Rapid Aerial Mapping with Multiple Heterogeneous Unmanned Vehicles

Eduard Santamaria, Florian Segor, Igor Tchouchenkov, and Rainer Schönbein
IAS – Interoperabilität und Assistenzsysteme
Fraunhofer IOSB
Karlsruhe, Germany
{eduard.santamaria, florian.segor, igor.tchouchenkov, rainer.schoenbein}@iosb.fraunhofer.de

***Abstract*—One focus of research at Frauhofer IOSB is the utilization of unmanned aerial vehicles for data acquisition. Past efforts have lead to the development of a hardware and software system able to rapidly generate a complete and up-to-date aerial image by combining several single high resolution pictures taken by multiple unmanned aerial vehicles. However, the path planning component of the system was not designed to support no-fly zones inside the area of interest. Besides, the system assumed that all vehicles would have equal flight range and the same sensor footprint. In this paper, we address these limitations and present a new complete coverage path planning algorithm with support for no-fly zones inside the area of interest. The proposed method is suitable for non-convex areas, possibly with holes, to be covered by one or more maneuverable systems such as multi-rotor aircraft. Range and sensor footprint of the aircraft may differ.**

Keywords - aerial situation image; unmanned aerial vehicles; complete coverage path planning

I. Introduction

The technical advance in the development of miniature unmanned aerial vehicles (UAVs) in the last decade has made unmanned aerial systems more capable and affordable. Hence, nowadays, civil applications are not only conceivable but already reality. Due to the current rate of development and the varied application possibilities of miniaturized unmanned aircraft, an exponential increase in the usage of these systems can be expected.

In this article, our efforts towards the development of a system for rapidly generating high-resolution aerial imagery using UAVs are described. The first contribution of the article is a new algorithm that efficiently generates a flight plan to completely cover a given area of interest. The area of interest may have complex contours and may contain zones that must not be over-flown by the vehicle. The ability to avoid certain parts will be necessary if obstacles are present or access is forbidden. It will also be useful to prevent irrelevant zones from being inspected. The path planning algorithm and initial results were presented in [1].

The second contribution is a methodology for partitioning the area of interest and distribute the workload of the mission among several aircraft with different ranges and sensor footprints (some initial steps were reported in [2] as a work in progress).

The research presented in this paper builds upon Fraunhofer IOSB's continued efforts to enable rescue and emergency forces to take advantage of UAVs' capabilities in an easy and intuitive way. Previous results include the development of an inspection system for generating high resolution up-to-date aerial images [3]. The system uses the payload capacity of one or several UAVs to scan a defined area with high-resolution image sensors and generates an image mosaic from the accumulated single frames.

Such image acquisition capabilities are integrated into AMFIS, a generic mobile ground control station for monitoring and controlling operation of multiple sensors and sensor carriers [4]. AMFIS features different working positions that enable the users to directly steer the vehicles, and to view their location and other information on a moving map. The unmanned vehicles can also be semi-autonomously controlled through point and click commands. The ground control station receives and displays the video streams and other data coming from the deployed sensors.

With the algorithms presented in this paper the system is now able to deal with no-fly zones and can handle different sensor footprints in a multi-vehicle scenario. Additionally, when compared to the algorithm originally in place, the new planning method yields more efficient paths. Three metrics are used to do this comparison: total flown distance, number of turns and number of "jumps" between non-adjacent cells. While the total distance travelled by the UAV is similar in both cases, with the new method, the number of necessary turns is significantly reduced. The number of jumps is slightly incremented, but with no impact on the total travelled distance.

The paper is organized as follows: related work is discussed in Section II. Section III briefly presents potential application scenarios. Section IV details the proposed path planning algorithm and its results are compared with those obtained with the algorithm previously in use. In Section V the proposed approach for the deployment of multiple heterogeneous vehicles is described. In Section VI, the results obtained with real flight tests after integrating the new algorithms into AMFIS are reported. Section VII provides a conclusion and lines of future work.

II. Related work

A taxonomy proposed by Choset divides coverage path planning algorithms into heuristic based algorithms and algorithms based on a cellular decomposition. The latter can rely on an exact, a semi-approximate or an approximate decomposition [5]. Heuristic based algorithms combine heuristics and randomness to drive the exploration process. These methods, that do not require expensive sensors and do not consume much computational resources, can provide a good ratio between cost and performance; however, parts of the area of interest may remain unvisited. Therefore, complete coverage is not guaranteed. Most of complete coverage path planning algorithms implicitly or explicitly adopt cellular decomposition to achieve completeness.

An exact cellular decomposition is a set of non-intersecting regions, each termed a cell, whose union fills the target environment. Typically, the robot can cover each cell using some kind of motion pattern, e.g., back-and-forth movements, and the path planning algorithm decides the order to visit the cells [6][7][8][9].

Semi-approximate algorithms rely on a partial discretization of space where cells are fixed in width but their top and bottom can have any shape [10][11]. The robot moves along these columns and different parts of a complex area are recursively explored in order to achieve completeness.

An approximate cellular decomposition generates a grid based representation of the area of interest. All cells have the same size and shape and their union approximates the target region. Coverage is complete when the robot visits each cell in the grid. The cell size typically depends on the footprint of the robot. This approach fits very well to our application, where the goal is to generate a complete aerial image of an area by combining aerial photos taken at different points. Many algorithms have been developed that fall into this category. Some of them are referenced in the next paragraphs.

Different authors have developed coverage path planning methods based on spanning trees [12][13][14]. These methods generate a continuous path around the spanning tree. This is a very good property for continuous surveillance operations. The nature of the algorithm requires that, if the cell size derived from the camera footprint is D, the area shall be decomposed into cells of size 4D. Different implementations to generate the spanning trees differ regarding computational complexity and quality of the generated results.

Zelinsky et al. proposed a complete coverage path generation method based on distance transforms [15]. With distance transforms each cell is assigned a value that represents the distance to the goal. These values can be used to find the shortest path from a starting point to the goal. Extensions to the distance transform path planning methodology can be used to generate a complete coverage path. One of the extensions proposed by Zelinsky et al. generates many unnecessary turns. An improved version creates a path that tends to follow the contour of the area. Recently, a distance transform based method has been used by Barrientos et al. to obtain optimal paths in the context of agricultural applications [16]. Their algorithm uses a costly backtracking algorithm to compute all coverage path candidates.

The method proposed by Carvalho et al. makes use of several interesting patterns to generate the path [17]. However, the scanning always takes place in the same direction, which would be a disadvantage in some circumstances, for instance, when a L-shaped area needs to be covered.

An approach developed by Choi et al. creates a path that follows a spiral pattern [18]. Because the algorithm has a tendency to propagate the contour corners towards the inner part of the area of interest, such kind of pattern will not be very efficient in terms of the number of turns when the cell grid has contours with many corners.

The grid that represents the area is used in the method proposed by Kang et al. to create a number of rectangular subareas by grouping cells [19]. Then, one of several patterns is applied to each subarea. We believe that this method can work well when the alignment of boundary cells tends to form rectilinear sides. A number of cells may be revisited when moving from the end of one pattern to the start of the next one.

Segor et al. describe a system that is able to use several small UAVs to efficiently obtain a complete aerial image [3]. Each UAV is allocated one subarea to scan. To cover each subarea two candidate paths are generated: one that makes progress by scanning the area column by column, and another one that does the same row by row. The one that yields better results is chosen. A drawback of such approach is that it does not adapt well to situations where a combination of different scan directions would be more beneficial. Besides, the original implementation was not designed to support no-fly zones inside the area of interest. This work, developed at Fraunhofer IOSB, was the starting point of the research presented in this article.

As shown in Section IV, the new path planning algorithm that we are proposing, significantly improves the results of the previous method used in the AMFIS system. The generation of more efficient flight paths does not come at the expense of a significant increase in computation time.

Regarding the use of multiple aircraft for aerial sensing applications, several projects exist that propose solutions for such scenario. In the SkyObserver project a geometric optimal placement algorithm is used to distribute the unmanned aircraft inside a given area [20]. In the AirShield and AVIGLE projects similar techniques are used [20][21][22]. However, in this case a more holistic approach that simultaneously considers spatial coverage optimization, the mobility strategy to explore the area, and communication awareness is followed.

In the COMETS project, the area of interest is first decomposed into non-intersecting regions taking into account UAV's relative capabilities and initial locations [6]. Afterwards, the resulting areas are assigned among the UAVs that will cover them using a zigzag pattern.

Finally, the problem of creating an aerial image of a given area is also addressed in the cDrones project [24][25]. Like in our case, the proposed solution is based on a grid approximation of the area of interest. A genetic algorithm is used to compute the flight path. While the system is able to use multiple aircraft, the method for distributing the workload is not described.

Decentralized planning methods as in SkyObserver, AirShield and AVIGLE try to maintain connectivity between

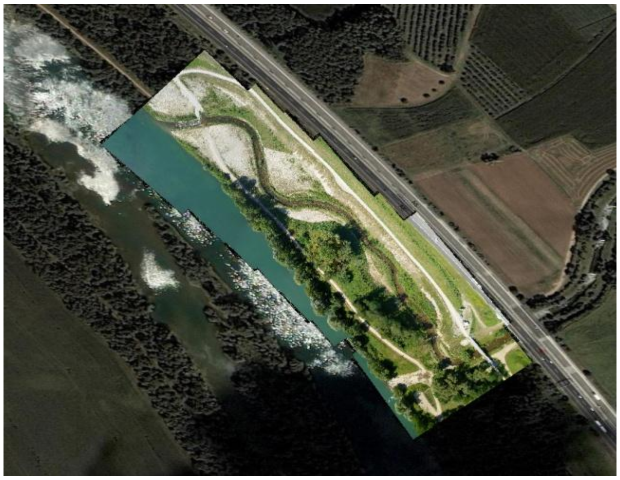


Figure 1. Photomosaic of Biotope at the Rhein River.

nodes. As a result, constraints imposed on the trajectories make complete coverage harder to achieve. We propose a practical approach where each aircraft executes a pre-assigned path. With the exception of cDrones, the referenced projects do not provide support for no-fly zones. Finally, most approaches do not consider the possibility to perform the mission with heterogeneous platforms.

III. APPLICATION SCENARIOS

The security feeling of our society has significantly changed during the past years. Besides the risks arising from natural disasters, there are dangers in connection with criminal or terroristic activities, traffic accidents or accidents in industrial environments. Especially in the civil domain in case of big incidents there is a need for a better data basis to support the rescue forces in decision making.

The search for buried people after building collapses or the clarification of fires at big factories or chemical plants are possible scenarios addressed by our system.

Many of these events have very similar characteristics. They cannot be foreseen in their temporal and local occurrence so that situational in situ security or supervision systems are not present. The data basis for decision making is rather thin and therefore the present situation is very unclear to the rescue forces at the beginning of a mission. Exactly in such situations it is extremely important to understand the context as fast as possible to initiate the suitable measures specifically and efficiently.

An up-to-date aerial image can be a valuable additional piece of information to support the briefing and decision making process of the first responders.

Helicopters or supervision airplanes that can supply this information are very expensive or may be unavailable. High-resolution pictures from an earth observation satellite could also be a good solution in many cases. But, under normal circumstances, these systems will not be available in time or they may not be able to deliver good pictures because of clouds or smoke. A small, transportable, fast and easily deployable system that is able to produce results with higher spatial and temporal resolution is proposed to close this gap.

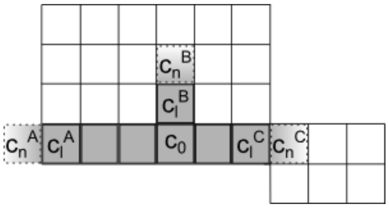


Figure 2. Stride formation starting at cell c0.

The aerial inspection tool described in this paper can provide the lacking information by creating an overview of the site of the incident in a very short time. The application can be used by first responders directly on site with relative ease. The results provide a huge enhancement to the available information.

Many other applications are also possible: support to fire-fighting work, clarification of debris and the surroundings after building collapses, search for buried or injured people, inspection of large objects, or for documentation and perpetuation purposes, as for example, of protected areas and biotopes (see Fig. 1).

IV. PATH PLANNING

Once the area of interest has been identified and its grid approximation has been computed, a flight path will be generated for each unmanned vehicle participating in the mission. Two steps are required during this process: first, the area is partitioned according to the capabilities of each aircraft. Next, a flight path to cover each of the subareas is computed. These flight paths contain sequences of waypoints where pictures need to be taken in order to completely cover the area.

In this section, the path planning algorithm is described in detail. The algorithm will be applied to the whole grid if there is a single UAV, or to each one of the subareas in a multi-UAV scenario. Also in this section, the results of its application to several different areas are presented. These results are compared to the ones obtained with the previous algorithm used in AMFIS.

A. Path Planning Algorithm

The path planning algorithm tries to generate the longest possible straight flight segments. We call each one of these segments a stride. A stride is defined as a sequence of consecutive adjacent cells without turns. To compute a stride of max length the following rules are used:

- The stride starts at the current cell.
- The direction of the stride is determined by its starting point and the neighbor under consideration.
- A stride contains no turns.
- A number of conditions, explained below, determine where the stride ends.

We define $L(c)$ as the number of visited neighbor cells and area limits located orthogonally to the stride direction at cell c. Therefore, assuming a stride direction from left to right, the value of $L(c)$ will be 2 if the cells located immediately on top and below c are marked as visited or fall outside the area of interest. Conversely, $L(c)$ will be 0 if both

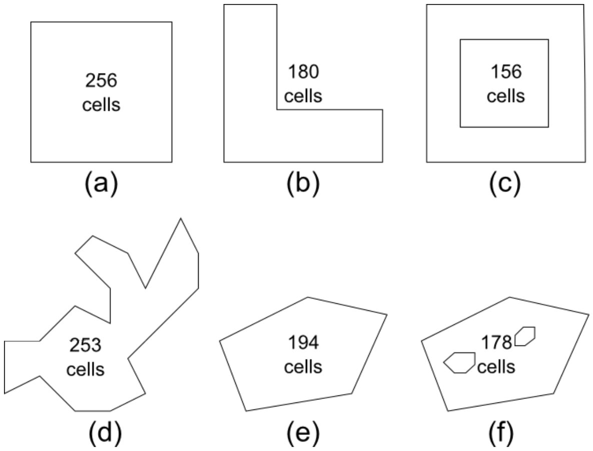


Figure 3. Contours, with their number of cells, used in the tests.

cells are inside the area of interest and still need to be covered.

Being c_0 the first cell of the stride, c_l the current last cell of the stride, and c_n a potential next cell in the stride direction, addition of c_n to the stride is subject to the following conditions:

- C_n is not added to the generated path if it is already marked as visited or if it falls outside the boundaries of the area.
- If the previous condition does not hold, and $L(c_l) = 2$, c_n is always added to the stride, because it is the only unvisited cell that can be reached from c_l.
- If $L(c_l) \neq 2$, any of the following conditions will prevent addition of c_n to the stride: (1) $L(c_n) = 0$; (2) $L(c_n) \neq L(c_0)$; or (3) $L(c_n) = 1$ but the limit at c_n is positioned opposite to the limit found at c_0. The purpose of condition (1) is to stop when the path is not following an area limit or a "wall" formed by cells already in the path. Conditions (2) and (3) dictate that stride formation will also stop when the limits of c_n differ from the limits of c_0.

To clarify the previous points, stride formation is illustrated in Fig. 2. Starting at c_0, there are three alternative strides that can be selected. Stride A (left of c_0) ends when an area limit is reached. Stride B (up) ends because $L(c_n^B) = 0$. Note that it would be possible to extend the stride because there are unvisited cells in that direction, but this would eventually lead to the partition of the area into two disconnected regions. Finally, in stride C (right of c_0), both c_0 and c_n^C have an area limit located orthogonally to the stride direction. Therefore $L(c_0) = L(c_n^C) = 1$, however, since the area limits of these cells are located at opposite sides (below in the case of c_0 and above for c_n^C), c_n^C is not added to the stride.

The algorithm for generating the complete coverage path works as follows:

1. Set the current cell to the initial cell.
2. Find all unvisited neighbor cells of the current cell (between 0 and 4 cells are returned).
3. Generate the longest possible stride in the direction of each unvisited neighbor cell.
4. Select the longest stride.

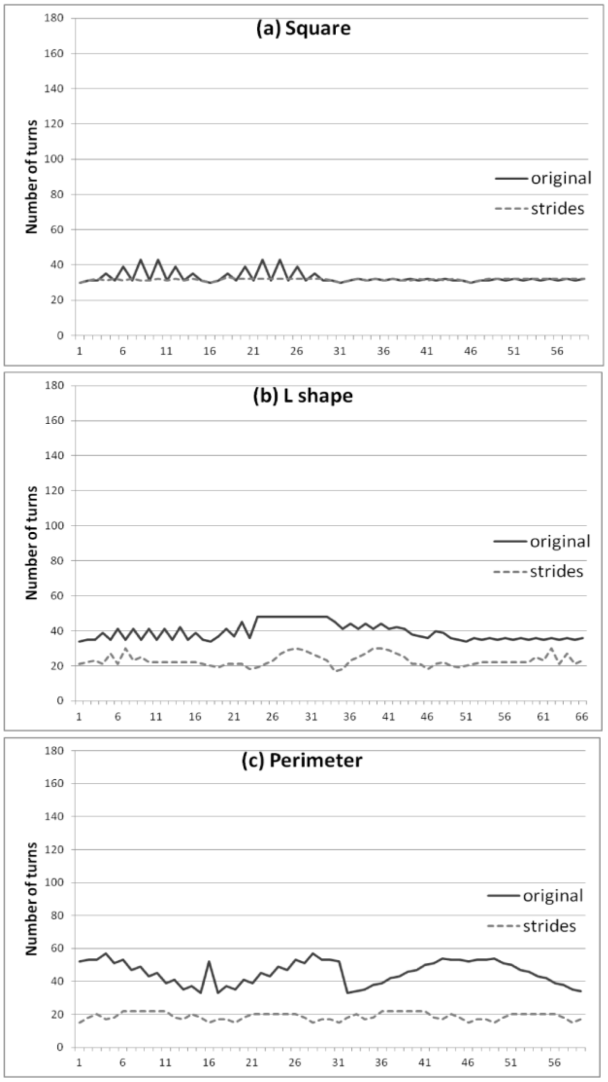


Figure 4. Number of turns starting at each contour cell of a, b and c.

5. Add all cells of the stride to the path and mark them as visited.
6. Set the current cell to the last cell of the stride.
7. Repeat starting at point 2 until all cells have been visited.

In the example presented in Fig. 2, stride A, with four cells, would be selected over the B and C alternatives, which respectively have two and three cells.

When all alternative strides have length two (c_0 plus a neighbor cell), some heuristics are used to perform the stride selection. These heuristics prioritize the selection of (1) the neighbor cells with a higher value for $L(c_n^C)$, (2) the ones located in the contour of the area, and (3) the ones that lead to a longer stride in the next step. These heuristics have been chosen after extensive testing with different area shapes. Heuristics (1) and (2) promote the selection of a path that moves alongside other visited cells or the contour of the area.

Sometimes it is inevitable to partition the area, creating two, or more, subareas of disconnected unvisited cells. When

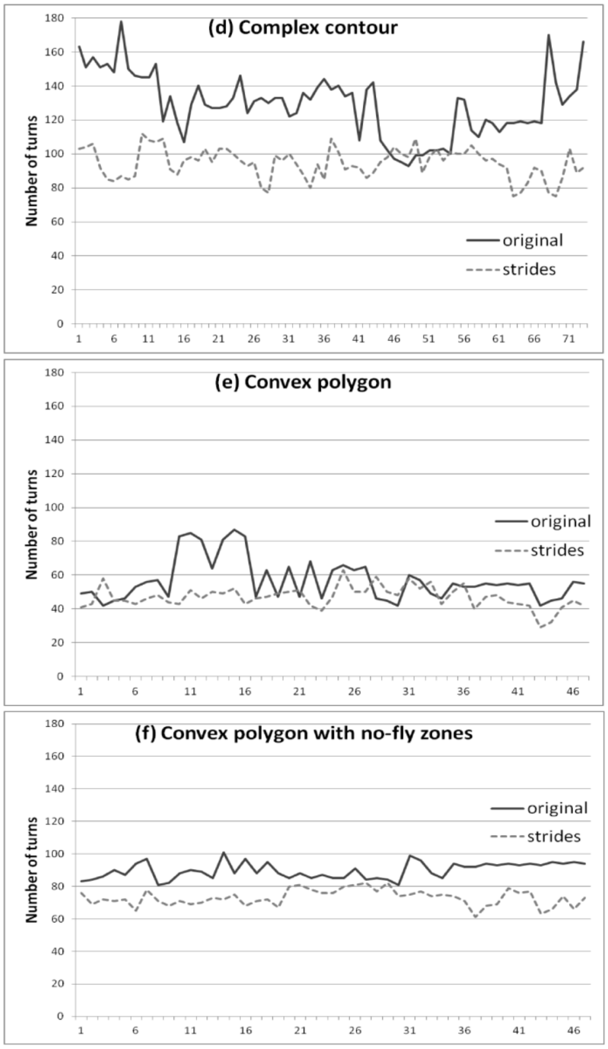


Figure 5. Number of turns starting at each contour cell of d, e, and f.

such situation occurs, the algorithm chooses to visit the cells of the smallest subarea first.

Another situation that needs to be addressed occurs when a dead end is reached, i.e., when there are no unvisited valid cells next to the current cell. In this case, our solution computes paths between the current cell and each one of the unvisited cells adjacent to cells already in the flight path. These paths are computed using distance transform path planning as described by Zelinsky et al. in [15]. After all the alternatives have been computed, the shortest path is selected and its cells are added to the complete coverage path. This same method can be used to compute a path to the landing point.

The distance wave propagation used to compute the shortest path (see [15]), is also used to determine if the area has been partitioned. The main idea behind this procedure is that, after performing the propagation of the distance wave, if any cells remain that have not been assigned a value they must be located in a different partition.

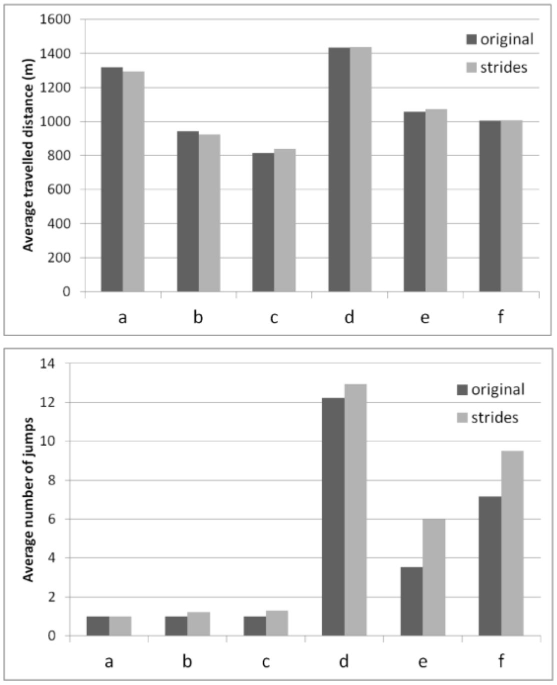


Figure 6. Average travelled distance (top) and average number of jumps (bottom).

One final step performs some clean-up on the generated path. Those sequences of revisited cells that connect two adjacent cells are removed from the path. In this way, the UAV transitions directly from a given cell to one of its neighbors and unnecessary repeats are prevented. Since the number of times each cell is visited is known, the implementation of the clean-up step is straightforward. For each cell C_i of the flight plan, the algorithm checks if the rest of the path contains a cell *Cj* that is adjacent to C_i. If all cells in between are visited more than once, they can safely be removed.

B. *Results*

The proposed algorithm has been tested with areas of different shape. In this section, the results obtained with the areas showed in Fig. 3 are presented. The new algorithm has been compared with the original algorithm of the photo flight tool. The metrics used are the number of turns, the travelled distance, and the number of jumps, which are the transitions between non-adjacent cells. Some considerations need to be taken into account to analyze the results:

1. In a situation where all neighbor cells next to the current position have been visited, but coverage is not complete, the original algorithm didn't provide a safe path to fly from the current position to the next free cell. For this reason the number of jumps between non-adjacent cells is compared instead of the number of revisited cells.
2. The original algorithm was not designed to handle no-fly zones inside the area of interest. Nevertheless, if such an area is provided as input, it is able to generate a complete coverage path.

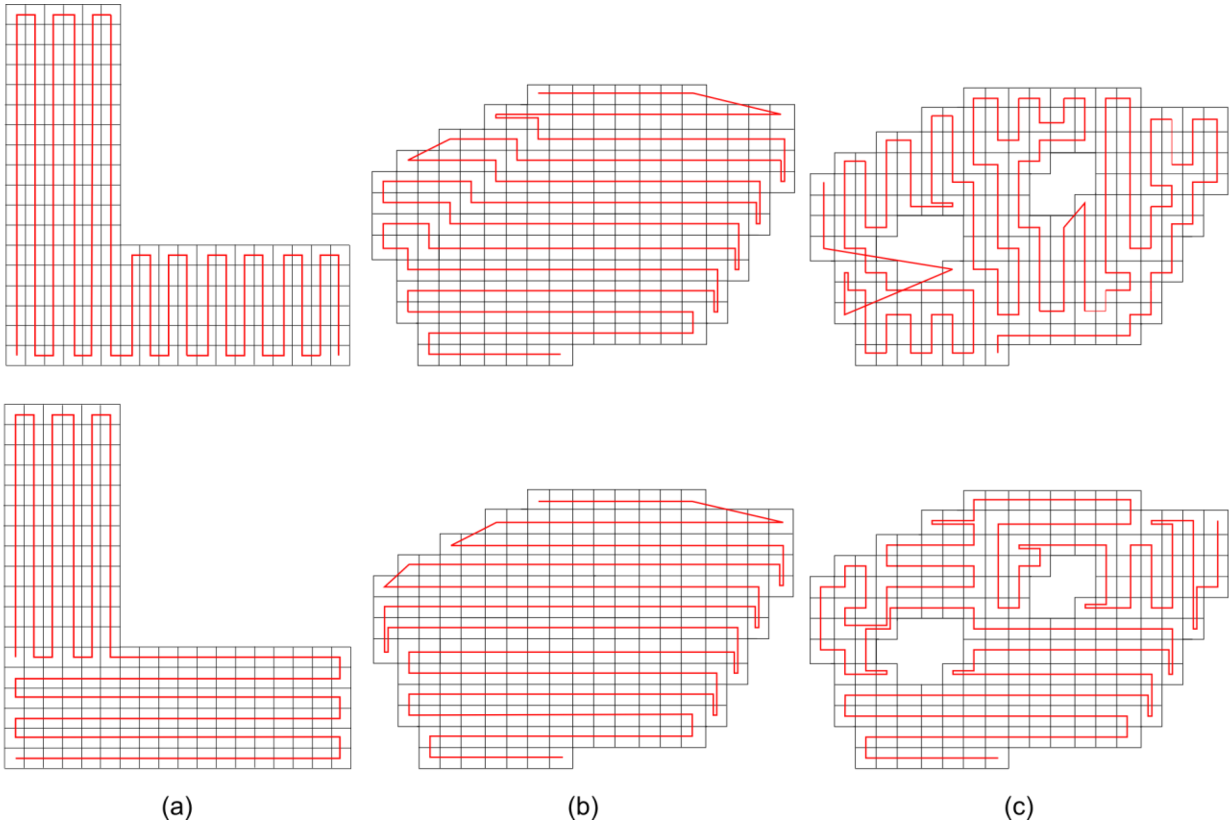


Figure 7. Complete coverage path generated for some example contours with both original (top) and strides based algorithm (bottom). In figure c the transitions between non-adjacent cells are also included.

3. The computed distance corresponds to the path length plus the distance between the last and the first cell of the path.

In Figs. 4 and 5, the number of turns obtained running both the original and the new strides based algorithms are displayed. The algorithms have been run starting at each contour cell of the example areas in Fig. 3. As it can be observed, the new algorithm provides better results with all tested areas.

Fig. 6 displays the average travelled distance (top) and average number of jumps (bottom) obtained, with both algorithms, for each area. It can be seen that, although the number of jumps is slightly increased in some cases, this increase has almost no impact on the average travelled distance.

To understand the reasons that lead to an improvement in the number of turns, we now compare the complete paths of the areas b and e of Fig. 3. In Fig. 7a, it can be seen that one source of improvement is the ability of the new algorithm to use different scan directions in different parts of the area of interest. Another source of improvement (see Fig. 7b), comes from the fact that the new algorithm is better at getting rid of contour corners, avoiding its propagation into the inner parts of the area (see Fig. 7b).

Finally, in Fig. 7c (bottom), the complete path generated for a complex area with no-fly zones is shown. When there are no unvisited neighbors, a safe path to reach the next free cell is computed. Thus, the complete coverage path does not contain jumps between non-adjacent cells. The generated path can be contrasted with a path generated by the original algorithm (top), which was not really designed to cope with holes in the area, and does not provide a mechanism to generate a safe path between non-adjacent cells.

It should be noted that the path planning algorithm always operates on a grid where each cell is a square. However, the actual footprint could also be a rectangle whose sides differ in length. In that case, the size of the square cells will be determined by the longest side of the rectangle. Once the path along the grid has been computed, an additional step is needed to compute the list of waypoints required to take all the pictures according to the actual footprint. The orientation of the sensor carrier will be determined in accordance to the flying direction.

V. PLANNING FOR MULTIPLE HETEROGENEOUS VEHICLES

In a multi-UAV scenario, the application of the path planning algorithm is preceded by a partitioning of the area of interest to distribute the workload of the mission among the available vehicles. In this section, the method used to perform such partitioning is described.

A. Workload Distribution

The partitioning algorithm needs to fulfill a number of requirements. It needs to provide support for using platforms with different capabilities regarding range of the vehicles, speed, sensor size, and sensor resolution. It must be able to

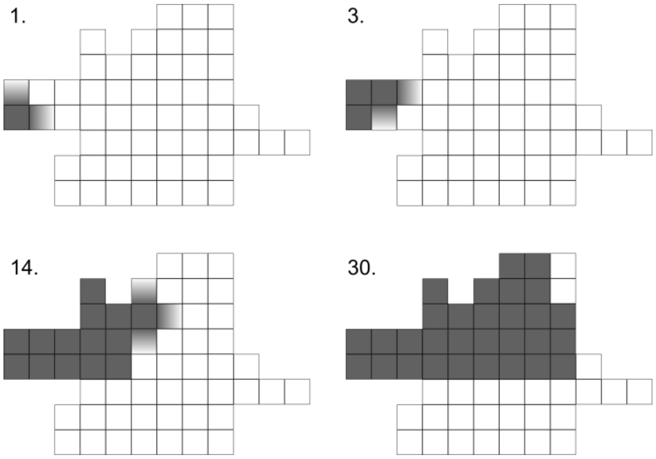


Figure 8. A flood-fill like algorithm is applied to assign cells to a partition.

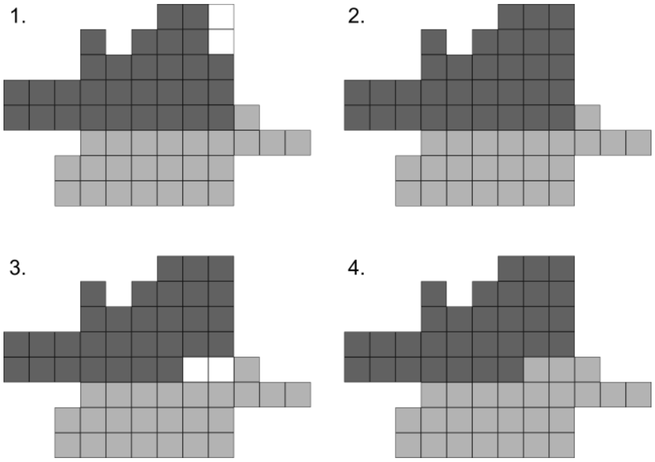


Figure 9. Initial partition may be rearranged to properly allocate all cells.

cope with areas with concave contours, possibly containing holes. Pieces that facilitate the path planning should be generated, i.e., the generation of subareas with unnecessary corners should be avoided. Finally, in order to provide a rapid response in emergency situations, the method for partitioning the area needs to be fast.

To partition the area of interest, an initial method already implemented in the AMFIS ground control station has been extended. As the path planning algorithm, the partitioning method relies on a grid approximation of the area of interest.

The first step consists in computing the percentage of the area of interest that should be covered by each vehicle. However, the actual length of a computed flight path does not only depend on the number of cells that need to be covered. Aspects such as shape of the area, sensor footprint and actual decisions made during the planning process also have an impact on the traveled distance. Since it is not possible to determine the flight length before actually computing the flight plan, the percentage of the area that will be allocated to each aircraft is based on a rough estimation that takes into account the sensor footprint of each aircraft.

The percentage values are used to determine the number of cells of the grid required for each subarea. To perform the partitioning an initial cell is selected and a flood-fill like algorithm is applied to extend the subarea until the desired number of cells is reached. The process, depicted in Fig. 8, prioritizes the selection of adjacent cells with a higher number of neighbor cells in the same subarea. Cells outside the area of interest or in holes are ignored.

Once an initial solution has been computed, remaining cells that have not been allocated (see Fig.9) are assigned to an adjacent area. This step is followed by a redistribution of cells to obtain the desired number of cells in each subarea.

B. Dealing with Multiple Sensor Footprints

If all aircraft involved in the mission have a sensor footprint of the same size, a single grid can be used to partition the area and to plan complete coverage flight paths. In Fig. 10, the results of partitioning a given area to divide the work between three aircraft are displayed (10.a), together with the flight paths generated by the path planning algorithm (10.b).

If the sizes of the sensor footprints differ, using a single grid is no longer possible. In the following paragraphs the method used to deal with multiple footprints is described.

The first step consists in creating a base grid of square cells to partition the area and distribute the work between the different vehicles. Assuming that there are n vehicles available to perform the mission, with w_i and h_i respectively representing the width and the height of the i^{th} vehicle's footprint, the side length s of the grid cells is computed as:

$$s = Min_{i=1}^{n}(Max(w_i, h_i)).$$

It should be noted that, for convenience, we use the word footprint to refer to an area that is actually smaller than the area on the ground surface captured by the sensor. The use of an area smaller than the real footprint is done in purpose to introduce an overlap between the images that will facilitate the stitching process. This overlap will also be helpful to mitigate inaccuracies of the positioning system.

Once the base grid has been created, the partitioning is performed as previously described. The resulting subareas may not be directly used by the path planning algorithm due to the mismatch between the cell size of the grid and the footprint of the different aircraft. A new grid with a cell size proportional to the sensor footprint will be created for each aircraft. The cells that will be covered in each new grid are those that overlap with the subarea assigned to the aircraft in the base grid (see Fig. 11.b).

For efficiency and safety reasons overlaps between areas allocated to different aircrafts should be minimized. The next step of the process, the generation of waypoints, tries to minimize these redundancies.

The main goals of the waypoint generation step are:

1. Convert the paths computed as cells on a grid to actual flight plan waypoints: To minimize the distance between consecutive waypoints, the on-board sensors will be oriented with their longest side orthogonal to the direction of movement. The length of the shortest side is used as the distance between consecutive waypoints.
2. Prevent overlapped zones between subareas from being visited multiple times: If a location has been

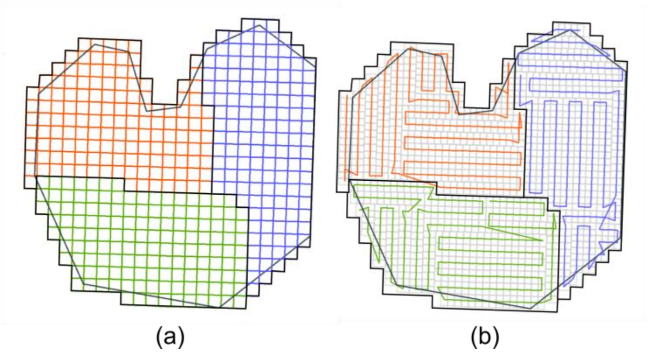


Figure 10. Area partitioning (a) and path planning (b) with same cell size.

photographed by one of the aircraft, waypoints for taking additional pictures of the same place with another aircraft are not generated.

Having multiple aircraft visiting nearby locations has safety implications. To avoid collisions between the unmanned vehicles several approaches are possible. One way to solve this problem would consist in improving the planning phase so that the probability of multiple vehicles being at the same place at the same time is minimized. Another possibility would be to add the necessary sensory and computing capabilities to the vehicles to autonomously detect and deal with possible conflicts. Finally, an effective and practical approach consists in ensuring that the vehicles fly at different heights with enough separation. Since the presented planning method is able to deal with different footprints, this approach can be easily implemented in our system.

VI. EXPERIMENTAL RESULTS

The platform chosen to perform the experiment is the Falcon 8 octocopter from Ascending Technologies. The Falcon 8 comes equipped with an autopilot and a GPS sensor that enable autonomous flight. A camera installed on a stabilized mount can be automatically triggered every time a waypoint is reached. While the high resolution pictures need to be recovered after landing, the system is able to provide a continuous video stream that can be displayed on the ground control station. The Falcon 8 is a lightweight system of 2.2 kg maximum take-off weight that provides up to 20 minutes flight time.

The partitioning and planning algorithms presented in this article have been integrated into the AMFIS system. The mobile ground control station provides tools for designing the mission, supervising its execution and analyzing the obtained results.

With the flight planning tool (partially shown in Fig. 12) the user will perform the following steps:

1. Mark the region of interest on a map.
2. Select the unmanned aircraft that will be used in the mission.
3. Set the flight altitude for each aircraft, or define the desired resolution.
4. Compute a flight path for each aircraft.

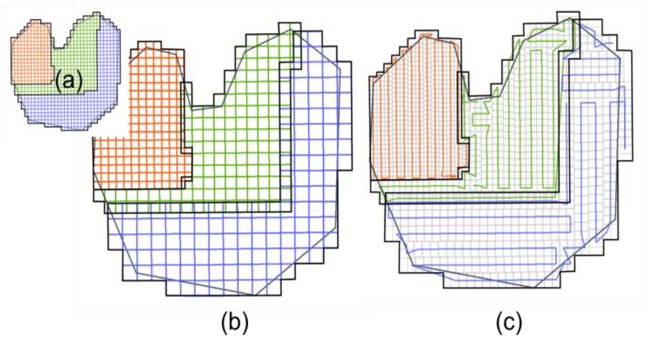


Figure 11. Area partitioning (a+b) and path planning (c) with different cell size.

5. Send the obtained paths to the different vehicles.

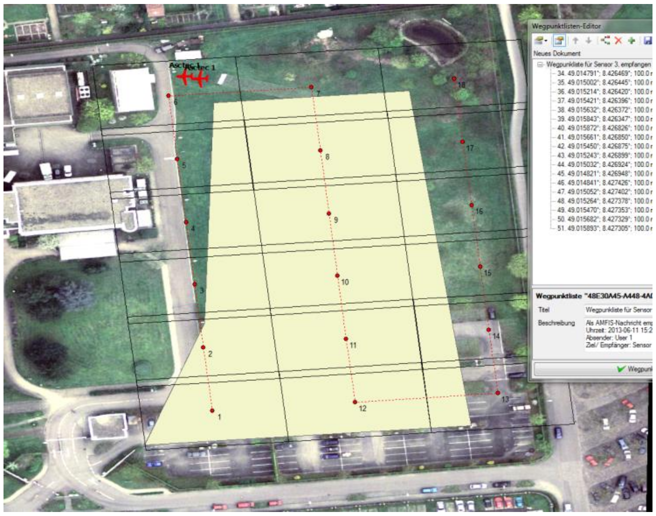
Once the previous steps are completed, the vehicles can be commanded to autonomously perform the mission. After the recovery of the pictures, the same photo flight tool can be used to stitch them together and provide a complete high resolution image of the area.

Our image acquisition system was tested in the Karlsruhe facilities of Fraunhofer IOSB. For safety reasons, the flight altitude was set to 100 meters. At such altitude the footprint of the camera has an approximate size of 120 x 90 square meters. Therefore, with only a few pictures the available area for performing the tests, which approximately measures 17.000 square meters, would have been covered. To force a smaller footprint, the flights were planned at a height below the safe altitude. Once the flight paths had been computed, we took advantage of the editing functions of the ground control station to set the altitude of all waypoints to a safe value. Such scenario results in a big overlap between pictures that facilitates the stitching process, but does not affect the path planning aspect, which was the focus of our interest.

Our tests consisted in three flights. The first flight covered the whole area with a single vehicle. The second and third flights were planned assuming that two vehicles would be flying simultaneously. The flights were planned using different altitudes to prevent collision between the vehicles. The use of different flight altitudes results in different footprint sizes, which our flight planning tool proved to be able to handle. To prevent any risks, the second and third flights were actually performed sequentially using the same vehicle. This also meant that a single emergency pilot would suffice. The number of taken pictures, the horizontal flown distance and the duration of each flight can be found in Table I. Distance and flight duration were measured between the first and the last waypoint.

Table I. Number of pictures, horizontally travelled distance and flight time.

Flight	Pictures	Distance (meters)	Time (mm:ss)
1st	18	420	03:26
2nd	16	320	02:15
3rd	13	240	01:57



Wegpunktlisten-Editor

Neues Dokument

Wegpunktliste für Sensor 3, empfangen
- 34. 49.014791°; 8.426469°; 100.0 r
- 35. 49.015002°; 8.426445°; 100.0 r
- 36. 49.015214°; 8.426420°; 100.0 r
- 37. 49.015421°; 8.426396°; 100.0 r
- 38. 49.015632°; 8.426372°; 100.0 r
- 39. 49.015843°; 8.426347°; 100.0 r
- 40. 49.015872°; 8.426826°; 100.0 r
- 41. 49.015661°; 8.426850°; 100.0 r
- 42. 49.015450°; 8.426875°; 100.0 r
- 43. 49.015243°; 8.426899°; 100.0 r
- 44. 49.015032°; 8.426924°; 100.0 r
- 45. 49.014821°; 8.426948°; 100.0 r
- 46. 49.014841°; 8.427426°; 100.0 r
- 47. 49.015052°; 8.427402°; 100.0 r
- 48. 49.015264°; 8.427378°; 100.0 r
- 49. 49.015470°; 8.427353°; 100.0 r
- 50. 49.015682°; 8.427329°; 100.0 r
- 51. 49.015893°; 8.427305°; 100.0 r

Wegpunktliste "48E30A45-A448-4A

Titel: Wegpunktliste für Sensor

Beschreibung: Als AMFIS-Nachricht emp
Uhrzeit: 2013-06-11 15:2
Absender: User 1
Ziel/ Empfänger: Sensor

✔ Wegpunk

Figure 12. Path planning tool detail.

The results of the experiments were highly satisfactory. We were able to successfully plan the mission with the path planning tool, send the flight plans to the vehicle, share them with other components of the ground control station, and finally execute the mission without incidents. Human intervention was only required to perform the take-off and landing operations because the Falcon 8 platform does not provide support to perform these operations automatically. The aerial image obtained after combining all individual pictures is shown in Fig. 13.

The main issues encountered during the tests relate to the use of a grid approximation to represent the area of interest. Sometimes most of the area covered by boundary cells falls outside the area of interest. While these cells are necessary to guarantee complete coverage, this situation can lead to inefficiencies that are more evident when the number of boundary cells is high in relation to the total number of cells.

Moreover, the presence of cells with large parts of their area lying outside of the region of interest can result in the aerial vehicle flying outside the region's boundaries. This can be a problem if such boundaries have been defined to avoid obstacles. In our tests we wanted to keep a safe distance from the buildings and also avoid crossing the limits of the institute's facilities.

Since the AMFIS ground control station provides support for editing the flight plans, such functions can be used to make manual adjustments and prevent the vehicle from flying over undesired zones. While this could be acceptable in some cases, it is necessary to improve the planning tool to provide a more general solution. We believe that this problem can be tackled with the addition of a new processing step that would take the flight plan as input and generate a modified version were all waypoints would lie inside the area of interest, possibly incrementing the overlap between the pictures.

VII. CONCLUSION AND FUTURE WORK

In this paper, we introduced our most recent work targeted at the development of a system to enable fast generation of aerial image mosaics. We believe that such system can provide highly valuable information in emergency situations. The proposed partitioning and path planning algorithms, which are able to generate efficient solutions in a short time, are core elements of the system. The method for solving the area partitioning problem in the presence of multiple vehicles is able to cope with different sensor footprints. During the path planning a criterion that prioritizes the selection of long straight segments is applied. Such approach results in the generation of flight paths with a reduced number of turns. Fast moving aircraft will particularly benefit from having to perform less turns. Its ability to scan the area in different directions and the fact that it does not rely on pre-defined patterns make the proposed planning algorithm suitable to generate complete coverage paths for complex contours, which may contain holes.

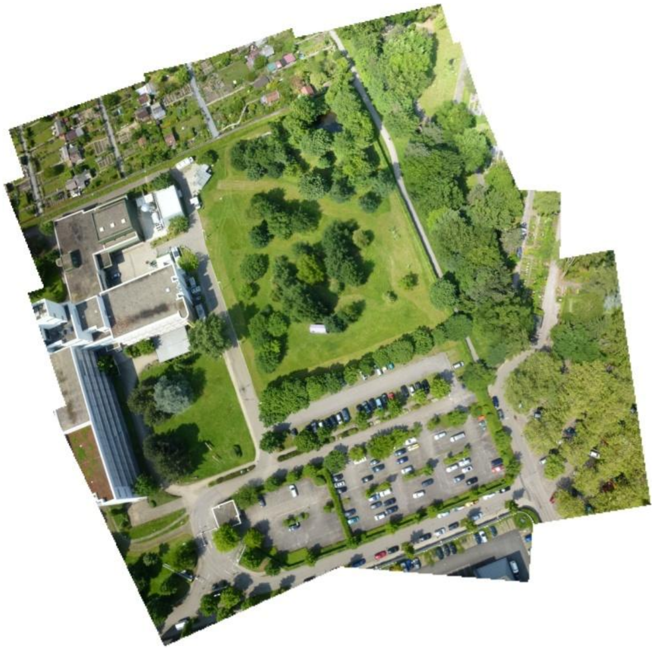


Figure 13. Mosaic obtained after stitching individual pictures.

The partitioning and planning methods have been integrated into the AMFIS ground control station and the results of experimental flights are reported. The system is appropriate for maneuverable vehicles, such as multi-rotor aircraft.

There are several aspects that require further work. More extensive and realistic tests, with bigger and more complex areas should be performed. For efficiency and safety reasons, the planning tool should be improved so that the aircraft do not cross the boundaries of the region of interest. The system should also automatically detect and provide solutions to situations where multiple flights are necessary due to range limitations of the vehicles. Another interesting extension would be to adapt the system to accommodate fixed-wing aircrafts, which are not able to perform sharp turns. Finally, it would be very interesting to explore the operational aspects and study how the aerial image acquisition system should be integrated into the decision making processes during emergency situations.

Acknowledgment

This work was carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme. This Programme is supported by the Marie Curie Co-funding of Regional, National and International Programmes (COFUND) of the European Commission.

References

[1] E. Santamaria, F. Segor, I. Tchouchenkov, and R. Schönbein, “Path Planning for Rapid Aerial Mapping with Unmanned Aircraft Systems”, Proceedings of the Eighth International Conference on Systems ICONS, 2013.

[2] E. Santamaria, F. Segor, and I. Tchouchenkov, “Rapid Aerial Mapping with Multiple Heterogeneous Unmanned Vehicles”, Proc. of the 10th International Conference on Information Systems for Crisis Response and Management ISCRAM, 2013.

[3] F. Segor, A. Bürkle, M. Kollmann, and R. Schönbein, “Instantaneous Autonomous Aerial Reconnaissance for Civil Applications - A UAV based approach to support security and rescue forces”, 6th International Conference on Systems ICONS, 2011, pp. 72-76.

[4] F. Segor, A. Bürkle, T. Partmann, and R. Schönbein, “Mobile Ground Control Station for Local Surveillance”, Proc.of the Fitth International Conference on Systems ICONS, 2010.

[5] H. Choset, “Coverage for robotics - A survey of recent results”, Annals of Mathematics and Artificial Intelligence vol.31, 2001, pp.113-126.

[6] 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, R. Alami and R. Chatila and H. Asama, Eds. Springer Japan, 2007, pp. 221-230.

[7] H. Choset, “Coverage of Known Spaces: The Boustrophedon Cellular Decomposition”, Autonomous Robots vol. 9, 2000, pp.247-253.

[8] R. Mannadiar and I. Rekleitis, “Optimal coverage of a known arbitrary environment”, IEEE International Conference on Robotics and Automation (ICRA), 2010, pp. 5525 -5530.

[9] W. Huang, “Optimal line-sweep-based decompositions for coverage algorithms”, IEEE International Conference on Robotics and Automation (ICRA) vol.1, 2001, pp. 27 - 32.

[10] S. Hert, S. Tiwari, and V. Lumelsky, “A terrain-covering algorithm for an AUV”, Autonomous Robots vol.3, 1996, pp.91-119.

[11] V. Lumelsky, S. Mukhopadhyay, and K. Sun, “Dynamic path planning in sensor-based terrain acquisition”, IEEE Transactions on Robotics and Automation 6(4), 1990, pp.462 -472.

[12] Y. Gabriely and E. Rimon, “Spanning-tree based coverage of continuous areas by a mobile robot”, Annals of Mathematics and Artificial Intelligence vol. 31, 2001, pp.77-98.

[13] P. J. Jones, “Cooperative area surveillance strategies using multiple unmanned systems”, PhD thesis, Georgia Institute of Technology, 2009.

[14] M. Weiss-Cohen, I. Sirotin, and E. Rave, “Lawn Mowing System for Known Areas”, International Conference on Computational Intelligence for Modelling Control Automation, 2008, pp. 539 -544.

[15] A. Zelinsky, R. Jarvis, J. C. Byrne, and S. Yuta, “Planning Paths of Complete Coverage of an Unstructured Environment by a Mobile Robot”, International Conference on Advanced Robotics, 1993, pp. 533—538

[16] A. Barrientos, J. Colorado, J. del Cerro, A. Martinez, C. Rossi, D. Sanz, and J. Valente, “Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots”, J. Field Robot. 28(5), 2011, pp. 667--689.

[17] R. De Carvalho, H. Vidal, P. Vieira, and M. Ribeiro, “Complete coverage path planning and guidance for cleaning robots”, ' IEEE International Symposium on Industrial Electronics, 1997. ISIE 97, vol. 2, pp. 677 -682.

[18] Y.-H. Choi, T.-K. Lee, S.-H. Baek, and S.-Y. Oh, “Online complete coverage path planning for mobile robots based on linked spiral paths using constrained inverse distance transform”, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009. IROS 2009, pp. 5788 -5793.

[19] J. W. Kang, S. J. Kim, M. J. Chung, H. Myung, J.H. Park, and S. W. Bang, “Path Planning for Complete and Efficient Coverage Operation of Mobile Robots”, International Conference on Mechatronics and Automation, 2007. ICMA 2007', pp. 2126 -2131.

[20] T. Passenbrunner and L. del Re, “SkyObserver: Decentralized, real-time algorithm for deployment of a swarm of Unmanned Aircraft Systems”, Proceedings of the 9th IEEE International Conference on Control and Automation, 2011.

[21] H. Daniel, S. Rohde, N. Goddemeier, and C. Wietfeld, “Cognitive Agent Mobility for Aerial Sensor Networks”, IEEE Sensors Journal, 2011, 11, 2671 -2682.

[22] S. Rohde, N. Goddemeier, C. Wietfeld, F. Steinicke, K. Hinrichs, T. Ostermann, J. Holsten, and D. Moormann, “AVIGLE: A system of systems concept for an avionic digital service platform based on Micro Unmanned Aerial Vehicles Systems”, IEEE International Conference on Man and Cybernetics, 2010.

[23] K. Daniel, B. Dusza, A. Lewandowski and C. Wietfeld, “AirShield: A system-of-systems MUAV remote sensing architecture for disaster response”, 3rd Annual IEEE Systems Conference, 2009.

[24] M. Quaritsch, K. Kruggl, D. Wischounig-Strucl, S. Bhattacharya, M. Shah, and B. Rinner, “Networked UAVs as aerial sensor network for disaster management applications” , e & i Elektrotechnik und Informationstechnik, Springer, 2010, 127, 56-63.

[25] M. Quaritsch, R. Kuschnig, H. Hellwagner, B. Rinner, “Fast Aerial Image Acquisition and Mosaicking for Emergency Response Operations by Collaborative UAVs”, Proceedings of the 8th International ISCRAM Conference, 2011.