Aircraft Inspection Using Unmanned Aerial Vehicles

Randa Almadhoun, Tarek Taha, Lakmal Seneviratne, Jorge Dias, Guowei Cai

***Abstract*— In this paper, we propose a coverage planning algorithm for inspecting an aircraft, using an Unmanned Ariel Vehicle (UAV). Inspecting structures (e.g. bridges, buildings, ships, wind turbines, aircrafts) is considered a hard task for humans to perform, and of critical nature since missing any detail could affect the structure's performance and integrity. Additionally, structure inspection is a time and resource intensive task that should be performed as efficiently and accurately as possible. In this paper we introduce a search space coverage path planner (SSCPP) with a heuristic reward function that exploits our knowledge of the structure model, and the UAV's onboard sensors' models to generate resolution optimal paths that maximizes coverage. The proposed method follows a model based coverage path planning approach to generate an optimized path that passes through a set of admissible waypoints to fully cover a complex structure. The algorithm predicts the coverage percentage by using an existing model of the complex structure as a reference. A set of experiments were conducted in a simulated environment to test the validity of the proposed algorithm.**

I. Introduction

The application of robotic inspection involves many robotic application components employing different types of robotic systems including: Unmanned Arial Vehicles (UAVs); Unmanned Ground Vehicles (UGVs); and maritime robots, all of which could be utilized for different inspection operations. The usage of UAVs particularly provides the flexibility of capturing visual information of the structure regions that are hard to reach, thus simplifying the reconstruction process which is essential for inspection. Inspecting large complex structures is particularly important in applications that require maintenance, fault traceability, anomaly and defects detection, and model digitizing. The presence of anomalies affects the performance and serviceability of complex structures, especially if these structures involve human lives such as an aircraft.

Technically, inspecting structures requires various robotic capabilities such as: localization in the environment where the structure exists; path planning and navigation in order to compute a set of achievable routes; sensing and perception in order to gather information about the structure from different viewpoints along the route. As such, it is important to equip the robot with intelligent sensing capabilities that enhance the quality of the information gathered in order to reconstruct, and inspect the structure of interest accurately.

Different research approaches have been followed in the past to perform inspection depending on the environment, the shape of the structure, and the level of the required details. The two main challenging research topics related to inspection are coverage path planning, and workspace 3D reconstruction. Coverage path planning is the process of computing a feasible path encapsulating a set of waypoints through which the robot must pass in order to completely scan the structure of interest. The two main coverage path planning categories include model based, and non-model based categories [1], [2]. The proposed algorithm follows a model based coverage path planning approach by which a set of waypoints is generated based on an existing reference model of the structure. The main aim of the model-based algorithms is to provide a set of waypoints that explores a structure so that every area of the structure is visible. Model based planning algorithms are further categorized into three categories based on the information embedded in the model of the structure including: set theory methods, graph theory methods, and computational geometry methods.

A number of structure-inspection algorithms that are suitable for robotics systems have been documented in the literature. In [3]–[6], an optimized path consisting of a set of stationary views providing full coverage of a ship hull was obtained by: using a polygonal mesh of the ship hull, and solving a set cover problem (SCP) that generates a redundant roadmap consisting of the waypoints. The path generation was preformed by solving a Travelling Salesman Problem (TSP) with lazy collision checking using Christofides approximation [7] and chained Lin-Kernighan improvement heuristic (LKH) [8]. A sampling based improvement using a modified Rapidly Exploring Random Tree algorithm (RRT*) [9] was used to reduce the path length. In [10], different search based algorithms including greedy variants and set cover with TSP algorithms were used with a known map in order to generate a set of sequenced waypoints that provide full coverage. It was found that using set cover method with TSP provides the best set of waypoints in terms of computation time and path cost [10]. Another model-based planning algorithm related to computational geometry methods was used in [11] in which a triangular mesh of the desired structure was utilized to determine the set of waypoints with the best configurations by solving it as an Art Gallery Problem (AGP) and connecting the waypoints by solving TSP using LKH. Additionally, Alexis et. al [12] proposed a Uniform Coverage of 3D Structures Inspection Path Planner (UC3D-IPP) that generates inspection paths by computing viewpoints and solving a TSP. An iterative strategy is used to improve the generated inspection path utilizing different remeshing techniques.

Moreover, a set of triangular meshes of objects of interest

Randa Almadhoun, Tarek Taha, Lakmal Seneviratne, Jorge Dias and Guowei Cai are with Khalifa University of Science Technology and Research, Abu Dhabi, UAE, e-mail: {randa.almadhoun,tarek.taha,guowei.cai,lakmal.seneviratne}@kustar.ac.ae

were used in [13] in order to generate motion planning roadmap and coverage spaces, and find the inspection path using self-organizing neural network. Heng et. al [14] proposed a coverage and exploration algorithm that explores cluttered unknown environments by choosing goals that maximizes information gain and coverage in real-time using a Micro Arial Vehicle (MAV). The path was generated by solving a submodular orienteering problem, which was approximated to a modular problem, then an approximate solution was computed using Gurobi Optimizer [15]. The generated path is not the shortest path but it provides the maximum coverage since it incorporates a variety of yaw angles.

Due to the size, geometric complexity, and the application criticality (inspection) of aircraft structures, we propose an algorithm that explicitly targets the coverage, and accuracy requirements during aircraft inspection applications. The proposed work consists of three main components: viewpoints generation, path planning, and coverage evaluation. Our proposed SSCPP algorithm was developed to integrate the sensors Field of View (FOV), range limitations, and measurement errors in order to generate an optimized path encapsulating viewpoints that achieves the maximum coverage. We then evaluate the achieved coverage percentage, and the produced model resolution through these set of viewpoints. Unlike other approaches, the proposed algorithm is resolution complete [16] that generates optimal inspection paths which provides efficient 3D reconstruction of complex shapes with certain coverage guarantees in addition to an estimation of resolution accuracy.

An overview of our proposed algorithm is presented in section II, followed by a description of each of the main components of the algorithm including: viewpoints generation in section II-A, coverage path planning in section II-B, and coverage evaluation in section II-C. The simulated experiments used to verify the proposed algorithm are presented in section III. Finally, we draw our conclusions, and propose future enhancements in section IV.

II. SSCPP Algorithm

The proposed SSCPP algorithm was developed to utilize an existing mesh model of the complex structure and the models of the sensors mounted on the UAV. The algorithm consists of three components: viewpoints generation, coverage path planning, and coverage evaluation, each of which will be explained in details in the next sections. Algorithm 1 presents an overview of the proposed coverage planning procedure including the three components.

A. viewpoints Generation

Viewpoints generation is the process of generating a set of viewpoints from which the structure of interest is visible. Our viewpoints generation method starts by discretizing the structure's workspace using a specific grid resolution which generates a set of sample viewpoints defining an xyz position in 3D space. Next, an orientation based discretization step is performed on each position sample to produce a set of yaw angle orientations that the UAV can conduct inspection from. The generated viewpoints are represented by xyz coordinates and ψ yaw angle. Each sensor viewpoint is then generated by applying a 4×4 transformation matrix that defines the sensor location with respect to the UAV body frame. Figure 1 shows an example of discretizing the orientations by $\pi/4$, and generating the corresponding sensor viewpoints. The sensor is placed on the UAV with a translation of $6cm$ and a rotation of $5.5°$ around the y axis.

These sample viewpoints are then filtered using collision, distance and coverage based filtering approach. Collision based filtering eliminates the sample viewpoints that are inside the model or collides with the model. Distance based filtering keeps the sample viewpoints available within a defined distance from the structure model determined based

Algorithm 1: Coverage Planning Overview

Input : structure model, 3D grid resolution, angular resolution, sensor FOV and range limitations

1 Discretize workspace into a cubic grid with input resolution.
2 Discretize orientations at each grid with input angular resolution.
3 Preform transformation to generate the sensor viewpoints at each waypoints sample (position and angle).
4 Filter out the grid samples according to the sensor range limitations.
5 Perform path planning and trajectory generation using the filtered set of viewpoints.
6 Navigate through the waypoints, and perform 3D reconstruction
7 **foreach** *viewpoint in the generated path* **do**
8 Find the occlusion cull between the FOV and the model.
9 Add the occlusion cull to the predicted model.
10 **end**
11 Confirm model coverage completeness: predicted vs constructed

Output: waypoint trajectory, 3D reconstructed model, coverage percentage, model accuracy

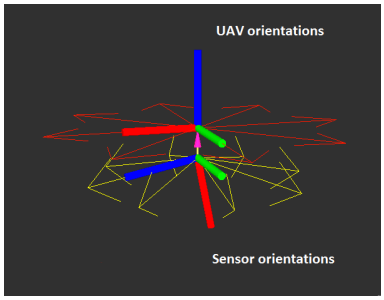


Fig. 1: Yaw orientation samples for the UAV body frame is shown in red, and the corresponding sensor orientation is shown in yellow

on the sensor's minimum and maximum effective range. To filter the viewpoints further, at each sensor viewpoint, the visible surface is extracted by performing frustum culling, which extracts the structure part that lies inside the sensor FOV frustum, and occlusion culling, which extracts the visible surface from the extracted frustum. Viewpoints that provide no coverage, as evident by the occlusion culling output, are then filtered out. The occlusion culling algorithm is presented in Algorithm 2 and Figure 2 shows an illustration of this process. Figure 3 shows the filtered pointcloud set, and sensor viewpoint.

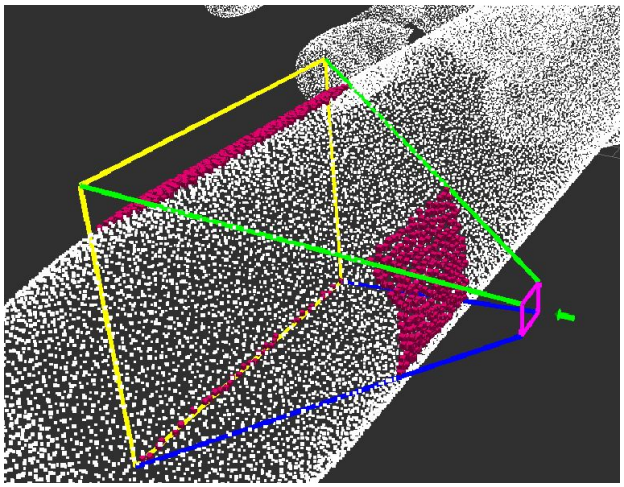
Algorithm 2: Occlusion Culling - RGBD Sensor

Input : sensor FOV, sensor origin O
Output: Visible surface point cloud
Identify the point cloud inside FOV frustum
Place the point cloud in a voxel grid
foreach *voxel in the voxel Grid* **do**
  Perform ray tracing from origin O to *voxel*
  if *voxel is not occluded by another voxel* **then**
    Store the *voxel* occluded free voxel point cloud
  end
end

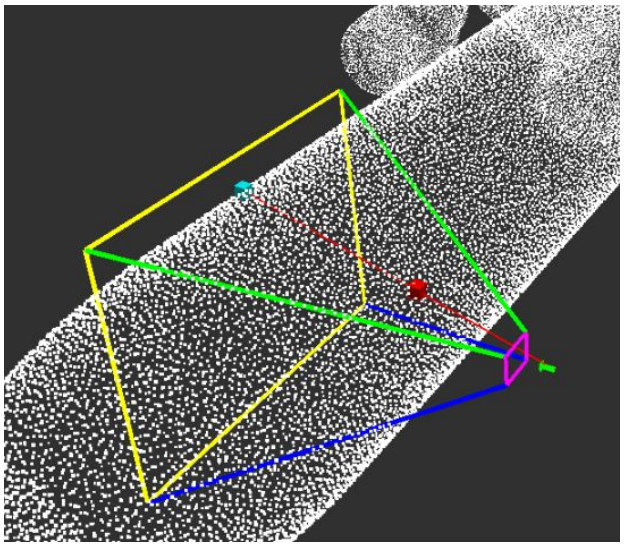
B. Coverage Path Planning

The next component of the algorithm is the coverage path planning. The discretized sample space at this stage consists of the set of filtered sample waypoints W, and the corresponding sensor viewpoints V. The search space is then generated by graphically connecting samples with their neighbors based on a pre-defined connection radius r as shown in Figure 4. The developed SSCPP, with a heuristic reward function, is then used to search this search space for an optimised path that achieves the desired coverage percentage. SSCPP is resolution complete graph search heuristic algorithm which increases the possibility of obtaining an optimal path as the discretization resolution increases. The heuristic reward function R minimizes travel distance δd and turns δa, and maximizes the coverage C computed at each step. The reward function is defined in equation (1), the first term is inversely proportional to the distance traveled, the longer the distance, the lower the ratio of the coverage C contribution to the reward. When the next waypoint involves only a rotation, then the reward is proportional to δa as shown in the second part of the equation. Algorithm 3 presents the search space coverage path planning algorithm which takes the set of waypoints and viewpoints, the connection radius r, the target coverage percentage, and the coverage tolerance as inputs and outputs the trajectory that will provide the target coverage.

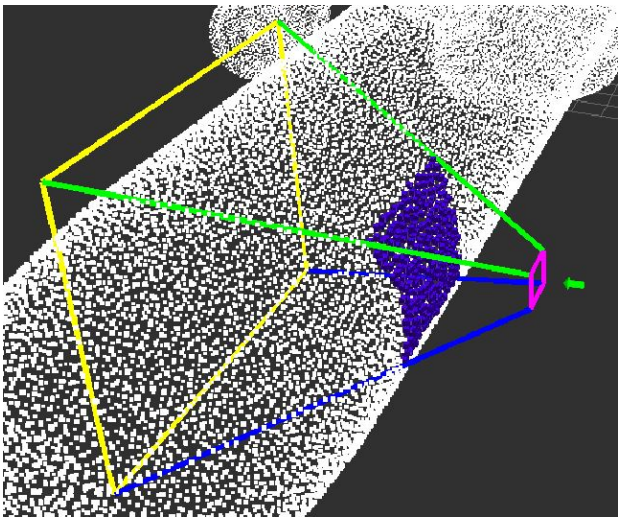
$$R = \begin{cases} \frac{1}{\delta d} \times C, & \text{if } \delta d > 0 \\ (1 - \frac{\delta a}{2\pi}) \times C, & \text{if } \delta d = 0 \end{cases} \quad (1)$$



(a) The visualization of the frustum cull (purple) of an aircraft model (white) from one sensor viewpoint



(b) The blue target voxel is occluded by the red voxel accros the red ray



(c) The visualization of the visible surface (blue) of an aircraft model (white) and the FOV planes from one viewpoint

Fig. 2: Visible surface extraction using occlusion culling

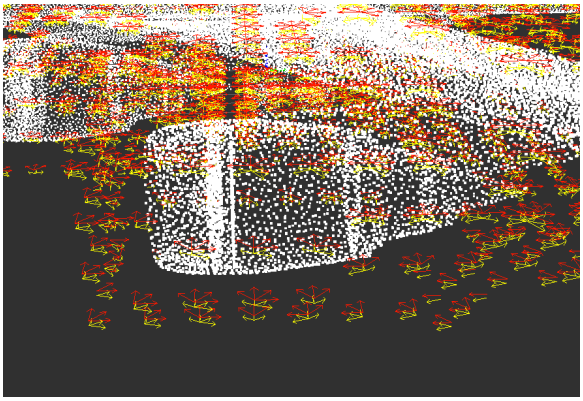


Fig. 3: Visualization of a set of filtered waypoints in red and the corresponding sensor viewpoints in yellow

Algorithm 3: Search Space Coverage Path Planner - SSCPP

Input : set of waypoints W and viewpoints V, connection radius r, starting pose $start$, target coverage percentage and coverage tolerance t.
Output: coverage path $C.P(s|S)$, coverage percentage

$W, V \subset \mathbb{R}^3$
Generate the search space nodes $S \leftarrow W, V$
Generate graph network by connecting S nodes based on radius r
Find the closest search space node to $start$
Initialize Open List O and Closed List C
Add the start node to O
while $s \in S$ *&* *target coverage not achieved* **do**
  Pick s_{best} from O such that $f(s_{best}) \geq f(s)$, $\forall s \in O$
  Remove s_{best} from O and add to C
  Expand s_{best}: for all $n \in Star(s_{best})$ & $n \notin C$
  foreach n *in* $Star(s_{best})$ **do**
    compute the extra coverage at n_v
    compute the distance between s_w & n_w
    compute the angle difference between s_w & n_w
    compute $R(n)$
    if $n \in O$ **then**
      $p \leftarrow n$
      if $R(p) > R(n)$ **then**
        remove n from O
      end
      else
        remove p from O
      end
    end
    else
      add n to O
    end
  end
  $C.P(s|S) \leftarrow$ s
end

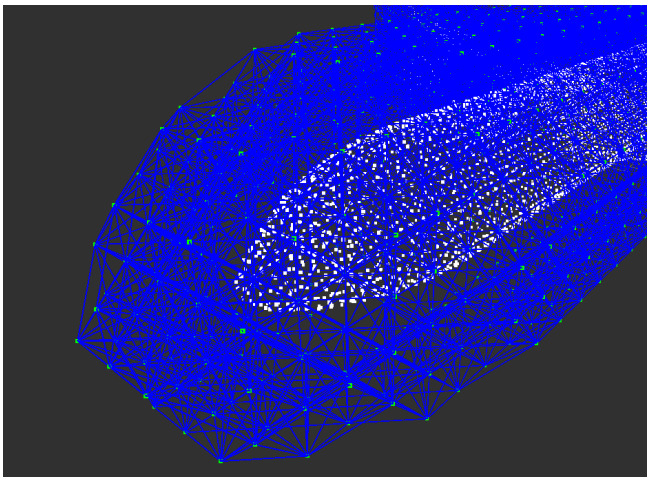


Fig. 4: Visualization of the connections generated from the search space samples applying a specific connection radius(3m)

C. Coverage Evaluation

The completeness of the coverage planning algorithm is assessed by quantifying the percentage of the covered volume of the structure compared to the predicted 3D structure volume across the generated path. The covered volume is the actual volume of the surface constructed by following the trajectory produced by the coverage path planning algorithm, and collecting data along the path. The predicted volume, however, is measured by performing frustum culling on the reference model at each trajectory waypoint, and accumulating the volume along the trajectory. Each of the covered and the original volumes of the structure is represented by a grid of voxels. These voxel grids are then used to calculate the coverage percentage as described in (2).

$$Coverage\ \% = \frac{Covered\ Volume\ _{Voxel\ Grid}}{Original\ Volume\ _{Voxel\ Grid}} \times 100 \quad (2)$$

This component was presented in literature as a critical part of the planning algorithm iterations, and as a significant performance criteria. The coverage completeness of the model-based algorithms can be computed directly based on the baseline used in the planning procedure as we preformed. A similar approach was adapted in [17], where the area coverage was quantified using a Monte-Carlo process by which the percentage is computed using the ratio of the number of sampled points that belongs to any of the sensed areas by the UAVs to the total number of the sampled points.

III. Experiments and Results

A. Experimental Setup

In order to evaluate the proposed algorithm, a set of experiments were performed in a realistic robot simulator Gazebo [18]. A A340 aircraft mesh model was used to represent a complex aircraft structure that contains 32496 triangular faces. Software-in-the-loop (SITL) simulated experiments were conducted using an Iris quadrotor platform equipped with RGBD sensor. The main critical parameters of

the algorithm include the sensor FOV, grid resolution, sensor range, target coverage percentage, and tolerance to the target coverage percentage.

B. Experimental Results

An experiment was designed to check the predicted coverage percentage and compare it with the real coverage achieved. In this experiment, the quadrotor is assumed to carry a RGBD sensor mounted at $5.5°$ with a FOV of [58H,45V]° and a maximum depth range of $7m$. The 4×4 matrix presented in (3) is the sensor transformation matrix with respect to the quadrotor body frame used in the designed experiment. The proposed approach was used with a grid resolution of $1.5m$, a sensor distance of (2 to 4)m from the model, a connection radius of $3m$, a set of different target coverage percentages and a target tolerance of 1%.

$$\begin{bmatrix} 0.995401 & 0 & 0.0957973 & 0 \\ 0 & 1 & 0 & 0 \\ -0.0957973 & 0 & 0.995401 & -0.06 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

We repeated the same experiment with various target coverage percentages, and we evaluated the results based on metrics that include the path distance, the target coverage percentage and the end coverage percentage. The results of the experiments are summarized in Table I, and Figure 5 illustrates the path generated to achieve 90% coverage. Figure 6 illustrates the sensor FOV along some sampled viewpoints along the generated coverage path. As evident by our results, our method generated feasible path trajectories that consist of 656 waypoints to achieve the desired 90% coverage percentage. It's worth mentioning that due to the sensor mounting position, proximity of the aircraft to the ground, and the fact that only stable horizontal hovering of the quadrotor was considered, the maximum achievable coverage percentage is 92.1922% for this particular setup (measured by considering all the samples in the search space).

We also compared our method to the approach described in [11] using the same model, and the same experimental setup. This approach requires the sensor FOV, minimum and maximum distance from the model, sensor pitch mounting angle, incidence angle, and the number of iterations. These parameter were set to: sensor FOV = [120H,120V]° , minimum viewpoint distance of $1m$, maximum viewpoint distance of $20m$, camera sensor mounting angle of $5.5°$, incidence angle of 30° and 10 optimization iterations. The mesh model had to be simplified with this approach to include 10248 triangular faces instead of the original 32496. This is due to the fact that, this complex aircraft model contains a lot of occluded regions with a dense set of triangular faces, and that this approach depends on generating a viewpoint for each triangular face, making it difficult to generate a path passing through all these viewpoints.

This approach generated the path shown in Figure 7, consisting of 10248 viewpoints. As mentioned previously, an AGP is used in this approach to generate a set of viewpoints corresponding to each triangular face of the mesh based on the visibility of this face. This approach targets the visibility of all the faces of the mesh model through a comprehensive set of viewpoints.

For large complex geometrical structures such as the aircraft mesh, it is shown that even with the simplification performed on the model, significantly larger set of viewpoints was generated, making it hard to follow such a complex path to preform the inspection. The length of the generated path using this approach is $6758.06m$ which is considered a very long path compared to $1182.99m$ and 656 viewpoints generated using our proposed approach. Generating a short inspection path is of great significance since the UAVs have limited flight time.

TABLE I: Scenario 1 Summary of experiments results

Grid resolution = 1.5m, Effective sensor range = 2-4m, Connection radius = 3m, Tolerance to coverage percentage = 1%		
Target Coverage %	**Path Distance**	**End Coverage**
20%	96.8714m	19.023%
50%	386.462m	49.262%
70%	490.545m	69.218%
90%	1182.99m	89.026%

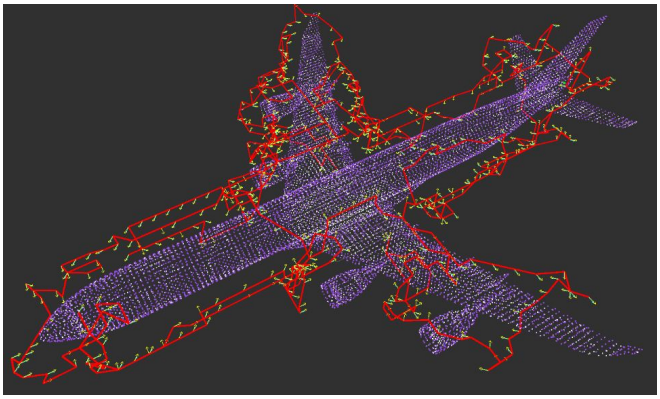
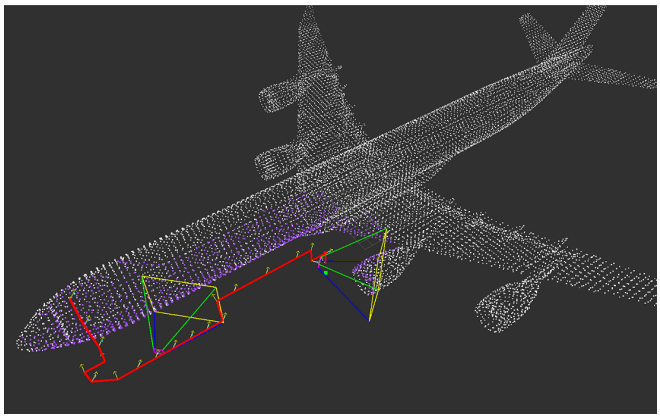
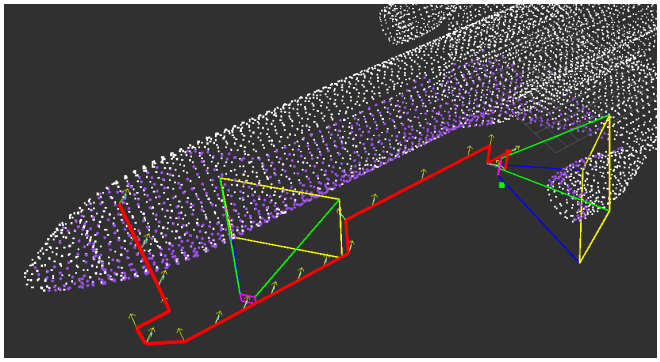


Fig. 5: Path generated to produce 90% coverage using our proposed method. The generated coverage path is shown in red, and the selected waypoints and their corresponding viewpoints in yellow. The original model pointcloud is shown in white and the covered part is shown in purple

Furthermore, the accuracy of the model generated by our approach was evaluated based on the RGBD sensor model. The accuracy of the 3D constructed model was evaluated by computing the standard deviation of error in depth Z at each point in the point cloud at each viewpoint in the generated coverage path following equation (4) presented in [19]. The values of $\frac{m}{f_b}$ and σ_d were computed in [19] as 2.85×10^{-5} and $\frac{1}{2}$ pixel by calibrating a Kinect RGBD sensor. The evaluation of accuracy helps in identifying and



(a) Full view



(b) Closeup view

Fig. 6: Sample path generated to produce 10% coverage. The path is displayed in red and the selected waypoints and their corresponding viewpoints in yellow. The original model pointcloud is shown in white and the covered part is shown in purple. The FOV for one of these waypoints is shown, where the top plane is shown in green, the bottom plane in blue, the near plane in red, and the Far plane in yellow

improving the regions with the lowest accuracy in order to provide highly accurate 3D reconstruction. Figure 8 shows an illustration of the results of accuracy computations visualized as a yellow color gradient that ranges from the lightest (highest accuracy) to the darkest ranges (lowest accuracy). It provides an accuracy indication of the generated model taking into consideration the RGBD sensor noise model.

$$\sigma_z = (\frac{m}{f_b}) Z^2 \sigma_d \tag{4}$$

One of the generated coverage paths that targets a coverage of 90% was evaluated in simulation using Gazebo SITL. Figure 9 shows the reconstructed 3D model as a result of following that path. The 3D reconstruction was performed using Real Time Appearance Based Mapping (RTAB) [20] and Octomap [21]. The resulting model demonstrates that the generated path are traversable, and generate the desired coverage percentage.

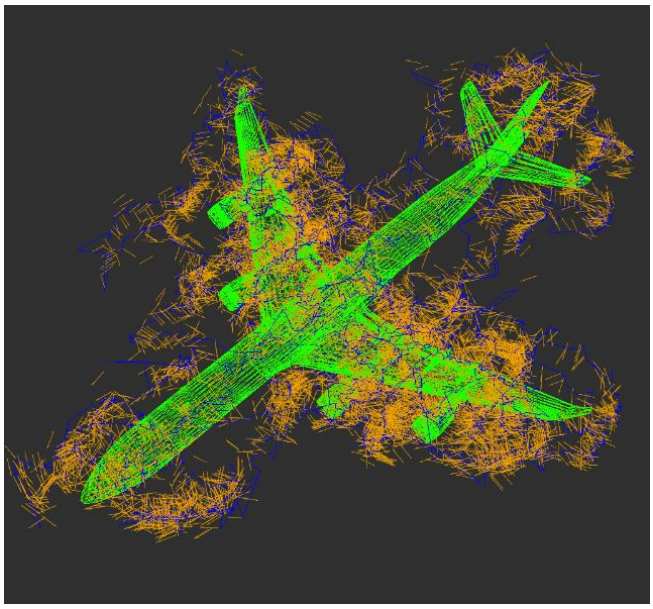


Fig. 7: Visualization of the coverage path generated using the approach of [11]. Orange arrows represent the viewpoints, and blue line segments represent the generated path.

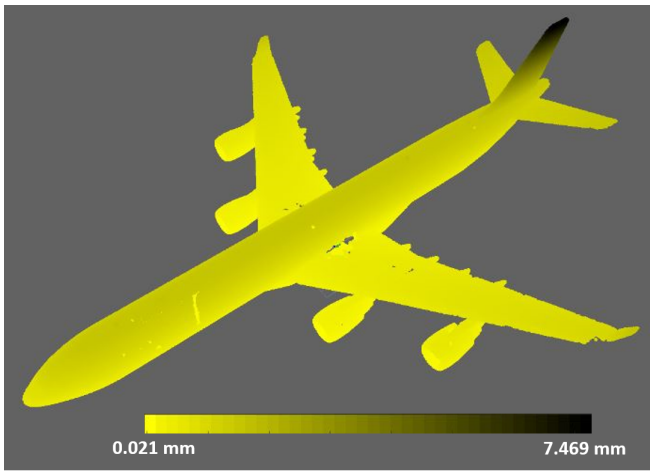
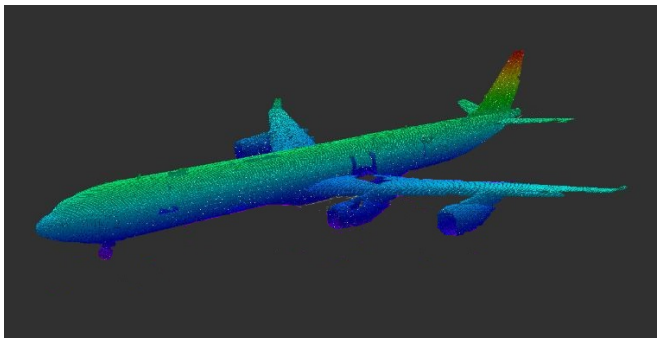


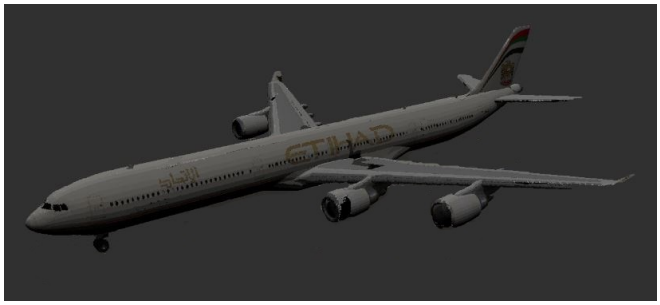
Fig. 8: Color gradient model representing the accuracy across the 90% path generated by our approach ranging from the lightest (highest accuracy) to the darkest (lowest accuracy). A colormap shows the lowest and highest standard deviation of depth and their corresponding color range

IV. Conclusion

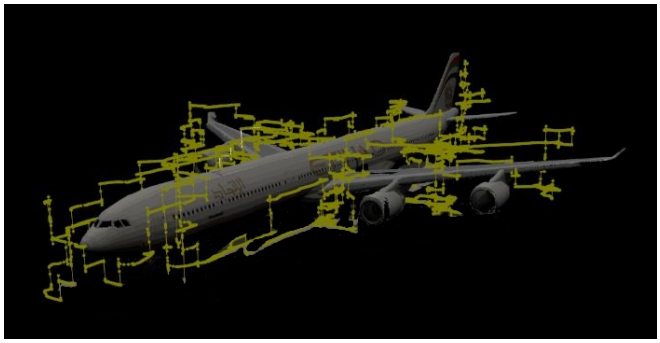
In this paper we introduced a coverage path planning approach that facilitates the inspection of large, geometrically complex structures using a quadrotor platform. We integrated sensor models to generate a coverage path offline and provide a prediction of the coverage percentage. The algorithm and the overall approach was verified using a realistic robot simulator where a quadrotor follows the generated coverage path and generates a 3D reconstructed model. Future work will focus on integrating the sensor accuracy directly in the planning heuristics to generate coverage paths that not



(a) 3D reconstruction using Octomap



(b) 3D reconstruction using RTAB



(c) The path followed by the quadrotor during the 3D reconstruction

Fig. 9: 3D reconstruction models generated by following the 90% coverage path generated in 1

only guarantee coverage to a certain percentage, but also guarantee that the model accuracy will be within a certain resolution.

REFERENCES

[1] W. R. Scott, "Model-based view planning," *Machine Vision and Applications*, vol. 20, no. 1, pp. 47–69, 2009.

[2] W. R. Scott, G. Roth, and J.-F. Rivest, "View planning for automated three-dimensional object reconstruction and inspection," *ACM Computing Surveys*, vol. 35, no. 1, pp. 64–96, 2003.

[3] B. Englot and F. Hover, "Inspection planning for sensor coverage of 3D marine structures," *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, pp. 4412–4417, 2010.

[4] B. Englot and F. S. Hover, "Sampling-based sweep planning to exploit local planarity in the inspection of complex 3D structures," *IEEE International Conference on Intelligent Robots and Systems*, pp. 4456–4463, 2012.

[5] B. Englot and F. Hover, "Sampling-Based Coverage Path Planning for Inspection of Complex Structures." *Icaps*, pp. 29–37, 2012. [Online]. Available: http://www.aaai.org/ocs/index.php/icaps/icaps12/paper/download/4728/4711

[6] F. S. Hover, R. M. Eustice, a. Kim, B. J. Englot, H. Johannsson, M. Kaess, and J. J. Leonard, "Advanced Perception, Navigation and Planning for Autonomous In-Water Ship Hull Inspection," *Intl. J. of Robotics Research*, vol. 31, no. 12, pp. 1445–1464, 2012.

[7] N. Christofides, "Worst-Case Analysis of a New Heuristic Prepared for the Travelling Salesman Problem," no. February, 1976.

[8] K. Helsgaun, "Effective implementation of the Lin-Kernighan traveling salesman heuristic," *European Journal of Operational Research*, vol. 126, no. 1, pp. 106–130, 2000.

[9] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[10] C. Dornhege, A. Kleiner, and A. Kolling, "Coverage search in 3D," *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics, SSRR 2013*, 2013.

[11] A. Bircher, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, and R. Siegwart, "Structural Inspection Path Planning via Iterative Viewpoint Resampling with Application to Aerial Robotics," pp. 6423–6430, 2015.

[12] K. Alexis, C. Papachristos, R. Siegwart, and A. Tzes, "Uniform coverage structural inspection path-planning for micro aerial vehicles," *IEEE International Symposium on Intelligent Control - Proceedings*, vol. 2015-Octob, pp. 59–64, 2015.

[13] P. Janousek and J. Faigl, "Speeding up coverage queries in 3D multi-goal path planning," *Proceedings - IEEE International Conference on Robotics and Automation*, no. 1, pp. 5082–5087, 2013.

[14] L. Heng, A. Gotovos, A. Krause, and M. Pollefeys, "Efficient Visual Exploration and Coverage with a Micro Aerial Vehicle in Unknown Environments," *in IEEE International Conference on Robotics and Automation*, 2015.

[15] (2014) Gurobi optimizer reference manual. [Online]. Available: http://www.gurobi.com/

[16] Y. Bestaoui Sebbane, *Planning and Decision Making for Aerial Robots*, 2014, vol. 71.

[17] A. Wallar, E. Plaku, and D. A. Sofge, "A Planner for Autonomous Risk-Sensitive Coverage (PARC OV) by a Team of Unmanned Aerial Vehicles," *in IEEE Symposium on Swarm Intelligence (SIS)*, 2014.

[18] Gazebo. [Online]. Available: http://gazebosim.org/

[19] K. Khoshelham and S. O. Elberink, "Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications," *Sensors*, vol. 12, no. 12, pp. 1437–1454, 2012. [Online]. Available: http://www.mdpi.com/1424-8220/12/2/1437/

[20] M. Labbe and F. Michaud, "Online Global Loop Closure Detection for Large-Scale Multi-Session Graph-Based SLAM," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2014, pp. 2661–2666.

[21] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.