Path Planning for Rapid Aerial Mapping with Unmanned Aircraft Systems

A fast complete coverage path planning algorithm with no backtracking

Eduard Santamaria, Florian Segor, Igor Tchouchenkov, 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. In this paper, we present a new complete coverage path planning algorithm that provides support for no-fly zones inside the area, and also increases efficiency by reducing the number of turns. The proposed method is suitable for complex areas that may contain known obstacles and no-fly zones, to be covered by maneuverable systems such as multirotor aircraft.**

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, a civil application is not only conceivable but already reality with some facilities of fire brigade or police. Due to the current rate of development and the varied application possibilities of miniaturized unmanned aircraft, it can be assumed that usage of these systems will rapidly increase.

At the Fraunhofer IOSB the most different applications are examined and systems are developed which should allow rescue and emergency forces to use UAVs in an easy and intuitive way [1,2].

Within the scope of this research in 2011 we presented an inspection system for generating high resolution up-to-date aerial images to support rescue forces and first responders [3]. It uses the payload capacity of one or several UAVs to scan a defined area with high-resolution image sensors to generate an image mosaic from the accumulated single frames. These mosaics can be generated with low resources and costs and their applications have been demonstrated in different realistic scenarios.

During different trials the concept proved itself to be right and the system generated useful results. However, the restriction that some zones within the inspected area could not be excluded turned out as a disadvantage. Because this feature is sometimes necessary to avoid obstacles, forbidden zones or areas of no interest, an adaptation of the original solution was required. As a part of this adaptation different alternatives were examined to further improve the efficiency of the process.

The focus of this advancement is, on this occasion, the shortening of flying time by simplifying the flight route, reducing the number of turning points, together with added support for no-fly zones inside the area of interest.

Results are compared with the raster based method previously in place. The comparison is therefore based on three metrics: total flied 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.

II. RELATED WORK

A taxonomy proposed by Choset divides coverage path planning algorithms into heuristic based algorithms and algorithms that are based on a cellular decomposition. The latter can rely on an exact, a semi-approximate or an approximate decomposition [4]. Heuristic based algorithms combine heuristics and randomness to drive the exploration process. These methods do not require expensive sensors and do not consume much computational resources. They 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 pattern, and the path planning algorithm decides the order in which cells are visited [5, 6, 7, 8].

Semi-approximate algorithms rely on a partial discretization of space where cells are fixed in width but the top and bottom can have any shape [9, 10]. 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. The size of each cell will be directly related to the sensor footprint. 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 [11, 12, 13]. These methods generate a continuous path around the spanning tree. In the likely situation of the UAV landing in the same zone where it took off, this is a good property. 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 [14]. 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 [15]. 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 [16]. 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 [17]. Such kind of pattern will not very efficient, in terms of the number of turns, when the cell grid has complex contours.

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 [18]. 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.

The work presented in this paper continues previous efforts at Fraunhofer IOSB. 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.

In this paper we present a new method for generating a path that completely covers the area. Computation is fast and the number of turns is significantly reduced when compared against the results obtained by the algorithm currently in use. Other metrics, such as travelled distance and jumps between non-adjacent cells, are also compared.

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 on which decisions can be made 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.

Figure 1 Photomosaic of Biotope at the Rhein River.

However, helicopters or supervision airplanes that can supply this information are very expensive or even 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 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 similar results is proposed to close this gap.

The “photo flight” 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).

The AMFIS system, which the photo flight tool is part of, provides a ground control station with an intuitive and ergonomic interface, and is capable of controlling multiple aircraft simultaneously [3]. By using more than one single UAV, the same search area can be covered in less time or respectively a bigger area can be searched in the same time. The automatic path planning capabilities presented in this paper reduce the workload of the user, who only has to define the area of interest and indicate which vehicles can be used to perform the mission.

IV. PATH PLANNING ALGORITHM

The algorithm presented in this paper is applied once the grid of cells that approximates the area of interest is available. The AMFIS system already performs this step, together with a division into subareas for a multi-UAV scenario. The algorithm can then be used to compute a path to cover each subarea.

The concept of stride is central to the algorithm. We define a stride as a sequence of consecutive adjacent cells with no 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. 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:

- We do not add c_n if it is already in the generated path or if it falls outside the boundaries of the area.

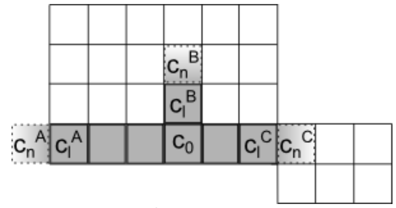

Figure 2 Stride formation starting at cell c_0.

- If the previous condition does not hold, and $L(c_l) = 2$, we always add c_n to the stride, because c_n is the only possible cell where we can go to 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 we are 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.

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 $L(c_n^B) = 0$. Finally, in stride C, 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, 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.
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 neighbors with more limits or visited cells around it, (2) the ones located in the contour of the area, and (3) the ones that lead to a longer stride. 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, the addition of a cell to the path partitions the area, creating two, or more, subareas of disconnected unvisited cells. When this happens, the algorithm chooses to visit the cells of the smallest sub-area first.

 ISBN: 978-1-61208-246-2

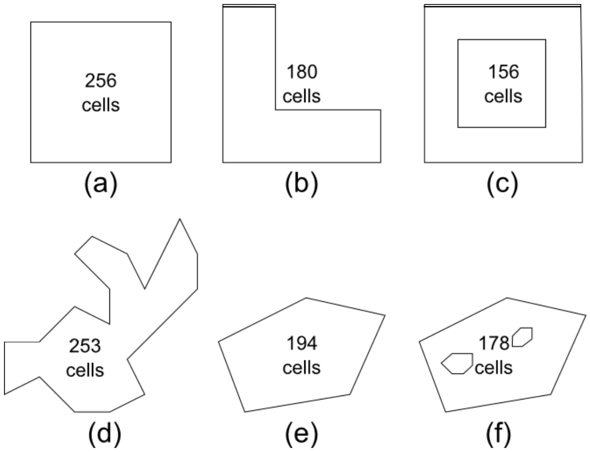

Figure 3 Contours, with their number of cells, used in tests.

Another situation that needs to be addressed happens 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 a path between the current cell and each one of the unvisited cells located next to cells already visited. These paths are computed using distance transform path planning as described by Zelinsky et al. in [15]. 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.

One final step performs some clean-up on the generated path. If the path contains a sequence of revisited cells that connect two adjacent cells, the sequence of revisited cells is removed from the path. In this way, the UAV will transition directly from the first cell to the adjacent one avoiding unnecessary repeats.

V. 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:

- 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 we compare the number of jumps between non-adjacent cells, instead of the number of revisited cells.
- 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.

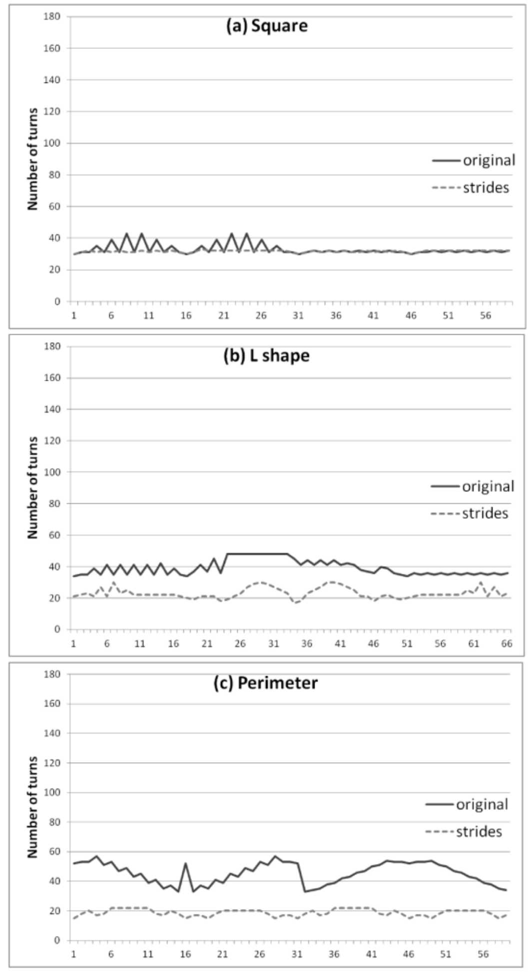

Figure 4 Number of turns starting at each contour cell of a, b and c.

- 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 teeth, avoiding its propagation into the inner parts of the area (see Fig. 7b).

 ISBN: 978-1-61208-246-2

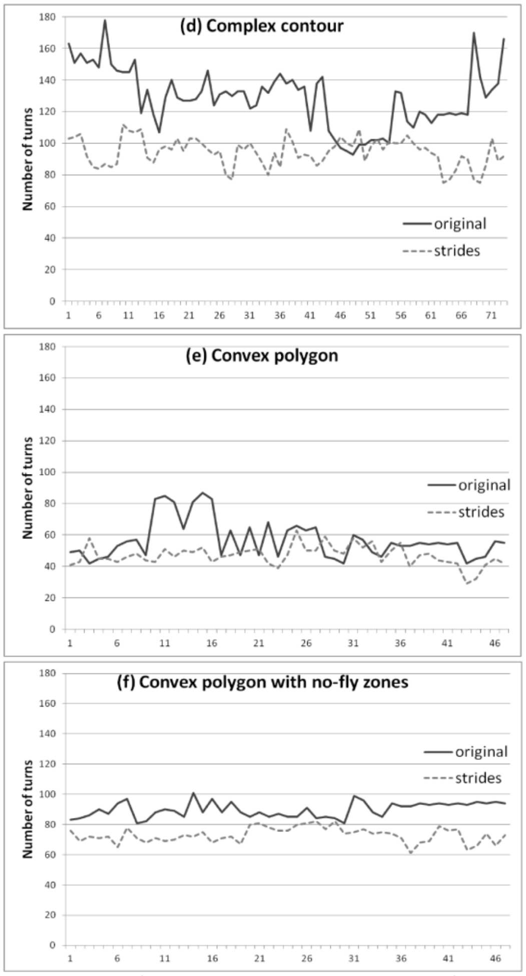

Figure 5 Number of turns starting at each contour cell of d, e, and f.

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.

VI. CONCLUSION AND FUTURE WORK

In this paper we introduced our most recent work and advances in complete coverage path planning for generating image mosaics.

The proposed algorithm is able to compute the path in a very short time, which makes it suitable for rapid response situations. Additionally, a path computation criterion that prioritizes the selection of long straight segments results in a reduced number of turns, which can be seen as a big advantage particularly in the case of a fast moving unmanned aircrafts. Its ability to scan the area in different directions and the fact that it does not rely on pre-defined patterns allow it to efficiently generate complete coverage paths for complex contours which may contain holes.

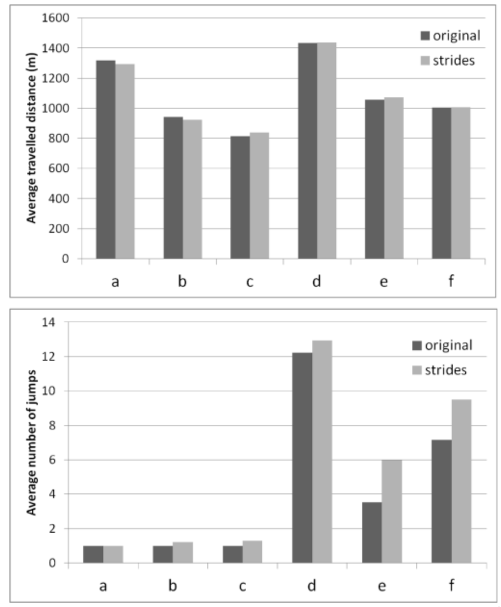

Figure 6 Average travelled distance (top) and average number of jumps (bottom).

The presented path planning method is appropriate for maneuverable systems, such as multirotor aircraft, and scenarios with known obstacles.

The photo flight tool of the AMFIS system will be updated to rely on the new algorithm for the path computation and support will be added to enable the user to specify no-fly zones. An essential aspect of the photo flight software is its support for multiple aircraft flying in parallel. In this case, the area of interest must be split among the available UAVs. In this point work is still pending as we see some potential for improvement to the system.

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] Bürkle, A., “Collaborating Miniature Drones for Surveillance and Reconnaissance”, Proc. of SPIE Vol. 7480, 74800H, Berlin, Germany, 1-2 September (2009).

[2] Leuchter, S., Partmann, T., Berger, L., Blum, E.J. and Schönbein, R., “Karlsruhe generic agile ground station”, Beyerer J. (ed.), Future Security, 2nd Security Research Conference, Fraunhofer Defense and Security Alliance, 159-162 (2007).

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

[5] 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.

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

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

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

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

[10] V. Lumelsky and 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.

[11] 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.

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

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

[14] A. Zelinsky and R. Jarvis and 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

[15] A. Barrientos and J. Colorado and J. del Cerro and A. Martinez and C. Rossi and 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.

[16] R. De Carvalho and H. Vidal and 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.

[17] Y.-H. Choi and T.-K. Lee and 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.

[18] J. W. Kang and and S. J. Kim and M. J. Chung and H. Myung and 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.

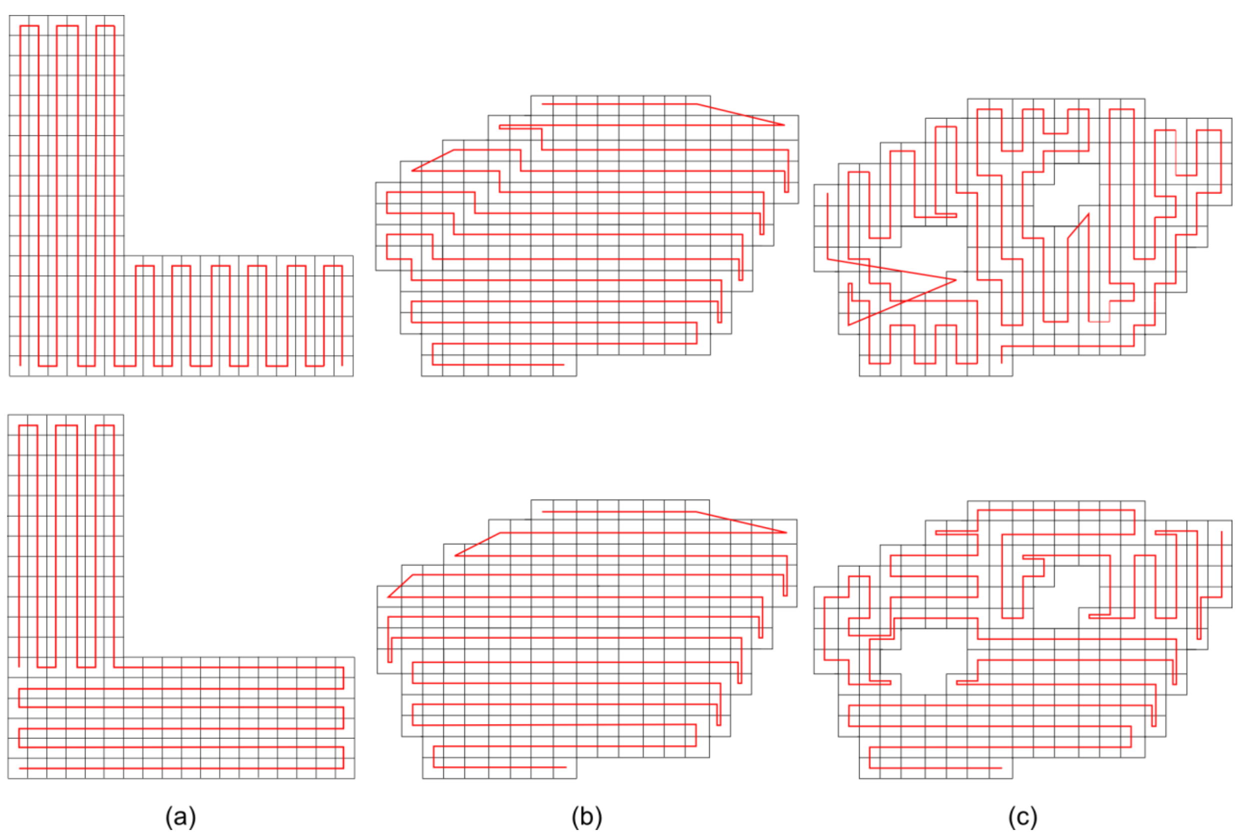

Figure 7 Complete coverage path generated for some example contours with both original (top) and strides based algorithm (bottom). In figure c the transitions between non-adjacent cells are also included.