Automating Aircraft Scanning For Inspection With A UAV And Reinforcement Learning Technique

Yufeng Sun (✉ sunyf@mail.uc.edu)

University of Cincinnati <https://orcid.org/0000-0003-2508-4207>

Ou Ma

University of Cincinnati

Research Article

Keywords: Aircraft scanning, Path planning, UAV, Reinforcement learning

Posted Date: December 28th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-1193469/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Automating Aircraft Scanning for Inspection with a UAV and Reinforcement Learning Technique

Yufeng Sun^{1*} and Ou Ma¹

¹Department of Aerospace Engineering and Engineering Mechanics, University of Cincinnati, 2600 Clifton Ave, Cincinnati, 45221, Ohio, USA.

*Corresponding author(s). E-mail(s): sunyf@mail.uc.edu;
Contributing authors: maou@ucmail.uc.edu;

Abstract

Visual inspections of aircraft exterior surface are usually required in aircraft maintenance routine. It becomes a trend to use mobile robots equipped with sensors to perform automatic inspections as a replacement of manual inspections which are time-consuming and error-prone. The sensed data such as images and point cloud can be used for further defect characterization leveraging the power of machine learning and data science. In such a robotic inspection procedure, a precise digital model of the aircraft is required for planning the inspection path, however, the original CAD model of the aircraft is often inaccessible to aircraft maintenance shops. Thus, sensors such as 3D Laser scanners and RGB-D (Red, Green, Blue, and Depth) cameras are used because of their capability of generating a 3D model of an interested object in an efficient manner. This paper presents a two-stage approach of automating aircraft scanning with a UAV (Unmanned Aerial Vehicle) equipped with an RGB-D camera for reconstructing a digital replica of the aircraft when its original CAD model is not available. In the first stage, the UAV-camera system follows a predefined path to quickly scan the aircraft and generate a coarse model of the aircraft. Then, a full-coverage scanning path is computed based on the coarse model of the aircraft. In the second stage, the UAV-Camera system follows the computed path to closely scan the aircraft for generating a dense and precise model of the aircraft. We solved the Coverage Path Planning (CPP) problem for the aircraft

scanning using Monte Carlo Tree Search (MCTS) which is a reinforcement learning technique. We also implemented the Max-Min Ant System (MMAS) strategy, a population-based optimization algorithm, to solve the CPP problem and demonstrate the effectiveness of our approach.

Keywords: Aircraft scanning, Path planning, UAV, Reinforcement learning

1 Introduction

Visual inspections are usually required for ensuring the airworthiness of an aircraft in maintenance routine [1]. Traditional way of visual inspection is time-consuming and error-prone as it requires human inspectors to visually inspect the entire aircraft exterior body to identify possible anomalies such as dents, cracks, leaking, broken or missing parts, etc., and manually measure the parameters of the found defects. Robotics-based inspection methods are being developed for aircraft inspection to reduce the workload, time, and cost for defect detection. These methods use UGVs [2] or UAVs [3] [4] equipped with laser scanners or cameras to perform Non-Destructive Inspection (NDI) around the aircraft. The inspection vehicles either follow human operator's guides or automatically navigate themselves based on an existing map of the environment to acquire images or other sensed data of the aircraft. These images or data can be used for defect detection and analysis leveraging the power of machine learning techniques. The efficiency of aircraft visual inspection can be significantly improved by automating the robotic inspection procedure. Research of using a robotic system, such as a UAV, to automatically inspect large structures [5], specifically an aircraft [6] solves the key problem: the Coverage Path Planning (CPP) problem for automatic robotic inspection, with different approaches based on heuristic optimization algorithms or Reinforcement Learning (RL) techniques. These solutions require a pre-existing CAD model of the target structure [7] for computing the inspection viewpoints. Unfortunately, for aircraft inspection, such a CAD model is usually inaccessible to a Maintenance, Repair, and Operation (MRO) shop due to reasons such as old aircraft or non-accessibility of digital models. Therefore, 3D laser scanners or RGB-D cameras can be used to generate a digital model of the target aircraft because of their efficient manner of 3D model reconstruction. However, manually scanning a large aircraft shares the same drawbacks as visually inspecting the aircraft. Alternatively, UAVs equipped with RGB-D cameras can be used to accelerate the aircraft scanning process because UAVs have been successfully applied in many other applications [8] such as surveying, mapping, and surveillance. In addition, it is possible for small MROs to effectively use UAV technology and apply the "Smart MRO" principle in the pre-flight inspection of aircraft [9].

Aiming to automating the aircraft scanning and inspection, we developed a two-stage approach of aircraft scanning using a UAV equipped with an RGB-D

camera in this paper based on our previous research [10]. Given a static MRO environment with an indoor GPS system, the UAV-Camera system follows a predefined scanning path to quickly reconstruct a coarse model of the aircraft in the first stage. However, the low-resolution and low-accuracy model is not qualified for guiding the UAV-Camera system to perform safe and detailed inspection task. The UAV-Camera system needs to closely scan the aircraft for the purpose of reconstructing a dense and precise digital model in the second stage. The resulted model then can be used for aircraft inspection, maintenance, or even as a key component of a digital twins to support the operational prediction and prognosis of the aircraft [11]. In order to automate the scanning in the second stage, an "optimal" scanning path for fully covering the area of interest of the target aircraft with short flying distance must be computed. We solved this CPP problem using a Monte Carlo Tree Search (MCTS) [12] algorithm which is a RL technique. We also implemented a population-based heuristic optimization algorithm named Max-Min Ant System (MMAS) [13] to demonstrate the effectiveness of our two-stage approach of aircraft scanning.

The rest of this paper is structured as follows. Section 2 introduces the existing methods for aircraft scanning and inspection including the solutions of the CPP problem. Section 3 describes the proposed two-stage approach of aircraft scanning. Section 4 provides a simulation of solving the CPP problem in simplified scenarios with both the MMAS and the MCTS algorithms. Section 5 presents digital experiments with an example aircraft. Finally a conclusion is drawn and future works are discussed to end of our paper.

2 Related Work

UAVs equipped with cameras were employed for collecting image data that can be used to create detailed 3D models of an Airbus fleet [14]. Such solutions were also adopted in small MRO companies to develop procedures for providing inspection solution for airports operations. In fact, an aircraft is large and its shape is complex, the path planning problem, as a key task of automated robotic scanning and inspection using the UAV-Camera system, remains challenging. This problem is referred as the CPP problem, a non-deterministic polynomial time (NP)-hard problem, which is for computing an "optimal" path in terms of flying distance, flying time, or energy consumption of the UAV to fully cover the exterior area of interest of the target aircraft. Considering that aircraft scanning and inspection in MRO shops is usually in a static environment, the CPP problem can be solved offline based on an exist digital model of the aircraft. Many studies [15] [16] [17] tackled the CPP problem in two steps by solving the Set Covering Problem (SCP) following solving the Travelling Salesman Problem (TSP). The SCP algorithm finds the smallest set of viewpoints to cover the target area of interest. Typical SCP algorithms are greedy algorithm and its variants. The TSP algorithm finds the shortest path to visit all selected viewpoints. Population-based optimization algorithms

such as Genetic Algorithm (GA), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO) can be applied in the solutions. Instead of dividing the CPP problem into two sub-problems, other researchers studied one integrated solution for achieving multiple objectives which are maximizing coverage and accuracy, meanwhile minimizing flying distance, time, or energy consumption. Phung et al. [18] formulated the inspection path planning problem as an extended TSP in which both the coverage and obstacle avoidance were taken into account. They proposed an enhanced discrete PSO algorithm to solve the extended TSP for robotic inspection of large surfaces. Almandhoun et al. [6] developed the Adaptive Search Space Coverage Path Planner (ASS-CPP) algorithm for aircraft inspection using a graph search algorithm instead of TSP algorithms. They produced a discretized sample space which consists of a set of viewpoints and the sample space was used to generate the search space by graphically connecting samples with their neighbors. The algorithm also included a heuristic function that evaluates the expected Information Gain (IG) of the waypoints by updating a probabilistic volumetric map represented as an octree. This approach was extended by Silberberg et al. in [19] for developing a visual inspection system of military aircraft using a multi-rotor UAV and their analysis shown that multi-rotor UAVs are a viable inspection platform for military aircraft. More generally, the CPP problem of inspection by UAVs could be solved using grid-based methods, which guarantee a complete coverage of the target scenario by applying cellular decomposition in the area of interest and splitting the target free space into cells in order to simplify the coverage [20]. The grid-based maps were also leveraged in Kyaw et al.'s solution for optimizing the CPP problem in a large complex environments based on the TSP and deep RL [21]. They decomposed the environment to form a TSP problem and determined the solution to the TSP by training a recurrent neural network with long short term memory layers using RL.

Above studies for large structure inspection with a UAV-Camera system require a pre-existing mesh model of the structure for solving the CPP problem. However, for small or medium MRO shops, such model is often unavailable. This makes our approach of aircraft scanning and inspection unique because it does not need an existing mesh model and hence, beneficial to those businesses.

3 Methodology

3.1 UAV-Camera Scanning System

The UAV-Camera scanning system consists of a UAV and an RGB-D camera. The camera is installed on the UAV and can be tilted upward and downward with respect to the UAV. An indoor GPS system [22] is used for UAV localization due to its advantage of large measurement range, non-contact, and easy to setup. In addition, the indoor GPS system can achieve millimeter-level accuracy on tracking a slowly moving object [23], which makes it suitable for both stages of scanning. The architecture of the two-stage scanning system is shown in Figure 1. A control station is used for point cloud data alignment, candidate

viewpoints generation, coverage path planning, and UAV-Camera system control. The Robot Operating System (ROS) [24] is used for data communication between the control station and the UAV-Camera system.

Fig. 1: Architecture of two-stage aircraft scanning system

3.2 Coarse Model Acquisition

RGB-D cameras have limited field-of-view (FOV), thus, a plenty of data frames need to be aligned together to form a global model of the aircraft if the UAV works at a close distance to the aircraft. In order to reduce the time of generating a coarse model, the UAV is controlled to automatically fly along a predefined path with a long working distance so that less amount of image frames needs to be acquired. The position and orientation of the UAV-Camera system can be obtained through the indoor GPS system and sent to the control station in real-time manner. Therefore, a transformation matrix can be calculated for aligning the data captured by the RGB-D camera into a global model using

$${}^wP = {}^wT_c {}^cP \quad (1)$$

where cP is the vector of a 3D point in the camera coordinate frame, wP is the vector of the same 3D point in the global coordinate frame; and wT_c is the transformation matrix for mapping the point from the camera frame to the global frame, which can be calculated based on the position and orientation of the UAV-Camera system measured by the indoor GPS. The point cloud data frames are aligned together using Point Cloud Library [25]. Noisy data in each data frame is filtered by applying the sparse outlier removal method. Then, to simplify the large-scale point cloud, a voxel-grid based down sampling method is applied to the input data frame before aligning it with the global model. An example of the resulted coarse digital model in octree representation is shown in Figure 2.

Fig. 2: Octree representation of the coarse point cloud model of aircraft

3.3 Viewpoint Generation

To scan or inspect the aircraft, the camera must be placed at a specific pose which is also known as viewpoint, denoted as $[x, y, z, \alpha, \beta, \gamma]$. $[x, y, z]$ are the 3D position coordinates of the camera in the global coordinate frame and $[\alpha, \beta, \gamma]$ are the heading, tilt, and roll of the camera with respect to the vehicle frame correspondingly. This pose is determined by the characteristics of a camera such as FOV, and the safety requirement such as working distance to the aircraft exterior surface.

In order to generate sufficient viewpoints to cover the entire area of interest, a set of candidate viewpoints needs to be computed based on the camera's FOV and the safe working distance to the aircraft. Based on the octree representation of the coarse model, for each voxel, the surface normal of the voxel is estimated based on the points in the voxel. By offsetting the camera with the specified working distance along the voxel's normal, a candidate viewpoint can be determined, and its viewshed can then be computed by examining visible octree voxels using frustum culling method. Therefore, the generated candidate viewpoints are uniformly distributed over the area of interest for ensuring the full coverage. An example of the candidate viewpoint and its viewshed (visible voxels are marked in red cube) is shown in Figure 3.

3.4 Coverage Path planning

3.4.1 Max-Min Ant System

The CPP problem of aircraft scanning or inspection can be solved in two steps applying heuristic optimization algorithms. In the first step, we simply apply the greedy algorithm to solve the SCP for finding a minimum set of viewpoints that fully covers the area of interest of the target aircraft. The SCP in our paper is defined as: *Given a universe U of voxels that represents the target area*

Fig. 3: A viewpoint and its viewshed

of scanning or inspection, a collection of subset $S = \{S_1, S_2, \dots, S_m\}$, where every subset S_i represents i 'th viewpoint in total m candidate viewpoints and the subset has an associated cost. Find a minimum cost sub-collection of S that covers all voxels of U . The greedy algorithm may not provide an optimal solution all the time and hence, we run the algorithm multiple times and choose the best optimal solution as the one for the following TSP. In the second step, we apply the MMAS algorithm to solve the TSP to find an approximate shortest path for visiting all the viewpoints selected in the first step.

The MMAS is an ACO algorithm extended from ant systems, the first ACO algorithm, that was inspired by the foraging behavior of real ants. ACO algorithms make use of artificial ants which iteratively construct candidate solutions to a combinatorial optimization problem. The construction is guided by artificial pheromone trails and problem-dependent heuristic information. Typically, solution components which are used by more ants will receive a higher amount of pheromone and, hence, will more likely be used by the ants in future iterations of their strategy. The MMAS algorithm only allows the best solution found during an iteration or the run of the algorithm to update the pheromone trail for avoiding early search stagnation so to achieve the best performance of the algorithm. When applying the MMAS algorithm to the TSP, ants are randomly placed on chosen cities initially. To construct a tour, each ant moves to a "city" which has not been visited based on a probabilistic decision which is biased by the pheromone trail $\tau_{ij}(t)$ and a locally available heuristic information η_{ij} . An ant k currently located at city i chooses city j to go with a probability as

$$p_{ij}^k(t) = \frac{(\tau_{ij}(t))^\alpha \cdot (\eta_{ij})^\beta}{\sum_{l \in N_i^k} (\tau_{il}(t))^\alpha \cdot (\eta_{il})^\beta} \quad j \in N_i^k \quad (2)$$

where α and β determine the relative importance of the pheromone trail and the heuristic information; and N_i^k is the feasible neighborhood cities of ant k . Pheromone trails $\tau_{ij}(t)$ associated with each arc (i, j) are modified through pheromone trail evaporation and pheromone trail reinforcement by the ants, which is defined as

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij}^{best} \quad (3)$$

where parameter ρ (with $0 \leq \rho < 1$) is the trail persistence, and $\Delta\tau_{ij}^{best} = 1/f(s^{best})$ and $f(s^{best})$ denotes the solution cost of either the iteration-best or the global-best solution.

3.4.2 Monte Carlo Tree Search

Instead of dividing the CPP problem to two sub-problems and solving them separately, we propose a neighborhood viewpoints-based MCTS algorithm to solve the CPP problem for the aircraft scanning or inspection. For a viewpoint in the candidate viewpoints, its neighborhood viewpoints are the viewpoints that have overlapped voxels in their viewsheds. In order to find the best neighborhood viewpoints of a selected viewpoint, we score all the rest viewpoints based on the distance to the selected viewpoint and the number of overlapped voxels with the selected one. Then the lowest scored viewpoint appears to be the best neighborhood viewpoint as it is close to the selected viewpoint and does not share the same voxels. We can select one or more neighborhood viewpoints based on the score. These neighborhood viewpoints are the possible next viewpoints to visit during MCTS tree expanding.

Starting from a root node which represents the starting viewpoint, the MCTS algorithm selects a child node, representing a neighborhood viewpoint, to explore if the current node is not fully explored (A fully explored node means all its neighborhood viewpoints has been visited). Then, a rollout simulation based on randomly choosing a untried neighborhood viewpoint to visit is played until the desired coverage ratio is reached or the neighborhood viewpoints are all visited. The result of the simulation is a reward r considering the coverage ratio $r_{coverage}$, overlap ratio $r_{overlap}$, and total traveled distance d_{travel} . The reward is defined as

$$r = c_1 \cdot r_{coverage} - c_2 \cdot r_{overlap} - c_3 \cdot d_{travel} \quad (4)$$

where c_1 , c_2 , and c_3 are the control parameters for weighting the coverage ratio, the overlap ratio, and the total traveled distance. In this paper, c_1 is 100.0, c_2 is 10.0, and c_3 is 0.01. The reward of each simulation rollout is backpropagated to current child node and its parent nodes in the tree. Therefore, for one specified node, the best action of choosing a child node v to visit can be determined once all of its child nodes are explored, and the best child policy is defined as

$$v = \arg \max_{v \in V} (\sigma \cdot q_v + (1 - \sigma) \cdot \frac{R_v}{N_v}) \quad (5)$$

where V is the set of child nodes of the current node; σ is a hyper parameter for controlling the weighted combination of the average reward (by dividing the total reward R_v by the visited count N_v) and the maximum reward q_v of a simulation for a child node. We also apply the decay ϵ -greedy policy in child node selection to balance the exploration and exploitation during the run of the algorithm.

4 Simulation and Results

A simulation based on a 2D grid map was carried out to demonstrate the effectiveness of our methods for solving the CPP problem. The map is a 30.0 meters $\times$ 30.0 surface area discretized into 900 squared grids of one square-meter each. The UAV-Camera system works at a distance of 3.0 meters above the surface and the camera's FOV is (80° , 80°). There are 900 candidate viewpoints uniformly distributed on the map with one viewpoint above each grid.

Starting from a specific viewpoint, the SCP algorithm found 26 viewpoints among the total 900 candidate viewpoints for covering all the grids of the map (100% coverage), and the MMAS computed an approximate scanning path with a total length of 138.13 meters after 2000 simulation iterations, with the hyper-parameters α of 1, β of 2, and ρ of 0.95. The path is shown in Figure 4a. While the MCTS algorithm found 25 viewpoints for the complete coverage without any overlap among the viewsheds, which means that each grid is only visible to one viewpoint. However, the total length of the scanning path is 173.8 meters, which is about 25% longer than the one found using the combination of the SCP and the MMAS algorithms, as shown in Figure 4b. In the figures, the starting viewpoint is marked with a blue hexagon, the terminated viewpoint is marked with a big blue point, and the path connects all the viewpoints (small blue point) with straight lines. The overlapped grids are colored in red.

The impact of the hyper-parameter σ to the effectiveness of the MCTS algorithm was also evaluated, as shown in Figure 5. The MCTS algorithm finds a scanning path that completely covers the target area in 900 simulation iterations with the $\sigma=0.6$, which is the fastest trial.

Just as for the other learning-based methods, the combination of the SCP and the MMAS algorithms cannot guarantee to provide an optimal scanning path but it provides a full coverage suboptimal path although there are some overlaps among the viewsheds. They could be used in the scenario where the full coverage is important. Meanwhile, the MCTS algorithm can provide a full coverage scanning path with no overlaps among viewsheds in a problem with a small map requiring less amount of viewpoints to cover. However, our tests showed that the MCTS algorithm cannot provide a full coverage scanning path in a limited computing time (100,000 simulation iterations) if the map is large and requires more amount of viewpoints for fully covering it, as shown in Figure 6b. The map is 90.0 meters in length and 60.0 meters in width which contains 5400 grids and the size of each grid is 1.0 square-meter. To increase

(a) Scanning path found by the SCP and MMAS algorithms

(b) Scanning path found by the MCTS algorithm

Fig. 4: Examples of approximate optimal scanning paths

Fig. 5: Required number of iterations for different values of σ

the complexity of the problem, only the valid grid (colored in green) is counted in camera's viewshed, while the white grid is invalid for the viewpoint. The MCTS algorithm found the best scanning path consisting of 68 viewpoints with a total length of 576.93 meters, which covers 82.04% of the valid grids. For the same map, the combination of the SCP and MMAS algorithms found the best scanning path consisting of 91 viewpoints with the total length of 640.16 meters, as shown in Figure 6a.

(a) Scanning path found by the SCP and MMAS algorithms

(b) Scanning path found by the MCTS algorithm

Fig. 6: Examples of approximate optimal scanning paths on a large map

5 Digital Experiment

A digital experiment was also performed in Gazebo. We used the hector-quadrotor (a ROS package) [26] with an Intel Realsense D435 camera (whose characteristics are listed in Appendix A) for scanning the upper part of a Boeing 757, as shown in Figures 7a and 7b. The camera can be tilted from -90° (looking downward) to 90° (looking upward). We used the simulated pose of the quadrotor in Gazebo as the data from an indoor GPS system, which contains both position and orientation of the quadrotor.

(a) Boeing 757 mesh model

(b) Quadrotor with an RGB-D camera

Fig. 7: Digital experiment environment

Taking the upper part of the fuselage as an example, based on the acquired coarse model of the aircraft in the first stage, 250 candidate viewpoints were generated with the working distance of 3.0 meters and the voxel resolution of 1.0 meter, as shown in Figure 8.

The combination of the SCP algorithm and the MMAS algorithm found a full-coverage scanning path consisting 109 viewpoints with total length of 195.39 meters, as shown in Table 1 and Figure 9a. Meanwhile, by considering 6 neighborhood viewpoints in the search, the MCTS algorithm found a scanning path consisting of 55 viewpoints with hyper parameters specified in Appendix

Fig. 8: Candidate viewpoints of the upper part of the aircraft fuselage

B, and its total length is 152.37 meters. The scanning path covers 85.89% of the target area, as shown in Figure 9b.

Table 1: Computed scanning paths.

Algorithm	Coverage Ratio (%)	Viewpoint Count	Path Length (m)
SCP+MMAS	100	109	195.39
MCTS	85.89	55	152.37

(a) A scanning path found using the SCP and the MMAS algorithms

(b) A scanning path found using the MCTS algorithm

Fig. 9: Examples of scanning paths for the upper part of the aircraft

Increasing the number of neighborhood viewpoints will lead to more exploration in the MCTS algorithm, thus, require more computing resources. It is not a problem for an offline approach for computing the scanning path with a powerful computer. However, when the computing resource is limited or online

path computing is required, reducing the number of neighborhood viewpoints is a simple way to reduce the learning cost and generate a scanning path more quickly although the resulted path will be less optimal. In addition, increasing the terminated ϵ value will also result in more exploration in the MCTS algorithm but the computed path will be more optimal. When the target area of interest is large and requires more viewpoints to cover, the solution found by MCTS tends to be less optimal, therefore, reducing the scope of search is a practical strategy in real-world aircraft scanning or inspection.

By following the computed scanning paths, the UAV-Camera system can then perform close scanning to acquire a precise and dense digital model of the upper part of the fuselage, as shown in Figures 10a and 10b. The generated model following the scanning path computed by the combination of the SCP and MMAS algorithms covers the entire area of the upper part of the fuselage. But the model following the scanning path computed by MCTS algorithm contains few gaps. This could be improved by using a camera with larger FOV during the second-stage scanning. Another strategy to improve the MCTS coverage is to divide the large target area of interest into several small regions, and solve the CPP problem for each region, as the MCTS algorithm tends to perform better for search an area which requires less viewpoints to cover. An additional benefit of this strategy is that we could leverage multiple cheap UAV-Camera systems to simultaneously scan the aircraft, which further improves the efficiency of aircraft scanning.

Fig. 10: Dense models generated in the second-stage scanning

6 Conclusion and Future Work

We developed a two-stage approach of UAV-based 3D scanning system for scanning and creating an 3D digital model of an aircraft. The UAV is equipped with a consumer-grade RGB-D camera and works with an indoor GPS system, The technology can reduce human workload, eliminate human error, and improve work efficiency. It solved a CPP problem using a reinforcement learning algorithm and a heuristic optimization algorithm. Both algorithms can

effectively find an approximate optimal path for automated scanning. This two-stage approach has been verified by dynamic simulation for scanning the upper part of an aircraft fuselage to quickly generate a dense digital model which can be further used for visualization, inspection, and other aircraft maintenance purposes.

The scanning work can be easily extended to aircraft inspection which requires a full coverage inspection based on either a discrete or continuous path. Another future topic is to use a team of UAVs to perform scanning or inspection work, so that they can complement each other for team synergy and strength. The team can also be combination of UAVs and UGVs because the UGVs can work better than UAVs for scanning or inspecting the belly areas of an aircraft.

Declarations

- Funding
This research was supported by the funding from the Space Research Institute of the University of Cincinnati.
- Conflict of interest/Competing interests
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.
- Ethics approval
Not applicable.
- Consent to participate
Not applicable.
- Consent for publication
Not applicable.
- Availability of data and materials
Not applicable.
- Code availability
The software that supports this study is available in "<http://gazebosim.org/>". The code that support this study is available in "https://github.com/suneric/aircraft_scanning".
- Authors' contributions
Yufeng Sun and Ou Ma conceived of the presented idea. Yufeng Sun implemented the software code and performed the simulated experiment. Yufeng Sun took the lead in writing the manuscript. Ou Ma provided critical feedback and helped shape the research, analysis, and manuscript.

Appendix A Characteristics of RGB-D camera

See table A1.

Table A1: Sensor parameters.

Parameter	Value
FOV ($H \times V \times D$)	$69.4^\circ \times 42.5^\circ \times 77^\circ$
Resolution (pixels)	640×480
Frame Rate (fps)	30
Depth Range (m)	0.15 - 10

Appendix B Hyper parameters used in MCTS algorithm

See table B2.

Table B2: Hyper parameters in MCTS algorithm.

Parameter	Value
Reward Control Parameter σ	0.9
Terminated ϵ	0.1
ϵ Decay Rate	0.99999
Maximum Number of Iteration	1,000,000

References

- [1] Drury, C.G., Watson, J.: Good practices in visual inspection. Human factors in aviation maintenance-phase nine, progress report, FAA/Human Factors in Aviation Maintenance (2002)
- [2] Donadio, F., Frejaviile, J., Larnier, S., Vetault, S.: Human-robot collaboration to perform aircraft inspection in working environment. In: Proceedings of 5th International Conference on Machine Control and Guidance (MCG) (2016)
- [3] Miranda, J., Larnier, S., Herbulot, A., Devy, M.: Uav-based inspection of airplane exterior screws with computer vision. In: 14h International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications. (2019)
- [4] Tzitzilonis, V., Malandrakis, K., Zanotti Fragonara, L., Gonzalez Domingo, J.A., Avdelidis, N.P., Tsourdos, A., Forster, K.: Inspection of aircraft wing panels using unmanned aerial vehicles. *Sensors* **19**(8), 1824 (2019)

- [5] Almadhoun, R., Taha, T., Seneviratne, L., Dias, J., Cai, G.: A survey on inspecting structures using robotic systems. *International Journal of Advanced Robotic Systems* **13**(6), 1729881416663664 (2016)
- [6] Almadhoun, R., Taha, T., Dias, J., Seneviratne, L., Zweiri, Y.: Coverage path planning for complex structures inspection using unmanned aerial vehicle (uav). In: *International Conference on Intelligent Robotics and Applications*, pp. 243–266 (2019). Springer
- [7] Galceran, E., Carreras, M.: A survey on coverage path planning for robotics. *Robotics and Autonomous systems* **61**(12), 1258–1276 (2013)
- [8] Skorobogatov, G., Barrado, C., Salamí, E.: Multiple uav systems: A survey. *Unmanned Systems* **8**(02), 149–169 (2020)
- [9] Bugaj, M., Novák, A., Stelmach, A., Lusiak, T.: Unmanned aerial vehicles and their use for aircraft inspection. In: *2020 New Trends in Civil Aviation (NTCA)*, pp. 45–50 (2020). IEEE
- [10] Sun, Y., Zhang, L., Ma, O.: Robotics-assisted 3d scanning of aircraft. In: *AIAA AVIATION 2020 FORUM*, p. 3224 (2020)
- [11] Tao, F., Zhang, H., Liu, A., Nee, A.Y.: Digital twin in industry: State-of-the-art. *IEEE Transactions on Industrial Informatics* **15**(4), 2405–2415 (2018)
- [12] Browne, C.B., Powley, E., Whitehouse, D., Lucas, S.M., Cowling, P.I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., Colton, S.: A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games* **4**(1), 1–43 (2012)
- [13] Stützle, T., Hoos, H.H.: Improving the ant system: A detailed report on the max–min ant system. *FG Intellektik, FB Informatik, TU Darmstadt, Germany, Tech. Rep. AIDA–96–12* (1996)
- [14] Fendt, M.: Airbus launches advanced indoor inspection drone to reduce aircraft inspection times and enhance report quality. *Commercial Aircraft* **10** (2018)
- [15] Jing, W., Polden, J., Lin, W., Shimada, K.: Sampling-based view planning for 3d visual coverage task with unmanned aerial vehicle. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1808–1815 (2016). IEEE
- [16] Bircher, A., Alexis, K., Burri, M., Oettershagen, P., Omari, S., Mantel, T., Siegwart, R.: Structural inspection path planning via iterative viewpoint

- resampling with application to aerial robotics. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 6423–6430 (2015). IEEE
- [17] Jing, W., Deng, D., Xiao, Z., Liu, Y., Shimada, K.: Coverage path planning using path primitive sampling and primitive coverage graph for visual inspection. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1472–1479 (2019). IEEE
 - [18] Phung, M.D., Quach, C.H., Dinh, T.H., Ha, Q.: Enhanced discrete particle swarm optimization path planning for uav vision-based surface inspection. *Automation in Construction* **81**, 25–33 (2017)
 - [19] Silberberg, P., Leishman, R.C.: Aircraft inspection by multirotor uav using coverage path planning. In: 2021 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 575–581 (2021). IEEE
 - [20] Cabreira, T.M., Brisolara, L.B., Ferreira Jr, P.R.: Survey on coverage path planning with unmanned aerial vehicles. *Drones* **3**(1), 4 (2019)
 - [21] Kyaw, P.T., Paing, A., Thu, T.T., Mohan, R.E., Le, A.V., Veerajagadheswar, P.: Coverage path planning for decomposition reconfigurable grid-maps using deep reinforcement learning based travelling salesman problem. *IEEE Access* **8**, 225945–225956 (2020)
 - [22] Norman, A., Schönberg, A., Gorlach, I., Schmitt, R.: Cooperation of industrial robots with indoor-gps. In: Proceedings of the International Conference on Competitive Manufacturing, pp. 215–224 (2010)
 - [23] Wang, Z., Mastrogiacomio, L., Franceschini, F., Maropoulos, P.: Experimental comparison of dynamic tracking performance of igps and laser tracker. *The international journal of advanced manufacturing technology* **56**(1-4), 205–213 (2011)
 - [24] Koubâa, A., *et al.*: Robot Operating System (ROS). vol. 1. Springer, ??? (2017)
 - [25] Rusu, R.B., Cousins, S.: 3d is here: Point cloud library (pcl). In: 2011 IEEE International Conference on Robotics and Automation, pp. 1–4 (2011). IEEE
 - [26] Meyer, J., Sendobry, A., Kohlbrecher, S., Klingauf, U., von Stryk, O.: Comprehensive simulation of quadrotor uavs using ros and gazebo. In: 3rd Int. Conf. on Simulation, Modeling and Programming for Autonomous Robots (SIMPAN), p. (2012)