SENSOR AND COVERAGE PATH PLANNING FOR THE MONITORING OF AIRCRAFT EMERGENCY LANDING SITES

David Nospes[1], Peter Stütz[1]

[1]Institute of Flight Systems, University of the Bundeswehr Munich, 85577 Neubiberg, Germany

Abstract

This work presents our approach to sensor and coverage path planning for the monitoring of a set of aircraft emergency landing sites with a single onboard camera in the context of the MOREALIS project. Due to ground sample distance requirements of obstacle detection, often only partial landing sites can be processed by the sensor in one view. Therefore, a monitoring sequence for multiple landing sites must be created and combined with an area specific coverage plan to obtain an overview of the complete hazard situation. For this, we investigate the combination of different coverage path planning methods with a landing site monitoring sequence generation procedure. We investigate three different coverage path planning approaches: two of these are the simple patterns of a *Spiral* and a *Creeping Line* overlayed over the landing site polygon as well as an approach for of solving the problem as a *Traveling Salesman problem* with a simple heuristic by using a grid of point across one landing site as nodes. The landing site monitoring sequence generation method treats all of the emergency landing sites as single nodes and connects these by formulating the problem as a combinatorial traveling salesman approach. The resulting order is determined with respect to the pairwise landing site distance and the landing site priority influenced by the estimated sensor performance for hazard detection as well as the relative emergency landing site suitability for landing. The combinations of both aspects are evaluated on a dataset of landing site sets regarding their capabilities of area coverage as well as regarding their time required for a complete scan. Furthermore, we perform a test flight in our simulation and evaluate the number of missed obstacles on scanned landing sites for each combination. The results regarding our experiment with the created dataset show the most promising results for the simple *Creeping Line* pattern, leading to the highest coverage with the lowest required time to execute the full sensor and coverage path followed by a small margin by the *TSP* solution. The test flight showed that no obstacles at the scanned landing sites were missed by any approach, but the number of landing sites monitored differed.

Keywords: sensor path planning, coverage path planning, landing site monitoring, ultralight aircraft

1. Introduction

Safety critical situations may occur suddenly and unpredictably during flight. Worst cases require the immediate descent to off-airport emergency landing sites. Evaluating these kinds of sites for their suitability and ranking them appropriately requires the fusion of uncertain information about the sites itself, their environment and the hazard situation. The MOREALIS project addresses this problem in general aviation and investigates a pilot assistant system for micro aircraft in distress situations with automatic landing capabilities in structured and unstructured surroundings. In circumstances where the pilot is incapable of further controlling the aircraft, the decision for an automatic descent to a suitable landing site must be made timely and shall be based on as few uncertainties as possible. To reduce these uncertainties regarding possible hazards, a fast and reliable monitoring of the approach destination as well as the coverage of potential alternatives and last resorts is necessary. Furthermore, an additional benefit of this verification is the possibility to not only maintain a single landing site for an emergency landing, but to maintain a set of possibly valid landing options. In this context, we present an approach for single sensor and coverage path planning with a single electro-optical camera mounted on a gimbal, which strives to increase the capabilities of observing multiple

pre-screened sites with a high potential for an emergency landing. *Ground Sample Distance (GSD)* requirements for hazard detection during flight often do neither allow for the monitoring of multiple nor of single complete landing sites in one view. Therefore, to achieve this verification of hazard freeness, a combination of sensor path planning to generate a monitoring sequence for landing sites as well as coverage path planning for individual landing site scanning is required.
The remainder of this paper is structured as follows. First an overview of related work is presented and our contributions are highlighted. Then in chapter 3 the combination of a landing site monitoring sequence generation approach with multiple coverage path planning methods is described, followed by a short description of our experimental setup and our results. Finally, a conclusion and an outlook for future work is given.

2. Related Work

Sensor and coverage path planning techniques have been investigated in multiple domains, often applying methods originating from research fields without the application of sensors. In sensor path planning often spatially located targets must be observed without the need of covering complete areas during surveillance, but rather observing multiple scattered targets. Examples include fixed sensor-oriented UAV path planning to monitor multiple regions [1], sensor-oriented path planning for information collection with multiple UAV [2] as well as sensor path planning and scheduling for aerial multi-target tracking with a single, gimbaled sensor [3]. Coverage path planning is investigated in a wide range of applications in robotics and related areas. Surveys on this topic can be found in [4, 5, 6]. Applications of sensor-oriented coverage path planning often approach the problem by generating a path with minimal overlap through a set of sensor footprints which fully cover a given area. Multiple surveys highlight corresponding methods for aerial applications, including *Creeping Line* as well as *Spiral* based patterns [7, 8]. Additionally, [9] proposes a coverage path planning for an aerial survey of an area under influence of wind with an UAV carrying a sensor.
The application of monitoring multiple emergency landing sites requires information about the targets, which are static landing zones based on a variation of [10], as well as information about their suitability for the actual landing and for their observation, which are estimated based on [11] and [12]. With this information a landing site sequence for monitoring can be created. In this work, we compare different combinations of a landing site monitoring sequence generation approach with coverage path planning techniques for their applicability in emergency landing site monitoring and hazard detection. The application and evaluation of these methods is performed by creating a dataset with monitoring sequences and analyzing the impact of different coverage path planning methods on the area coverage as well as the required time for scanning of all the landing sites. Finally, we simulate obstacles on multiple sets of landing sites and evaluate each combination of sensor path and coverage path planning for their landing site scanning capabilities and the number of missed obstacles at the landing sites during a simulated flight.

3. Sensor and Coverage Path Planning

Due to the requirement of light, low-cost cameras for the use with ultralight aircraft, limitations regarding the available sensor resolution arise. Combined with the for our use case of obstacle detection required *GSDs*, at higher aircraft heights above the terrain often a complete landing site monitoring within one sensor view is not possible. Therefore, an approach for the successive verification of obstacle clearance for a set of potential landing sites is required. The sensor and coverage path planning we propose aims for creating a sensor path consisting of a prioritized sequence of landing sites for their complete monitoring as preparation for potential distress situations combined with a coverage path planning to account for the complete monitoring of the landing sites within the sequence. An illustration of the problem is shown in Figure 1. The generation of the landing site sequence is implemented with respect to information of the relative suitability of potential landing sites for an emergency landing, of the probability of a successful hazard detection given the available sensor capabilities estimated using a sensor performance model as well as information of the actuator effort required and therefore the time required by the gimbal to switch between landing sites. Our approach to the formulation of a method creating such a sequence of landing sites based on these requirements has been shown in [13]. The centroids of the landing sites are computed and treated as nodes to create a complete graph. The edges between the nodes are asymmetric and

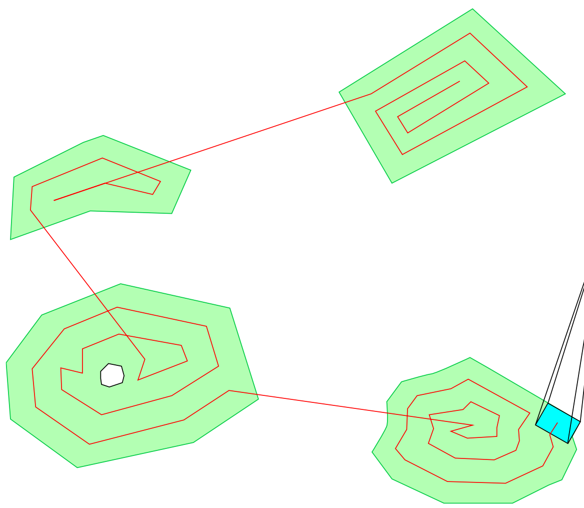

Figure 1. Illustration of a possible sensor path composed of the combination of a monitoring sequence for potential emergency landing sites and their corresponding coverage paths to find potential hazards using a gimbal. The sensor path is drawn as a red line, the potential emergency landing sites are shown as green polygons and the sensor cone as black outlined pyramid with a turquoise sensor footprint.

calculated by performing a weighted sum of the distance between the centroids, the importance of the landing sites for an emergency landing and the estimated object detection performance when orienting the sensor to the landing site centroid. The first is used as an approximation for the gimbal actuator effort required to switch between landing sites and is based on a distance matrix of pairwise Euclidean distances between the landing site centroids. Since we perform object detection for our hazard detection, object detection performance is used as an estimate of the perception performance of an aerial surveillance system and therefore for the probability of successful hazard detection. It is estimated based on a graph-based representation of influences on sensor image based object detection. Influences are based on target object specific parameters, sensor system parameters as well as scene parameters and combinations of them. A subset of the model described in [11] is used in our case, including the object distance, the sensor field of view, the sensor resolution, the ground sample distance, the sensor depression and the object size. Environmental parameters and potential occlusions through the terrain, the aircraft itself or weather conditions like clouds are considered but currently not evaluated. The landing site emergency landing priority depends on a wide range of parameters including the distress type, the pilot and aircraft condition as well as environmental properties, weather conditions, the obstacle situation at the site and the duration until emergency services are available. It is based on ranking a catalogue of runway like stripes at potential emergency landing sites with the above-described information as inputs using a Bayesian Network approach [12]. The created graph from these influences is then solved by treating it as a variant of the traveling salesman problem. Investigations on the sensor path planning for numerous methods showed the best results for our use case using the *Farthest Insertion* heuristic in combination with *2-Opt* for result enhancement [13]. However, this only creates a sequence of landing sites and does not incorporate the monitoring of the full landing sites. To achieve this, we incorporate coverage path planning into the approach. For each available landing site in the landing site sequence, we compute a coverage path and insert it into the appropriate position of the schedule of the sensor scheduler. To solve the coverage path planning problem, we implement 3 different methods for pattern generation: two simple pattern creation methods and one optimization procedure. The first one consists of a spiral pattern. This kind of pattern has been implemented for coverage path planning in literature for example as *Energy Aware Spirals* [14]. In our case we only implement a simple *Archimedean Spiral* pattern. The second one generates *Creeping Line*s, which is quite commonly used for coverage path planning [7], and the third one places a fixed distance grid over the landing site polygon to decompose it into equally sized cells and performs a traveling salesman optimization based on the distance between the grid points to calculate a scanning pattern. The distance between different sweeps and points of the coverage path is defined by the sensor footprint, since it determines the in the sensor image visible area when following the coverage path line. Since the sensor footprint is not necessarily aligned in the direction of the coverage path, we use the smaller edge length of the sensor cone as a fixed distance between sensor sweeps. We compute this distance according to [15] with *h* being the height of the sensor cone, *gsd* being the *Ground Sample Distance* and *r* being the sensor resolution as shown in equation 1. To respect potential inaccuracies of the control of the gimbal as well as of the sensor *Field of View (FoV)* and therefore potential deviations of the real *GSD*, we include an additional safety factor s_f of 0.8 into the calculation of the distance between sensor sweeps. This increases the coverage path length due to the creation of overlap between parallel lines of the coverage path but increases the coverage in non-perfect conditions.

$$h = gsd\, r\, s_f \tag{1}$$

An additional important factor for gimbal control is the calculation of the footprint step size *s* with the footprint speed v_f and the timestep between the different footprints *t* as in equation 2 from [15]. Setting the footprint speed too high results in a loss of coverage in corners and curves depending on the quality of the available gimbal and its control. Lower values in contrast increase the total monitoring time. While this does not influence the planning of the coverage path itself, it impacts its execution and therefore the practical results.

$$s = v_f\, t \tag{2}$$

In the following the process of generating the different patterns is described. Potential landing sites are represented for this process by polygons consisting of a sequence of vertices containing information about its x, y and z location in the real world. These shapes often represent convex polygons due to having their origin in farmland and greenfields, but they may be of more complex shapes too. For now, we only implemented a couple of simple approaches to generate first insights in hazard monitoring capabilities with our setup.

3.1.1 Coverage Path Planning using Archimedean Spirals

Spirals are a common pattern in coverage path planning [7]. As a simple solution to a spiral pattern, we implement an *Archimedean Spiral* since it should offer a good basis for a fast landing site area monitoring due to its absence of corners and therefore its support for a continuous motion of the gimbal. However, the implementation of this continuous motion is partly prevented by the need for integrating these spirals into landing site polygons and therefore cutting them to decrease the length of the resulting coverage path. Furthermore, parts of the coverage path outside of the landing site polygon would not contribute to successful hazard detection. We compute the *Archimedean Spiral* for a single landing site as described in the following. Given a landing site polygon consisting of multiple points in a local tangent plane coordinate system, in our case the *North East Down (NED)* coordinate system, first the centroid of the polygon is computed to calculate the center point C_p of the Archimedean spiral. The next step for Archimedean spiral calculation is a preparation step and consists of the calculation of the length *l* of the longest dimension of the polygon. This is followed by the calculation of the number of necessary revolutions r_e to cover the complete area of the landing site polygon, if we would start at border of the polygon using the sensor cone width *h* as shown in equation 3. Since we start the spiral at the center, we need to divide this number of revolutions by 2 for further computations.

$$r_e = {}^{l}/_{h} \tag{3}$$

We then calculate the angle of the φ of the endpoint of the spiral by equation 4.

$$\varphi = r_e \pi \tag{4}$$

This is followed by the calculation of the length of the spiral as in [16] with equation 5 and equation 6.

$$L = \frac{a}{2}(\varphi\sqrt{\varphi^2+1} + Arsinh\varphi) \quad (5)$$

$$a = \frac{h}{2\pi} \quad (6)$$

Finally, the step size in angles a_s for which discrete points are placed on the spiral using the resolution r in meters are calculated as in equation 7.

$$a_s = \frac{L}{r} \quad (7)$$

The complete process of the generation of an Archimedean spiral covering the complete landing site polygons is shown in the listing below.

1. Compute the polygon centroid
2. Translate the polygon centroid to the coordinate origin
3. Compute the polygons longest dimension
4. Calculate the number of necessary revolutions r_e
5. Calculate the angle φ of the complete spiral
6. Calculate the spirals length L
7. Calculate the angles a_s at which points should be set
8. Calculate points on the spiral for each angle
9. Merge the points to a line
10. Exclude line parts outside the polygon
11. Buffer the line with h/2 to get the covered area
12. Check for uncovered areas of the polygon
13. Buffer the uncovered areas with h/2
14. Add line parts intersecting the buffered, not covered areas

Some examples of estimated *Archimedean Spiral* coverage paths are shown in Figure 2.

3.1.2 Coverage Path Planning using Creeping Lines

The *Creeping Line* pattern is a rather straightforward approach to solving the coverage path problem. It consists of straight lines connecting different landing site polygon shape based key points in a back-

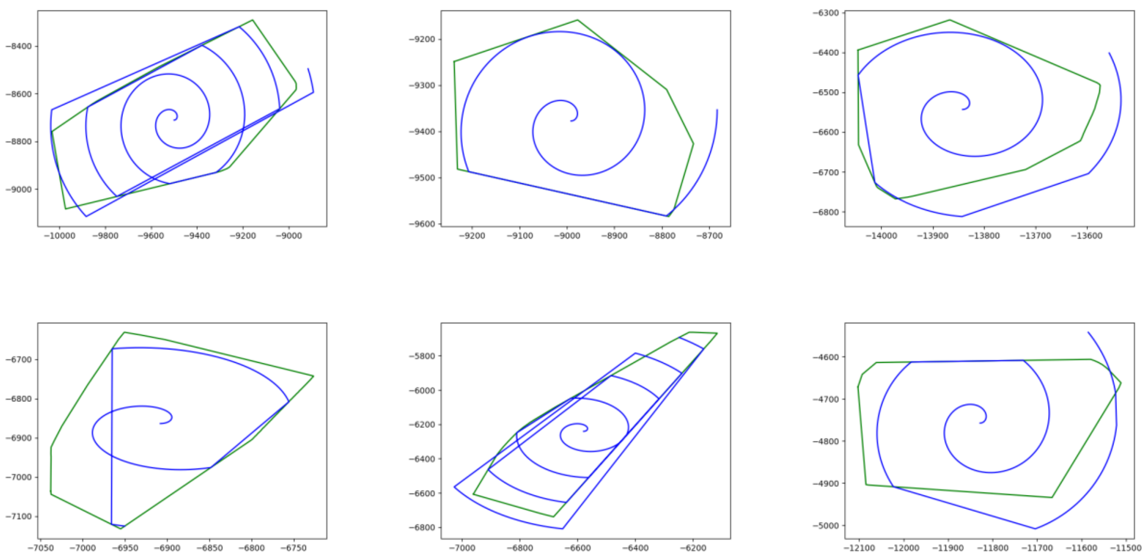

Figure 2. Computed Archimedean spiral coverage paths for the monitoring of 6 different landing sites.

and-forth style. The distance between the sweeps in different directions is defined by the sensor cone and the sweep direction is alternating in or opposite the direction of the smallest dimension. For simple shapes like rectangles, the estimation of a complete coverage path is rather simple since it only requires the calculation of the required number of lines to fill the rectangle width while bouncing between the shapes borders with a sensor cone specific distance between the lines. More complex shapes require more effort to obtain full coverage [7]. Therefore, we perform the following procedure to calculate a creeping line pattern based coverage path.

1. Polygon centroid computation
2. Calculation of the polygons longest dimension
3. Rotate the longest dimension onto the x-axis
4. Calculate the approximate number of sweeps by length of the longest dimension
5. Set starting point as bounding box corner + h/2 in x direction
6. For each necessary sweep:
    a. Calculate the next point by adding or subtracting the bounding box height to the y value previous point
    b. Create a line from both points
    c. Buffer the line with h/2 to get the covered area
    d. Cut the line at minimum and maximum height value of the intersection of the polygon and the buffered line
    e. Calculate the next point by adding h/2 to the x value of the endpoint of the line
7. Merge lines

Some examples of estimated creeping line coverage paths are shown in Figure 3.

3.1.3 Coverage Path Planning using TSP

The third way of creating a coverage path pattern we use is implemented by optimizing a coverage path onto a grid which is based on landing site and sensor cone shape. This implicates two subproblems: the generation of the grid with points overlaying the landing site in way leading to a complete coverage when connected, which can be performed by creating a symmetric grid using approximate cellular decomposition [7], while having a minimal number of points and solving the actual traveling salesman problem to connect these points in a way leading the smallest possible resulting coverage path length. We approach the first problem by simply creating a symmetric grid which overlays the landing site polygon and its nearby surroundings with a fixed distance between the grid points. A denser grid provides better coverage but also increases the number of points and therefore potentially the computation time for finding a solution with the TSP approach. To solve

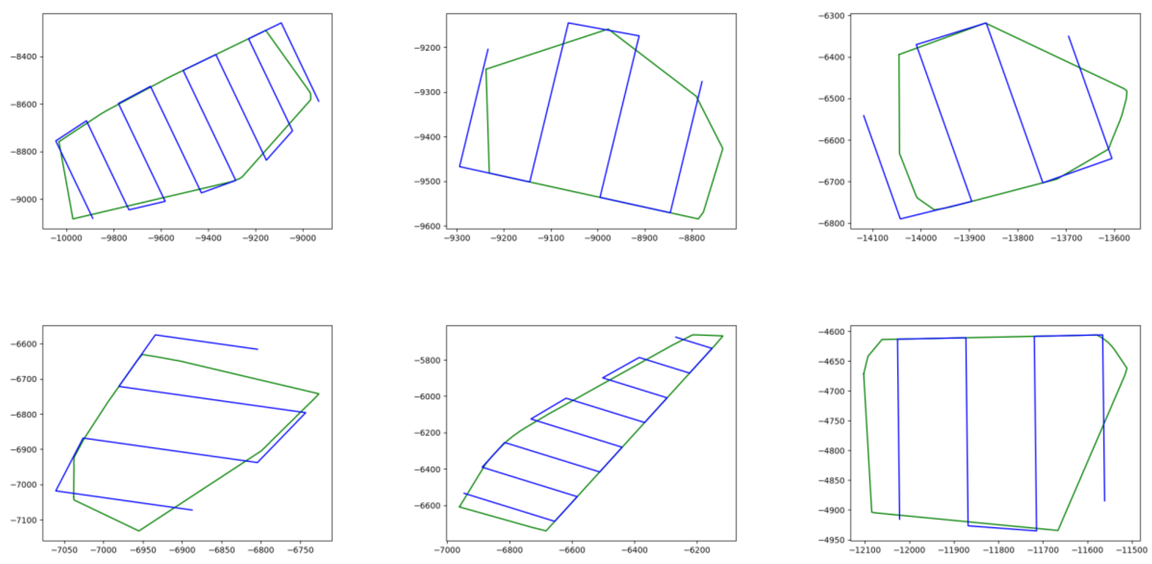

Figure 3. Computed creeping line coverage paths for the monitoring of 6 different landing sites.

connecting the points of this grid, for each pair of points the Euclidean distance matrix is calculated and stored in a distance matrix. An initial solution is created by simply connecting all points in the grid by column for each row. This solution is then gradually enhanced by using the 2-Opt-algorithm, which selects pairs of adjacent edges, swaps the edge connections, and evaluates the resulting tour costs for improvements. The procedure is described in detail in the following.

1. Buffer landing site polygon with h/2
2. Point sampling by creating a grid within the buffered polygon with a point distance of h
3. Moving of all grid points outside the polygon onto the polygon border
4. Calculation of the Euclidean distance matrix for an open TSP
5. 2-Opt solving of the distance matrix with a maximum allowed time of 5 seconds

Some examples of estimated creeping line coverage paths are shown in Figure 4.

4. Experiments

To compare the suitability of the combinations of our sensor path planning approach with the different coverage path planning methods we perform 2 different types of experiments. The first one aims at finding the best coverage path planning method for a dataset of landing site sets with given landing site sequences. This guarantees a comparison of the coverage path planning methods without the influence and potential distortion of the results through the execution of a heuristic potentially leading to different sequences of landings sites for the monitoring in each run. We then evaluate the achieved coverage as well as the time to complete the landing site monitoring with our simulated sensor setup for a fixed sensor position over the complete landing site sequence. The measurement of the covered area is performed by projecting the sensor footprint on the 3d terrain, merging all the footprints for one landing site sequence to a single entity and evaluating its coverage percentage when combined with the ground truth landing site geometries. The gimbal actuator influencing the result is implemented as an ideal motor without loss of torque. The connected dynamic gimbal model takes the gimbal inertia as well as the rotational speed dependent friction into account. Gimbal attitude and velocity control is performed using a PID-Controller. The target attitude is estimated from geolock. Furthermore, the kinematic aircraft model for our test purposes is based on [17]. We perform this for a set of 81 landing site sequences with an average of 36 landing sites per sequence and a standard deviation of 10 landing sites. The second experiment is a flight from Munich to the Starnberger Lake. The aircraft moves at a speed of 30 m/s at 1000 m above mean see level in the direction of 270° west until it passes the airport Oberpfaffenhofen. The gimbal speed is capped at a maximum of 70°

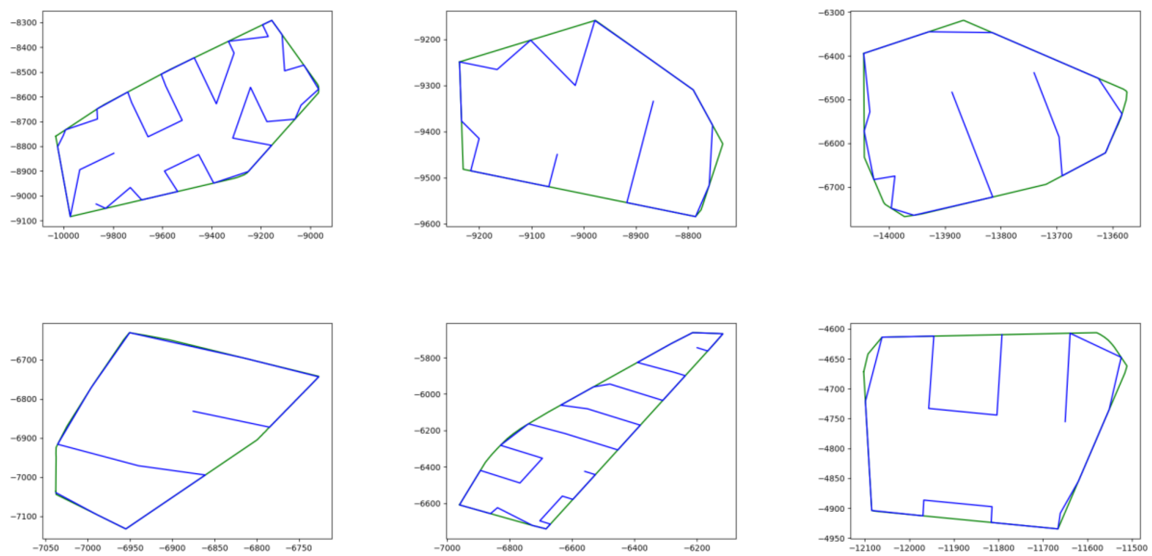

Figure 4. Computed TSP solutions for the coverage path problem based on a point grid overlayed over the landing site polygons for the monitoring of 6 different landing sites.

per second. During this flight we simulate obstacles on the landing sites to monitor. The obstacles move in random directions within the landing site until they hit a border of the landing site and then change their direction randomly. To only compare the scheduling without further influences from potential object detection inaccuracies, we assume a perfect detector meaning every sensor image with an obstacle in it results in a successful detection. Therefore, we assume every obstacle within the projected sensor cone as detected. For this experiment, the sensor scheduling is performed in a simple way: First a sensor path for all currently available landing sites in glide range is estimated. Then this sensor path is processed and the coverage path planning is performed for each landing site. After the full landing site sequence is processed, a new sequence is generated containing all landing sites in the new glide range. This is repeated until the airport is passed. We then compare the number of landing sites scanned as well as the number of missed obstacles on the landing sites. This compares the full sensor scheduling with landing site sequence generation and coverage path planning combined regarding their effectiveness. The evaluations are performed on a single computer equipped with an AMD Ryzen 9 3950x, 128 GB of RAM and an RTX 3090.

5. Results

The first experiment regarding area coverage of the pattern for a fixed landing site sequence shows that the simple *Creeping Line* pattern implementation leads to the most promising results when combined with our gimbal model and control. While all three patterns lead to coverage of more than 99% of all the areas over all the landing site sequences as shown in table 1, the *Creeping Line pattern* has a minimal edge over the *TSP* solution with an improvement of 0.01% and slightly better performance than the *Spiral* pattern with an improvement of 0.1% area coverage. In contrast, the *Spiral* pattern-based method is the most consistent regarding its area coverage standard deviation with only a value of 0.27, in comparison to the *Creeping Line* pattern with 0.36 and the *TSP* approach with 0.39. Therefore, it may be more difficult for the control to follow the constant curve of the spiral line in detail to achieve area coverage potentially leading to the frequent and constant output of minimal uncovered areas. This could also explain why the simplest pattern, the *Creeping Line* pattern, achieves the highest area coverage in our case. A visual comparison of the estimated sensor cone coverage overlayed over the landing site polygons can be seen in Figure 5. Regarding required monitoring time, the *TSP* version is only slightly slower than the *Creeping Line* approach with an average runtime of 254.65 seconds compared to 243.99 seconds of the *Creeping Line* approach. The Spiral pattern is a lot slower due to the increased path length resulting from the specific implementation with a total time of 344.45 seconds. Regarding runtime, the *Creeping Line* is the most stable with a standard deviation of 65.94 compared to 67.83 of the *TSP* and 92.27 of the *Spiral*. This deviation exists mainly due to the difference in the number of landing sites which need to be scanned across the dataset. A set with more landing sites is leading to a longer path which therefore increases the time required to completely scan the full set and vice versa. Therefore, a smaller standard deviation in this case mainly means a more consistent resulting coverage path length. All in all, the simple *Creeping Line* pattern achieves the best results for our case, with the *TSP* based approach being only slightly worse. In general, no implementation always covers 100% of the area. This could be due to the estimation of the sensor cone width for the coverage path planning for each landing site depending on the *GSD* calculation based the distance of the sensor to the landing sites centroid. Depending on the distance to the sensor, the size of the landing sites as well as on the sensor depression, the sensor cone therefore may be distorted and may not be rectangular shaped

Pattern	Area coverage mean (%)	Area coverage std (%)	Time mean (s)	Time std (s)
Spiral	99.49	0.27	344.45	92.27
Creeping Line	99.59	0.36	243.99	65.94
TSP	99.58	0.39	254.65	67.83

Table 1. Time and area overlap comparison for the different patterns of coverage path planning.

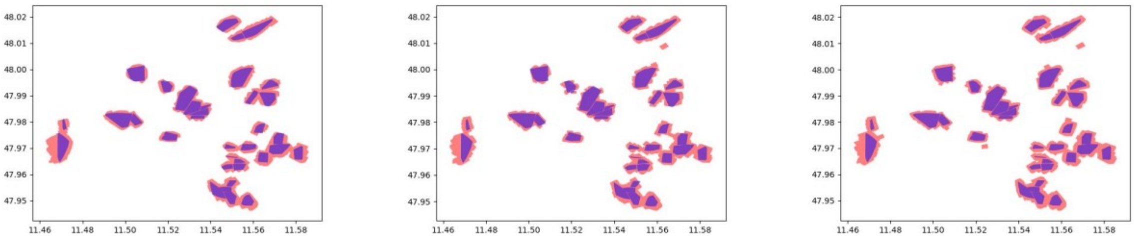

Figure 5. Exemplary visual comparison of the area coverage for the *Spiral* pattern (left), the Creeping Line pattern (middle) and the *TSP* based pattern (right) with the sensor footprint coverage at each landing site in red and the corresponding landing site polygons in purple.

which could lead potentially to uncovered areas. Another possible impact could be that more complex shapes in the dataset are not yet perfectly covered by the current coverage path implementations and due to imperfect gimbal control. In experiment 2, as expected from the area coverage in experiment 1, all obstacles on every monitored landing site have been found. However, the total number of investigated landing sites differs depending on the algorithm, with 68 out of 76 for the *Spiral* pattern, 74 out of 76 for the *Creeping Line* pattern and 76 out of 76 for the *TSP* solution. This indicates a problem with the for this experiment used simple scheduling scheme of the sensor with increasing time required for the coverage path processing. This is due to completely missing potentially interesting landing sites which are only in the glide range of the aircraft during an active sequence processing but at the exact time when a new sequence is created. While the *TSP* solution did achieve the best results in this exemplary use case due to the scanning of all landing sites, the results are also influenced by the heuristic for the creation of a landing site sequence. Since this heuristic does not necessarily lead to the same results at different runs, an above average result could lead to a faster processing and therefore a complete scanning for the *Creeping Line* pattern too, since it has proven to be the faster method in experiment 1.

6. Conclusion

This paper provided some insight to our approach on sensor and coverage path planning for emergency landing site monitoring in the context of the MOREALIS project. Different coverage path planning methods were combined with a simulated gimbal system and control and compared for their suitability. For this, two experiments were performed. The first used a dataset of 81 landing site sequences estimated by our sensor path planning approach described in detail in a previous paper. The evaluation of the different methods was performed regarding the average coverage and its standard deviation achieved by the projected sensor cone when following the coverage paths as well as the average runtime and its standard deviation. The *Creeping Line* pattern-based method achieved the highest average coverage and runtime in our experiments. The second experiment consisted of a simulated flight in the area of Munich with simulated moving obstacles at the landing sites. The sensor path planning combined with different coverage path planning methods was then tested for their capability of finding these obstacles. This showed that the coverage path planning may have a huge impact regarding the hazard detection probability, but a smart sensor scheduling for the creation and processing of the landing site sequence combined with the coverage path planning is required to achieve full landing site coverage and therefore to maintain a larger number of potential landing sites for an emergency landing. While the coverage path planning approaches achieved near perfect coverage in combination with our simulated sensor system and control, there is still room for improvement. Faster algorithms with shorter resulting coverage paths could lead to even better obstacle detection capabilities due to the decreased active sensor time required for scanning a single landing site. Future works include the investigation of sensor performance estimation for landing site observation positions aiming at increasing the chance for timely hazard detection in emergency cases and therefore to increase the overall hazard awareness to maintain a number of approachable landing site alternatives.

7. Copyright Statement

Acknowledgements

This research is funded by the German Federal Aviation Research Program (LuFo VI-I) of the German Federal Ministry for Economic Affairs and Climate Action.

References

[1] Li J., Chen J., Wang P. and Li C. Sensor-Oriented Path Planning for Multiregion Surveillance with a Single Lightweight UAV SAR. *Sensors 2018*, *18*, 548. https://doi.org/10.3390/s18020548

[2] Ergezer H. and Leblebicioğlu M. K. 3D path planning for UAVs for maximum information collection. *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, Atlanta, GA, USA, pp. 79-88, 2013, doi: 10.1109/ICUAS.2013.6564676.

[3] Koch S. and Stütz P. Sensor Scheduling for Object Tracking with UAVs: A Comparison of Scheduling Paradigms. *2021 IEEE Aerospace Conference (50100)*, Big Sky, MT, USA, pp. 1-8, 2021, doi: 10.1109/AERO50100.2021.9438139.

[4] Galceran E. and Carreras M. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61, 12, pp. 1258-1276, 2013, ISSN 0921-8890, https://doi.org/10.1016/j.robot.2013.09.004.

[5] Bormann R., Jordan F., Hampp J. and Hägele M. Indoor Coverage Path Planning: Survey, Implementation, Analysis. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD, Australia, pp. 1718-1725, 2018, doi: 10.1109/ICRA.2018.8460566.

[6] Almadhoun R., Taha T., Seneviratne L. and Zweiri Y. A survey on multi-robot coverage path planning for model reconstruction and mapping. *SN Applied Sciences*, 1, 2019, doi: 10.1007/s42452-019-0872-y.

[7] Cabreira T., Brisolara L. and Ferreira Jr P. Survey on Coverage Path Planning with Unmanned Aerial Vehicles. *Drones,* 3, 4, 2019, doi: 10.3390/drones3010004.

[8] Aggarwal S. and Kumar N. Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges, *Computer Communications*, 149, pp. 270-299, 2020, ISSN 0140-3664, https://doi.org/10.1016/j.comcom.2019.10.014.

[9] Coombes M., Chen W. H. and Liu C. Boustrophedon coverage path planning for UAV aerial surveys in wind. *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, Miami, FL, USA, pp. 1563-1571, 2017, doi: 10.1109/ICUAS.2017.7991469.

[10] Nospes D. and Stütz P. Preflight Aircraft Emergency Landing Field Estimation using Optimization. *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*, Portsmouth, VA, USA, pp. 1-9, 2022, doi: 10.1109/DASC55683.2022.9925734.

[11] Koch S., Ostler J. and Stütz P. Estimating Object Perception Performance in Aerial Imagery Using a Bayesian Approach. *2023 IEEE Aerospace Conference*, Big Sky, MT, USA, pp. 1-9, 2023, doi: 10.1109/AERO55745.2023.10115828.

[12] Nospes D. and Stütz P. Aircraft Emergency Landing Site Selection: A Hierarchical Approach. *2023 IEEE/ AIAA 42st Digital Avionics Systems Conference (DASC)*, Barcelona, Spain, 2023, pp. 1-7, doi: 10.1109/DASC58513.2023.10311169.

[13] Nospes D. and Stütz P. Sensor Path Planning and Scheduling for Aircraft Emergency Landing Field Monitoring. *2024 IEEE Aerospace Conference*, Big Sky, MT, USA, 2024, pp. 1-10, doi: 10.1109/AERO58975.2024.10521099.

[14] Cabreira T. M., Franco C. D., Ferreira P. R. and Buttazzo G. C. Energy-Aware Spiral Coverage Path Planning for UAV Photogrammetric Applications. *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3662-3668, Oct. 2018, doi: 10.1109/LRA.2018.2854967.

[15] Zwick M., Gerdts M. and Stütz P. Sensor-Model-Based Trajectory Optimization for UAVs to Enhance Detection Performance: An Optimal Control Approach and Experimental Results. *Sensors 2023*, 2, 664, 2023, https://doi.org/10.3390/s23020664.

[16] Bronstein I. N., Semendjaev K. A., Musiol G. And Mühlig H. *Taschenbuch der Mathematik*, 7. Aufl. Frankfurt am Main: Deutsch, 2008.

[17] Beard R. W. and McLain T. W. *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press. 2012