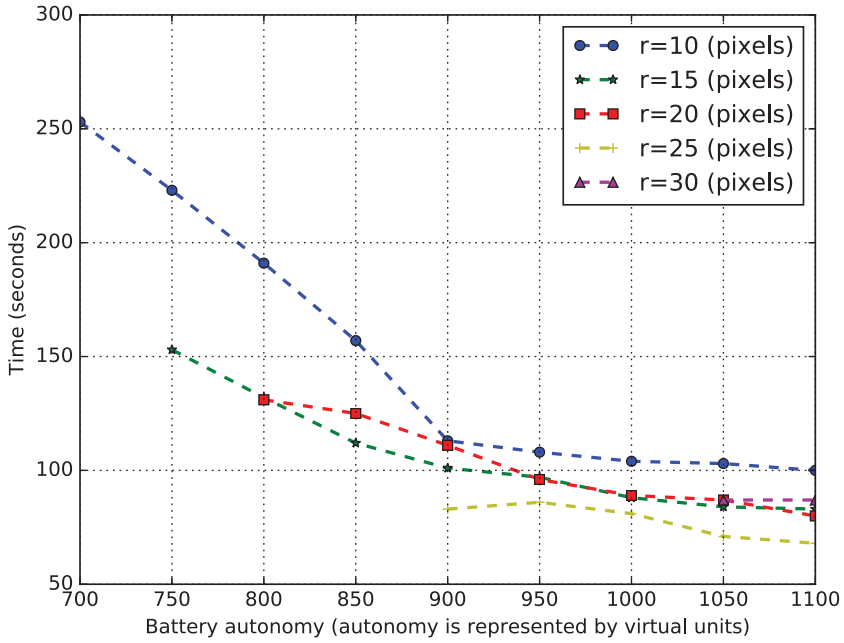


Fig. 14. Results obtained with variations in hexagon size and robot battery autonomy. Each color represents a different hexagon size.

different routes for one, two, and three teams of robots, with and without obstacles. The obstacles are defined by the black cells. As seen on Fig. 15, the method generated feasible routes for the different teams and reallocated the remaining non-occupied cells gracefully among the robots.

4.1.6. Comparative coverage time analysis. In this experiment, the coverage time is compared with a traditional approach for cooperative robots. An algorithm proposed by Maza *et al.*[32] was developed (Fig. 16), where instead of using the polygon segmentation approach presented in ref. [49], we use Voronoi cellular decomposition.

The Voronoi seeds are separated by dividing the total number of points that define the border of the P polygon (the border of the region of interest) between the k teams when $|P| > k$. The (x, y) of every point are obtained by

$$p_i = P_{(|P|/k)*i}. \tag{1}$$

This experiment used a region of interest of fixed size for both algorithms, varying the size and number of hexagons only in the evaluation of the methodology by this work. The distance between the parallel lines of the lawnmower pattern (h) is equal in both algorithms.

A comparison of the results with varying hexagon sizes are presented in Fig. 17. The results show that the algorithm presented in Maza *et al.*[32] results in a significantly lower coverage time. This is because the algorithm does not cover any extra area, which is a consequence of the use of hexagon segmentation, as illustrated in Fig. 18. Additionally, robots do not need to move between sub-regions. As mentioned previously, it is critical to determine an appropriate hexagon radius value.

A comparison of the results of the experiments with different numbers of teams (each team comprising one robot) is presented in Fig. 19. In this experiment, the hexagon size was set to $r = 30$ (pixels). The coverage time decreases with an increase in the number of robots for both techniques, with the technique developed in Maza *et al.*[32] presenting better results. However, as can be seen, the performance difference is minimized with a higher numbers of teams.

Although the proposed method results in higher coverage times than those resulting from the method presented in Maza *et al.*,[32] an important advantage of the proposed method is that it supports both convex and non-convex regions, including regions with obstacles. The technique proposed in Maza *et al.*[32] only supports convex regions and does not include a mechanism to avoid obstacles.

The proposed methodology also offers more flexibility since it can support different altitudes, resolutions, and coverage angles, and makes it possible to cover more area than the autonomy of one

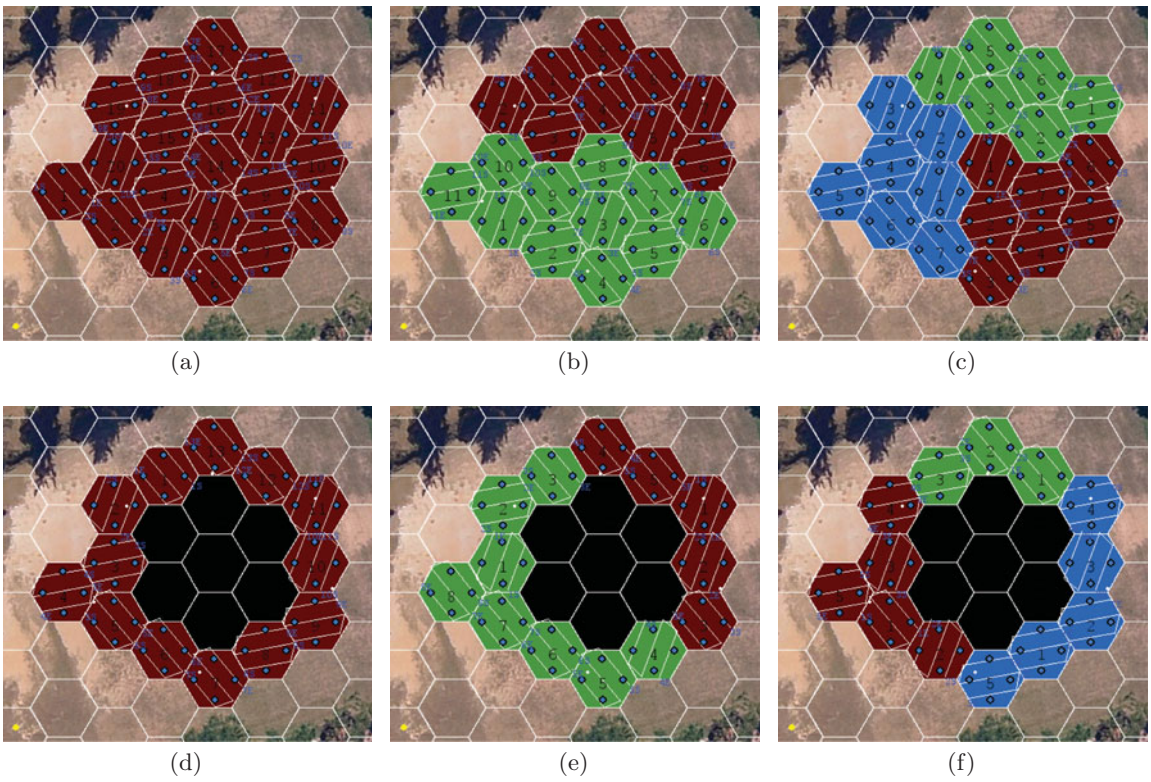


Fig. 15. Route generation with non-flyable zones or obstacles. (a) and (d) Routes for one team. (b) and (e) Routes for two teams. (c) and (f) Routes for three teams.

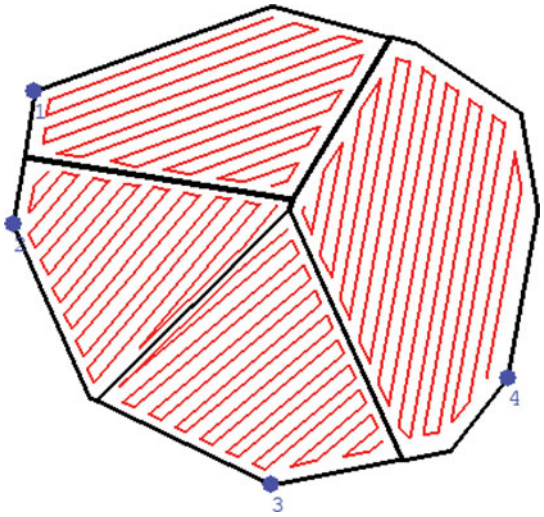


Fig. 16. Example result for the coverage algorithm proposed in Maza *et al.*[32]

robot and one battery allows since the robots can visit the starting point to change/charge batteries and continue with their task. The Maza *et al.* proposal only supports the best coverage angle that minimizes turns, and does not propose any method to fix the angles to a specified position, also since the area covered by that method is bigger for every robot, a detailed terrain following with the robots can be difficult if a height is defined only by cell altitude.

4.1.7. Comparative sensing analysis. This experiment compared the reconstruction quality of simulated magnetic anomalies between the approach presented in Maza *et al.*[32] and the proposed

https://doi.org/10.1017/S0263574718000292

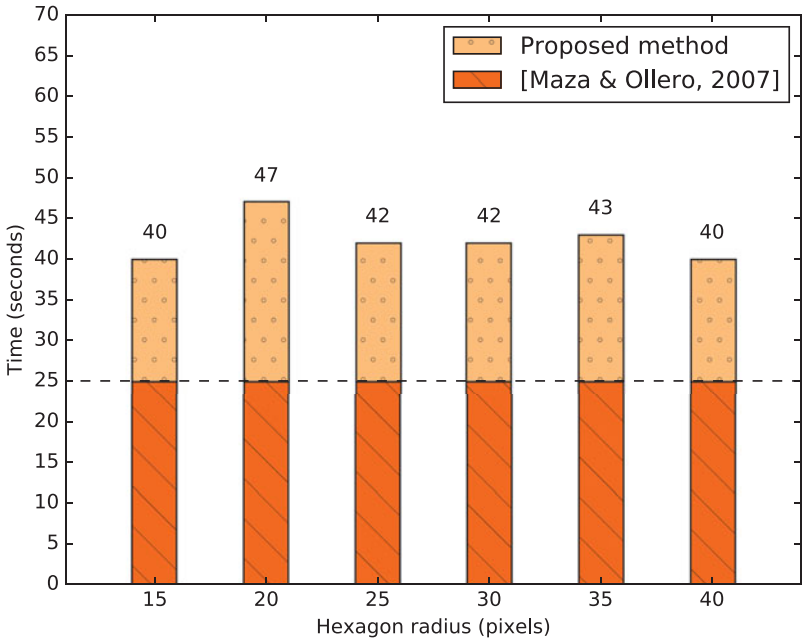
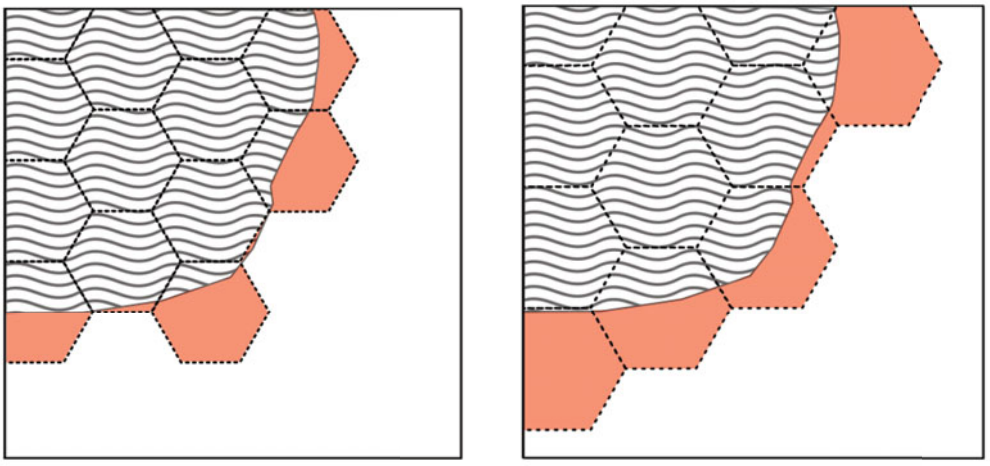


Fig. 17. Comparison of the coverage time between the algorithm used in Maza *et al.*[32] and the proposed method. Both techniques cover the same region. The dark orange bars show the coverage time for one robot covering the full area, and the light orange bars show the coverage time for the proposed method with area segmentation varying hexagon sizes.



(a) Small hexagons. (b) Large hexagons.

Fig. 18. Extra area covered by the hexagons in a region of the same shape and size. The wavy region is the region of interest to be surveyed. The red region is the extra area covered by the hexagons. (a) Small hexagons. (b) Large hexagons.

methodology. The main objective is to identify whether the use of teams of robots is beneficial in mapping reconstructions.

An environment was created with various simulated magnetic anomalies, which are represented by a Gaussian distribution. A matrix with the original Gaussians is used as a ground truth for the magnetic samples, as shown in Fig. 20. In the simulation, every robot obtains small samples of that matrix with the addition of white Gaussian Noise with a standard deviation of 0.25. If several robots have samples of the same location, the mean of those values is calculated.

All robots have a fixed sampling frequency (a sample every 30 pixels) and a sample size of 5×5 pixels. The robot that performs the visit path to the centroids has a larger sample size of 15×15 pixels

https://doi.org/10.1017/S0263574718000292

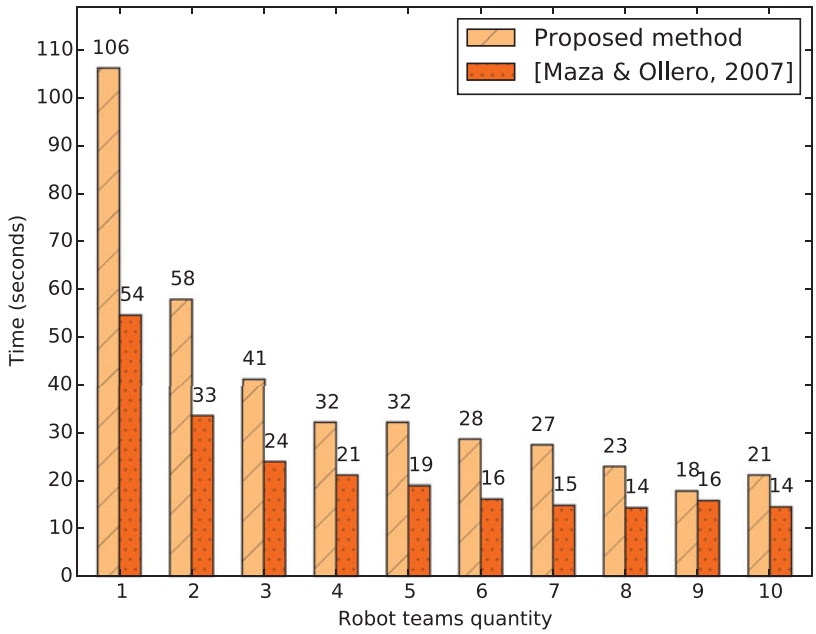


Fig. 19. Comparison of the coverage times with different numbers of robot teams using the method in Maza *et al.*[32] and the proposed method.

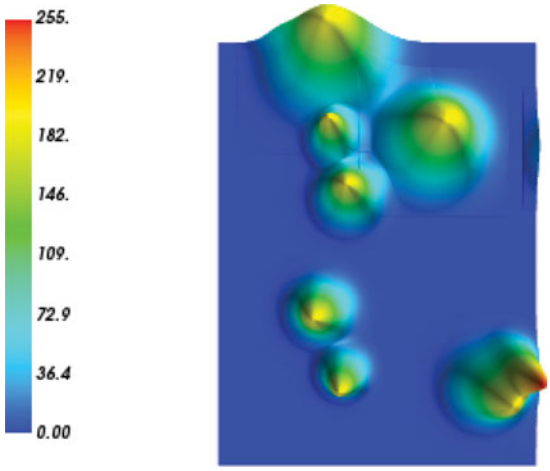


Fig. 20. Gaussians in a 2D plane with dimensions of 600 × 800. These Gaussians are used as ground truths for the simulated magnetic survey.

owing to the insight that a robot moving slowly will have a greater sensing range than one traveling at greater speeds.

The samples obtained by the robots are smoothed using a Gaussian filter with $\sigma = 3$. This minimizes the effects of holes between magnetic samples. Finally, the samples are normalized to a range of 0–255, and a histogram of the samples is generated. These histograms are compared using the Bhattacharyya distance measure[50,51] to find the histogram that is most similar to that of the ground truth. The algorithm returns 0 when the histograms are equal and a value greater than 0 when the values differ.

The comparison between the samples and the ground truth data show that the method proposed in this work generates a more representative map in every case. The quality of the reconstruction also increases when more robots are used, as shown in Fig. 21.

The increased quality of the reconstruction owes to the fact that with an increased quantity of robots, more data is generated, and in the case of the proposed methodology, the extra area covered helped acquire more samples. The quality of the reconstruction is also higher when the distance between

https://doi.org/10.1017/S0263574718000292

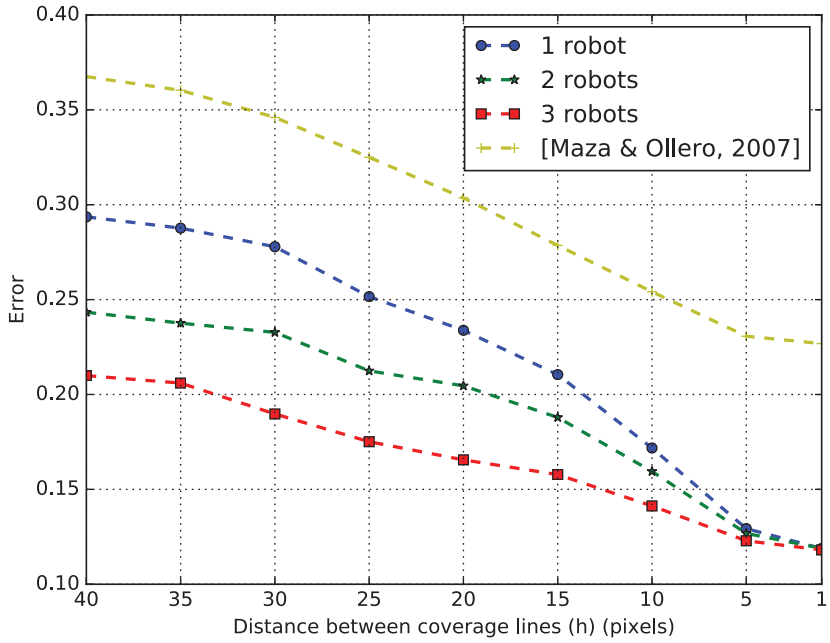


Fig. 21. Difference between the reconstructed map and the ground truth using the proposed methodology with 1, 2, and 3 robots and the method presented in Maza *et al.*[32]

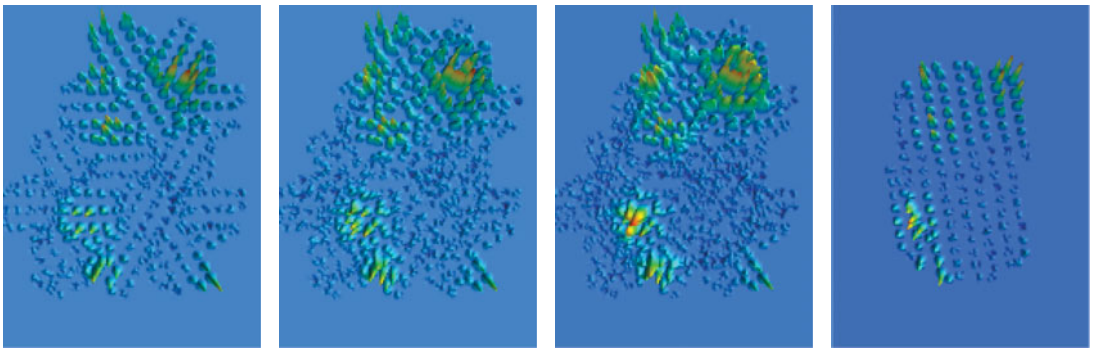


Fig. 22. Results of the magnetic map reconstruction with a distance of 40 pixels between coverage lines. The proposed method with (a) 1 robot (lawnmower pattern); (b) 2 robots (lawnmower and square patterns); and (c) 3 robots (lawnmower, square and centroid patterns). (d) The method proposed in Maza *et al.*[32]

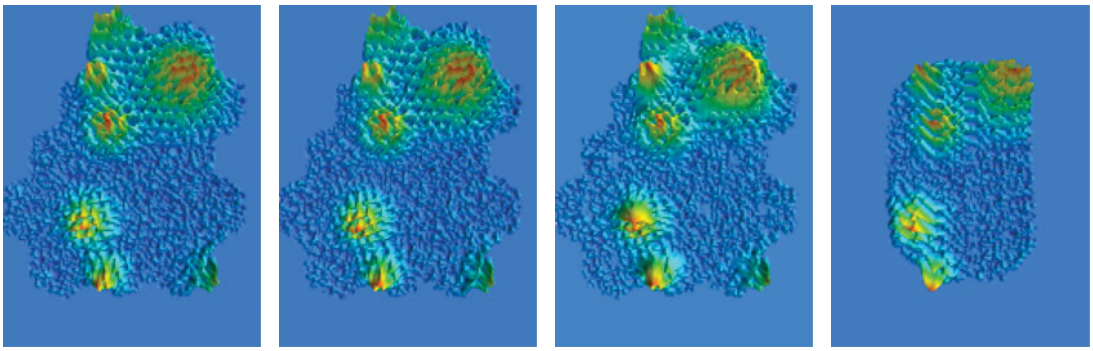
coverage lines is smaller, resulting in robots traveling between closer points and producing more overlap in the samples.

Examples of map reconstructions are presented in Fig. 22 (a distance of 40 pixels between coverage lines) and in Fig. 23 (a distance of 5 pixels between coverage lines). A correlation between h and map quality can be seen in the reconstructed maps. When the proximity between the coverage lines is increased, the reconstruction is very similar to the ground truth, but the method proposed in Maza *et al.*[32] exhibits less similarity.

4.2. Real experiments

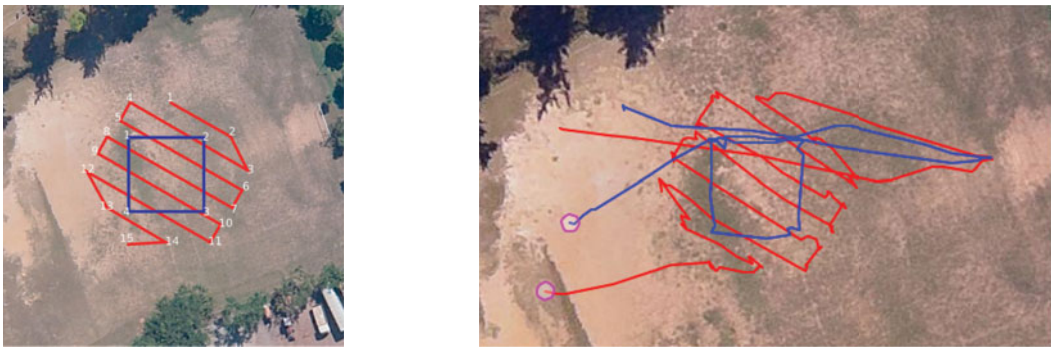
We also conducted different proof-of-concept experiments to demonstrate the feasibility of the planned paths in real-world conditions. These experiments consist of real quadrotors executing calculated paths in an outdoor environment. A group of two Hummingbird quadrotors[(2)] was used, and the framework was developed on top of the ROS[(3)] platform. The robots performed the mission autonomously using

[(2)]Ascending Technologies, http://www.asctec.de/
[(3)]Robot Operating System, http://www.ros.org/



(a) (b) (c) (d)

Fig. 23. Results of the magnetic map reconstruction with a distance of 5 pixels between coverage lines. The proposed method with (a) 1 robot (lawnmower pattern); (b) 2 robots (lawnmower and square patterns); and (c) 3 robots (lawnmower, square and centroids patterns). (d) The method proposed in Maza *et al.*[32]



(a) Planned path. (b) Executed path.

Fig. 24. Real-world experiment 1: Cooperation between a single team of two robots. The red line denotes the lawnmower pattern, the blue line denotes the square coverage pattern. (a) Planned path. (b) Executed path.

waypoints. Communication with the quadrotors was performed in real-time through a Xbee connection, and a base system that controls the location of every robot via GPS coordinates was developed. The velocity of the robots was set to 1.5 m/s.

Two types of experiments were executed:

1. *Same-team cooperation*: When a team has more than one robot, the proposed FSM must be used (Fig. 9) to synchronize the paths between the robots within an hexagon.
2. *Different-team cooperation*: When there is more than one team, the robots must cover different sub-regions simultaneously.

Two robots were used in the experiment with same-team cooperation, one using the lawnmower pattern and another using the square pattern, as shown in Fig. 24b. The results of this experiment can be seen in Fig. 24. The results are promising, showing that the FSM approach allows several aerial robots to perform simultaneous tasks and that the use of different heights to avoid collision is a reasonable approach.

In these experiments, collisions were avoided by defining a different height for every robot in a team. In this case, the height difference was 1.5 m between the robots.

The planned path for the experiment with two single-robot teams is shown in Fig. 25a. The performed path can be seen in Fig. 25b. We verified that the proposed methodology can be used with different sub-regions with the possibility of extending the method to support more area than one battery charge will allow.

The real-world experiments demonstrate that the influence of the environment on small aircraft decreases the performance of the robots. Depending on the available aircraft and the controller used, it is important to fly in conditions with less environmental disturbance. In our case, we used commercial,

(a) Planned path. (b) Executed path.

Fig. 25. Real experiment 2: Cooperation between two different teams. Each robot has to perform a lawnmower path inside its cluster. (a) Planned path. (b) Executed path.

off-the-shelf autopilot software without any tuning of the parameters, which is reflected in the path following performance.

5. Conclusions and Future Work

5.1. Conclusions

In this work, we proposed a methodology for planning complete and cooperative region coverage paths for multiple autonomous aerial robots. We approached the coverage problem using polygonal segmentation based on regular hexagons. The internal paths and the visit order of the paths are optimized depending on the type of survey.

Simulated experiments were executed to quantitatively evaluate the proposed method. As expected, the coverage time decreases with the use of multiple vehicles. The size of the hexagons also has a direct influence on the survey time. Experiments showed that larger hexagons have a fewer inter-hexagon paths, thus, decreasing the survey time. The distance between coverage lines also has a direct impact on the survey time: when the h distance is smaller, the coverage time increases due to the increased number of turns and coverage lines.

Although the proposed methodology exhibited an increased survey time when compared to other cooperative approaches, such as that proposed in Maza *et al.*,[32] it offers several advantages. In simulated sensing experiments, the proposed method generated a map more similar to the ground truth than that produced by the method introduced in Maza *et al.* As this method allows for equally separated sub-regions, cooperation between different coverage patterns and the possibility of different angle and height configurations, it can generate higher quality data in several types of geophysical surveys. The proposed method also allows small robots with very limited autonomy to fly and cover larger areas by returning home and changing batteries.

In real experiments, small quadrotors with a few minutes of flight time were used to recreate the simulated paths. It was demonstrated that the generated routes could be performed by real robots despite their noisy sensors and the influence of the environment.

5.2. Future work

Though this work can be implemented in real robots in its current state, it can also be extended to further decrease the coverage time and to support heterogeneous teams of robots.

Finding the best configuration or balance between the size of the hexagons and robot autonomy will considerably decrease the survey time. As observed in Fig. 14, there is a stabilization in the coverage time, which must be found to decrease extra coverage time. Choosing the correct parameters for the algorithm, such as the h distance and the hexagon radius, could also be automatized using machine learning or optimization techniques.

Future research directions include the following: (i) the development of robust techniques to detect robot failures and reallocate tasks between the remaining robots; (ii) updating the coverage path in real time using active sensing techniques, (iii) the use of more precise GPS systems, such as DGPS,

and (iv) improve the route verification algorithm including more parameters such as hovering time, real consumption of the aircraft and weather information.

Acknowledgments
This work was carried out with the support of Instituto Tecnológico Vale.

References

1. S. Thrun *et al.*, "Robotic Mapping: A Survey," **In**: *Exploring Artificial Intelligence in the New Millennium* (G. Lakemeyer and B. Nebel, eds.) (Morgan Kaufmann, San Mateo, CA, 2002) pp. 1–35.
2. D. W. Gage, "Command control for many-robot systems," *Unmanned Syst. Mag.* **10**(4), 28–34 (1992).
3. D. G. Macharet, H. I. A. Perez-Imaz, P. A. F. Rezeck, G. A. Potje, L. C. C. Benyosef, A. Wiermann, G. M. Freitas, L. G. U. Garcia and M. F. M. Campos, "Autonomous aeromagnetic surveys using a fluxgate magnetometer," *Sensors* **16**(12), 2075 (2016).
4. D. L. Applegate, R. E. Bixby, V. Chvatal and W. J. Cook, *The Traveling Salesman Problem: A Computational Study*, Princeton Series in Applied Mathematics (Princeton University Press, Princeton, NJ, USA, 2007).
5. E. M. Arkin and R. Hassin, "Approximation algorithms for the geometric covering salesman problem," *Discrete Appl. Math.* **55**(3), 197–218 (1994).
6. B. Englot and F. S. Hover, "Sampling-Based Coverage Path Planning for Inspection of Complex Structures," **In**: *22nd International Conference Automated Planning and Scheduling*, Sao Paulo, Brazil (Jun. 25–29, 2012) pp. 29–37, Palo Alto, CA: AAAI Press.
7. W.-P. Chin and S. Ntafos, "Shortest watchman routes in simple polygons," *Discrete & Computational Geom.* **6**(1), 9–31 (1991).
8. Y. Guo and M. Balakrishnan, "Complete Coverage Control for Nonholonomic Mobile Robots in Dynamic Environments," *Proceedings of the 2006 IEEE International Conference on Robotics and Automation ICRA* (2006) pp. 1704–1709.
9. Z. L. Cao, Y. Huang and E. L. Hall, "Region filling operations with random obstacle avoidance for mobile robots," *J. Robot. Syst.* **5**(2), 87–102 (1988).
10. H. Choset, "Coverage for robotics–A survey of recent results," *Ann. Math. Artif. Intell.* **31**(1–4), 113–126 (2001).
11. E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robot. and Auton. Syst.* **61**(12), 1258–1276 (2013).
12. V. J. Lumelsky, S. Mukhopadhyay and S. Kang, "Dynamic path planning in sensor-based terrain acquisition," *IEEE Trans. Robot. Autom.* **6**(4), 462–472 (1990).
13. E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar and D. Hull, "Morse decompositions for coverage tasks," *Int. J. Robot. Res.* **21**(4), 331–344 (2002).
14. E. Peless, S. Abramson, R. Friedman and I. Peleg, "Area coverage with an autonomous robot," (2003). US Patent 6,615,108.
15. W. H. Huang, "Optimal Line-Sweep-Based Decompositions for Coverage Algorithms," *Proceedings of the 2001 IEEE International Conference on Robotics and Automation ICRA*, vol. 1 (2001) pp. 27–32.
16. P. A. Jimenez, B. Shirinzadeh, A. Nicholson and G. Alici, "Optimal Area Covering Using Genetic Algorithms," *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (2007) pp. 1–5.
17. G. E. Jan, C. Luo, L.-P. Hung and S.-T. Shih, "A Computationally Efficient Complete Area Coverage Algorithm for Intelligent Mobile Robot Navigation," *Proceedings of the International Joint Conference on Neural Networks IJCNN* (2014) pp. 961–966.
18. A. Xu, C. Viriyasuthee and I. Rekleitis, "Optimal Complete Terrain Coverage Using An Unmanned Aerial Vehicle," *Proceedings of the IEEE International Conference on Robotics and Automation ICRA* (2011) pp. 2513–2519.
19. H. Choset and P. Pignon, "Coverage Path Planning: The Boustrophedon Cellular Decomposition," **In**: *International Conference on Field and Service Robotics*, Canberra, Australia (Springer, 1998) pp. 203–209.
20. L. Lin and M. A. Goodrich, "Hierarchical heuristic search using a Gaussian mixture model for UAV coverage planning," *IEEE Trans. Cybern.* **44**(12), 2532–2544 (2014).
21. R. J. Meuth, J. L. Vian, E. W. Saad and D. C. Wunsch, "Adaptive multi-vehicle area coverage optimization system and method," (2012). US Patent 8,260,485.
22. A. Carneiro and M. Amorim, "Coverage strategy for periodic readings in robotic-assisted monitoring systems," *Ad Hoc Netw.* **11**(7), 1907–1918 (2013).
23. H. Sagan, *Space-Filling Curves*, vol. 18 (Springer-Verlag, New York, NY, 1994).
24. I. Rekleitis, A. P. New, E. S. Rankin and H. Choset, "Efficient Boustrophedon multi-robot coverage: an algorithmic approach," *Ann. Math. Artif. Intell.* **52**(2–4), 109–142 (2008).
25. Y. Gabriely and E. Rimon, "Spanning-tree based coverage of continuous areas by a mobile robot," *Ann. Math. Artif. Intell.* **31**(1–4), 77–98 (2001).
26. N. Hazon, F. Mieli and G. A. Kaminka, "Towards Robust On-Line Multi-Robot Coverage," *Proceedings of the 2006 IEEE International Conference on Robotics and Automation ICRA* (2006) pp. 1710–1715.

27. N. Agmon, N. Hazon and G. A. Kaminka, "Constructing Spanning Trees for Efficient Multi-Robot Coverage," *Proceedings of the 2006 IEEE International Conference on Robotics and Automation ICRA* (2006) pp. 1698–1703.
28. A. Breitenmoser, M. Schwager, J.-C. Metzger, R. Siegwart and D. Rus, "Voronoi Coverage of Non-Convex Environments with a Group of Networked Robots," *Proceedings of the IEEE International Conference on Robotics and Automation ICRA* (2010) pp. 4982–4989.
29. C. Luo, S. X. Yang and D. A. Stacey, "Real-Time Path Planning with Deadlock Avoidance of Multiple Cleaning Robots," *Proceedings of the IEEE International Conference on Robotics and Automation ICRA*, vol. 3 (2003) pp. 4080–4085.
30. I. A. Wagner, Y. Altshuler, V. Yanovski and A. M. Bruckstein, "Cooperative cleaners: A study in ant robotics," *Int. J. Robot. Res.* **27**(1), 127–151 (2008).
31. P. Sujit, S. Saripalli and J. B. Sousa, "Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicless," *IEEE Control Syst.* **34**(1), 42–59 (2014).
32. I. Maza and A. Ollero, "Multiple UAV Cooperative Searching Operation Using Polygon Area Decomposition and Efficient Coverage Algorithms," **In**: *Distributed Autonomous Robotic Systems,* vol. 6 (Springer, 2007) pp. 221–230 (Chapter).
33. E. Santamaria, F. Segor and I. Tchouchenkov, "Rapid Aerial Mapping with Multiple Heterogeneous Unmanned Vehicles," **In**: *Proceedings of the 10th International ISCRAM Conference*, Kristiansand, Norway (May 24–27, 2015).
34. Y.-S. Jiao, X.-M. Wang, H. Chen and Y. Li, "Research on the Coverage Path Planning of UAVs for Polygon Areas," *Proceedings of the 5th IEEE Conference on Industrial Electronics and Applications ICIEA* (2010) pp. 1467–1472.
35. G. S. C. Avellar, G. A. S. Pereira, L. C. A. Pimenta and P. Iscold, "Multi-UAV routing for area coverage and remote sensing with minimum time," *Sensors* **15**(11), 27783 (2015).
36. P. Toth and D. Vigo, eds., *The Vehicle Routing Problem* (Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001).
37. C. Di Franco and G. Buttazzo, "Coverage path planning for UAVs photogrammetry with energy and resolution constraints," *J. Intell. Robot. Syst.* **83**(3–4), 445–462 (2016).
38. R. Mannadiar and I. Rekleitis, "Optimal Coverage of a Known Arbitrary Environment," *Proceedings of the IEEE International Conference on Robotics and Automation ICRA* (2010) pp. 5525–5530.
39. K. Easton and J. Burdick, "A Coverage Algorithm for Multi-Robot Boundary Inspection," *Proceedings of the 2005 IEEE International Conference on Robotics and Automation ICRA* (2005) pp. 727–734.
40. G. S. C. de Avellar, Navegação de Veículos Aéreos Não Tripulados Para Cobertura de Áreas com Minimização de Tempo *Master's Thesis*, Universidade Federal de Minas Gerais (UFMG), CDU: 621.3(043) (2014).
41. X. Zheng and S. Koenig, "Robot Coverage of Terrain with Non-Uniform Traversability," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS* (2007) pp. 3757–3764.
42. C. Reeves, "Aeromagnetic surveys: Principles, practice and interpretation," *Course Unit i* **50**, 33–99 (2005).
43. R. Kershner, "The number of circles covering a set," *Am. J. Math.* **61**(3), 665–671 (1939).
44. M. E. Snyder, Foundations of Coverage Algorithms in Autonomic Mobile Sensor Networks *Ph.D. Thesis* (Missouri University of Science and Technology, 2014).
45. K. C. Clarke, "Criteria and Measures for the Comparison of Global Geocoding Systems," **In**: *Discrete Global Grids: A Web Book.* (University of California, Santa Barbara, 2002). [http://www.ncgia.ucsb.edu/globalgrids-book].
46. T. Xiaochong, B. Jin, J. Song and Z. Yongsheng, "The Subdivision of Partial Geo-Grid Based on Global Geo-Grid Frame," *Proceedings of the ISPRS Workshop on Updating Geo-spatial Databases with Imagery & The 5th ISPRS Workshop on DMGISs,* Urumchi: Xinjiang, China (2007) pp. 229–235.
47. S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inform. Theory* **28**(2), 129–137 (1982).
48. H. I. A. Perez-imaz, P. A. F. Rezeck, D. G. Macharet and M. F. M. Campos, "Multi-Robot 3d Coverage Path Planning for First Responders Teams," *Proceedings of the IEEE International Conference on Automation Science and Engineering CASE* (2016) pp. 1374–1379.
49. S. Hert and V. Lumelsky, "Polygon area decomposition for multiple-robot workspace division," *Int. J. Comput. Geom. Appl.* **8**(4), 437–466 (1998).
50. A. Bhattacharyya, "On a measure of divergence between two statistical population defined by their population distributions," *Bull. Calcutta Math. Soc.* **35**, 99–109 (1943).
51. T. Kailath, "The divergence and Bhattacharyya distance measures in signal selection," *IEEE Trans. Commun. Technol.* **15**(1), 52–60 (1967).