

Coverage Path Planning for Structural Inspections with UAV

Clément Laguerre¹, Hae-In Lee¹[0000-0001-9831-0127], Namhoon Cho¹[0000-0001-9999-9501], and Antonios Tsourdos¹[0000-0002-3966-7633]

Cranfield University, MK43 0AL, United Kingdom
{Clement.Laguerre.972, haein.lee, n.cho,
antonios.tsourdos}@cranfield.ac.uk

Abstract. This study presents a comprehensive framework for complete inspection of known structures using an Unmanned Aerial Vehicle subject to position uncertainty. The proposed strategy is to take a two-stage approach consisting of viewpoint planning followed by routing. The proposed method enhances an existing Viewpoints Generation algorithm, ensuring 100% coverage of the structure while significantly reducing both computing time and the number of viewpoints by up to 99% and 28%, respectively. The path planning problem is formulated to optimise either energy consumption or mission duration. Various Travelling Salesman Problem (TSP) solvers are compared, with the Lin-Kernighan Heuristic providing near-optimal results in a short time frame when applied to a full-scale wind turbine model. Validation is conducted in the Gazebo simulation environment, offering realistic conditions, while PX4 Software-In-The-Loop (SITL) and ROS2 ensure seamless deployment on real hardware. The code is available in the Zenodo archive [11].

Keywords: Coverage Path Planning · Unmanned Aerial Vehicle (UAV)
· Travelling Salesman Problem (TSP)

1 Introduction

Operation and Maintenance (O&M) expenses constitute a substantial portion of the total life cycle cost of most structures. For offshore wind turbines in particular, O&M costs can account for as much as 30% of the overall lifetime expenditure [22]. Primarily due to downtime, this leads to reduced energy output and increased operational expenses [14]. Unmanned Aerial Vehicles (UAVs) are a promising solution for structural inspection, offering rapid and safe access to areas that are difficult to reach. UAVs are anticipated to play a pivotal role in future multi-robot systems designed for the maintenance of various structures [22].

Ensuring the complete coverage of a structure using UAVs presents challenges such as position uncertainty and stability issues. Additionally, visual inspections require specific conditions to be met for successful execution. For instance, a fundamental aspect of visual inspection is the Ground Sampling Distance (GSD) which represents the number of pixels by millimetres in an image. It is used as part of the regulation to assess the validity of crack detection surveys [19].

To address the challenges in inspection flight planning, most existing methods divide the task into two parts [2, 5, 8, 9, 18, 19]: Viewpoints Generation (VG) and Path Generation (PG). VG methods can be classified into four categories: space sampling, vertex sampling patch sampling and surface sampling [6, 16].

Space sampling methods [2, 8] discretise the space surrounding the target structure, initialise viewpoints, evaluate their visibility, and iteratively refine them using various strategies. A key advantage lies in the use of a simplified model of the structure. However, this simplification often leads to increased "self-occlusions and partial coverage" [16], which limits their practical applicability.

In contrast, **vertex sampling** methods [4, 7, 15] operate directly on the object mesh, offering improved geometric fidelity [6]. Although effective for complex geometries, they tend to underperform on large flat surfaces with sparse vertices. Additionally, they are computationally expensive and sensitive to hyperparameter tuning [16].

Patch sampling approaches group mesh faces based on geometric features [16]. Mosbach et al. used B-splines, requiring expert-driven parameterisation [13], while Asa et al. applied k-means++ clustering to segment the mesh [3]. Although promising, real-world UAV tests exposed limitations due to pose uncertainty, later mitigated in [19]. Still, these methods are often hindered by implementation complexity and high computational cost [16].

Finally, **surface sampling** combines space sampling with mesh-based candidate generation [16]. Viewpoints are evaluated for visibility and filtered to maximise coverage, often outperforming other state-of-the-art methods. However, its primary focus lies in manufacturing contexts, and it lacks validation through real-world field tests [16].

Table 1. Comparison of VG sampling methods

Method	Advantages	Disadvantages
Space Sampling	Simplified structure model	Self-occlusions; partial coverage
Vertex Sampling	Considers mesh geometry	Poor on flat areas; sensitive to mesh resolution
Patch Sampling	Strong simulation and real-world performance	High computational cost; complex implementation
Surface Sampling	High coverage performance	Not validated in real-world tests

The purpose of PG is to provide the order of waypoints the UAV should follow to inspect the structure completely while minimising a chosen performance index. The task is usually abstracted as to solve a Travelling Salesman Problem (TSP) in the field of UAVs path planning [10]. It is well suited to the field of Coverage Path Planning (CPP) since its aim is to find the shortest route to visit all points in a graph exactly once and return to the starting point. All papers reviewed used a form of the TSP or the Vehicle Routing Problem.

To minimise energy consumption for PG, either physics-based or data-driven models are necessary to estimate the energy consumed over a given path. Physics-based models in [17, 20, 21] use motors, quad-rotors and battery dynamics to obtain a function relating control inputs and energy consumption. The problem is then treated as an optimal control problem. However, this requires devising the control system of the UAVs. It also demands data that necessitate wind tunnel tests. Another approach is to construct a data-driven model as in [1], where empirical data is obtained through real-world flight tests. This method allows obtaining equations relating different flight parameters with energy consumption.

As previously discussed, the method proposed by Wu et al. [19] demonstrates strong potential for VG, albeit with significant computational demands. The primary contribution of this paper is therefore a modified VG algorithm, inspired by Wu et al. [19], that significantly reduces both computation time and the number of viewpoints. In alignment with the requirements defined by Marshall Futureworx, this work also focuses on delivering a practical and deployable solution for autonomous structural inspection. To this end, the proposed open-source framework includes a computationally efficient VG algorithm, two time-efficient PG optimisation scenarios, and a complete pipeline suitable for both simulation and real-world deployment on UAV hardware.

The rest of the paper is organised as follows: Section 2 gives an overview of the proposed inspection planning framework, consisting of VG and PG modules. Each of the modules is detailed in Section 2.1 and 2.2. Section 3 presents the performance of the proposed approach on benchmark structures and a wind turbine. Conclusions and future research directions are proposed in Section 4.

2 Inspection Planning Framework

The aim of visual inspection is to obtain a set of images of the entire target structure without omission. To achieve this, an array of viewpoints is computed based on a 3D model of the target structure. Described as a 1×6 vector, each viewpoint represents the position and orientation of a camera mounted on a gimbal. Knowing the position of the camera relative to the center of the UAV, a list of waypoint coordinates can be derived. These waypoints are then used as input for the PG algorithm, which determines the sequence in which the UAV should visit each waypoint to either ensure the shortest mission duration (Scenario 1) or the lowest energy consumption (Scenario 2). These steps and the data flow are summarised in Figure 1.

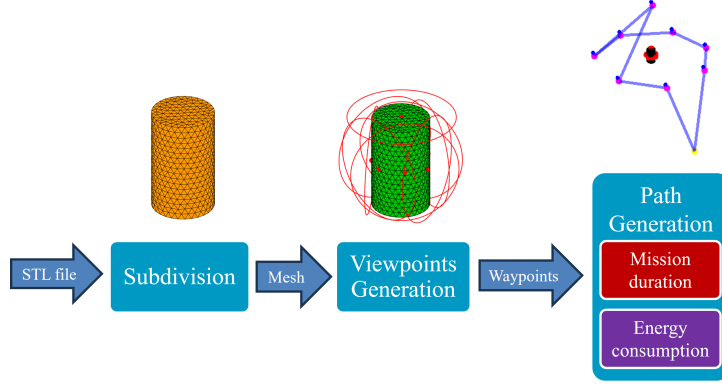


Fig. 1. Inspection Path Planning Framework

2.1 Viewpoints Generation

The patch sampling method described in [19] has been modified to reduce the number of viewpoints and computational time. Initially, the target structure is divided into a set of triangular polygons M_{sub} using the gmsh software, with a maximum edge length of 200 mm. The algorithm then clusters M_{sub} into k clusters, $C = \{c_1, c_2, \dots, c_k\}$ and, knowing the inspection distance d_{insp} , computes the inspection position p . A polygon is considered inspected if it falls within the inspection range of at least one viewpoint and if its view angle θ is below a user-defined limit θ_{lim} . The view angle is defined as the angle between the normal to the surface at the points c_i and the normal to the surface at the centroid of the polygon. In [19], the clustering algorithm is applied to M_{sub} , starting with $k = 1$. If a polygon is not inspected, k is incrementally increased by one until M_{sub} is fully inspected. This process is computationally intensive and does not minimise the number of viewpoints, as the clustering algorithm tends to distribute the viewpoints unevenly. The proposed method aims to address these two issues.

Firstly, instead of applying k-means++ to the entire structure, M_{sub} is segmented into N_s sections along the vertical axis, as defined by

$$N_s = \frac{h_s}{2 \cdot r_{\text{maj}} \cdot r_s} \quad (1)$$

where h_s is the height of the structure (m), r_{maj} is the radius of the inspection area (m), and r_s is an adjustable parameter, empirically set to 0.7. This segmentation ensures that the height of each section is slightly smaller than the diameter of the inspection area, facilitating the even distribution of medoids along a central line within each section. Subsequently, the number of clusters, k , is set to one, with inspected polygons I_p being sequentially removed from M_{sub} after each iteration until all polygons are inspected. The integration of these enhancements results in reduced computing time, minimised overlap, and fewer required viewpoints (see Algorithm 1).

Algorithm 1 Inspection Position Generation

```

1: Input:  $M_{\text{sub}}, d_{\text{insp}}, \theta_{\text{lim}}$ 
2: Output:  $P_{\text{insp}} = \{p_1, p_2, \dots, p_N\}$ 
3:  $N_s \leftarrow \left\lfloor \frac{h_s}{2 \cdot r_{\text{maj}} \cdot r_s} \right\rfloor$ 
4: Segment  $M_{\text{sub}}$  into  $N_s$  sections along the vertical axis
5:  $P_{\text{insp}} \leftarrow \emptyset$ 
6: for each section  $M_{\text{sub}}^i \subset M_{\text{sub}}$  do
7:    $M_{\text{remain}}^i \leftarrow M_{\text{sub}}^i$ 
8:   while  $M_{\text{remain}}^i \neq \emptyset$  do
9:      $c \leftarrow \text{Clustering}(M_{\text{remain}}^i, k = 1)$ 
10:     $p \leftarrow \text{Determine}(c, d_{\text{insp}})$ 
11:     $P_{\text{insp}} \leftarrow P_{\text{insp}} \cup \{p\}$ 
12:     $I_p \leftarrow \text{Determine}(c, r_{\text{maj}}, \theta_{\text{lim}})$ 
13:     $M_{\text{remain}}^i \leftarrow M_{\text{remain}}^i \setminus I_p$ 
14:   end while
15: end for

```

2.2 Path Generation and Optimisation

Scenario 1: Mission Duration To minimise the mission duration, the path length is minimised using the classical TSP formulation. The objective function is defined as:

$$\min \sum_{i=1}^n \sum_{j=1}^n d_{ij} \cdot x_{ij} \quad (2)$$

where n is the number of viewpoints, d_{ij} is the Euclidean distance between the waypoints i and j , and x_{ij} is a binary decision variable which is 1 if it is chosen to be connected, and 0 otherwise. To minimise the path length, various TSP solvers are considered: intlinprog [12], Lin-Kernighan Heuristic (LKH), Ant Colony Optimisation (ACO), Hybrid Particle Swarm Optimisation (HPSO), Genetic Algorithm (GA) and Simulated Annealing (SA).

Scenario 2: Energy Consumption Given the need for precise positioning and stabilisation, the UAV must effectively halt at each viewpoint. Considering that the number of viewpoints extends into the thousands for a real-size structure, and the relatively short distances between them, the speed between viewpoints is expected to be minimal to avoid unnecessary long stabilisation period. Consequently, trajectory optimisation does not appear to be necessary, and factors such as jerk are not considered. Hence, the primary contributors to energy consumption are altitude changes and path length, necessitating a multi-objective optimisation approach. Therefore, the objective function for Scenario 2 is formulated as the weighted sum of both distances between each waypoint (3).

$$\min \sum_{i=1}^n \sum_{j=1}^n [((1-w) \cdot a_{ij} + w \cdot d_{ij}) \cdot x_{ij}] \quad (3)$$

where a_{ij} is the altitude difference between the waypoints i and j , and w is the weight between the altitude change and path length minimisation. The value of w is changed from 0 to 1 with a step of 0.025 to obtain the Pareto optimal front. The weight that ensure the lowest energy consumption can be obtained based on line search over the Pareto optimal front.

To estimate the energy consumption, a data-driven model from [1] is used. First, the algorithm identifies the flight manoeuvres between each waypoint as flying vertically upwards or flying vertically downwards. Then the energy consumed over the whole path, E , can be approximated as:

$$E = \begin{cases} 315D - 211.261, & (E_v) \text{ if flying upwards} \\ 68.956D - 65.183, & (E_v) \text{ if flying downwards} \\ 308.709t - 0.852, & (E_h) \text{ if flying horizontally} \end{cases} \quad (4)$$

where $0 < D \leq 7.5$ m is the distance of the vertical flight and $t < 900$ s is the duration of the horizontal flight. This equation was obtained by recording the energy consumed on a real UAV performing different flights multiple times. The results show an error of 4.3% between the estimate and the real consumption. Using this method, it is possible to estimate the energy consumption without detailed knowledge of the platform. Since flying vertically downwards and upwards with an angle was not included in the test flights performed, the energy consumed over a diagonal vertical path is estimated as the sum of the energy consumption of the vertical (D_v) and horizontal distances (D_h) as in:

$$E = \begin{cases} E_v * r + E_h, & \text{if } D_h > D_v \\ E_v + E_h * r, & \text{if } D_v > D_h \end{cases} \quad (5)$$

The smallest of the distances is multiplied by a ratio r based on the hypotenuse of the triangle formed by the sequential waypoints:

$$r = \frac{\sqrt{D_v^2 + D_h^2}}{D_v + D_h}. \quad (6)$$

To get an accurate energy consumption model on a given UAV platform, the methodology explained in [1] should be followed. Then, as (6) is an estimate, performing additional test flight flying upwards and downwards at different angle would enhance the accuracy of the results. Moreover, as stated in [1], performing the different manoeuvres at different speeds also enables finding the one that results in the lowest energy consumption for each manoeuvre. On another note, if the sufficient test flights are performed, using the energy consumption function directly as the objective function in the TSP solver would result in the most accurate solution.

3 Results

3.1 Simulation Setup

The performance of the proposed approach was validated using benchmark structures and a full-scale model of a wind turbine. The benchmark structures included a flat wall of 73 m², as referenced in [19], to compare their VG performance, and a cylinder with a view angle limit of 10° to assess performance on a curved structure. Additionally, the algorithm's ability to handle increasing positional uncertainty was evaluated, as it is a critical factor contributing to coverage loss. To mitigate this effect, the inspection distance and range were reduced to ensure 100% coverage of the structure.

Multiple tests were performed using PX4 SITL, Gazebo, and ROS2 to validate these results. The X500 model from PX4 has been used without gimbal. Each time the UAV reaches a waypoint, its ground truth position is stored in a .csv file, which is then processed to issue the resulting coverage, overlap and position error. The resulting path, waypoints and medoids can be overlaid on top of the computed ones as shown in Figure 2. The position error is achieved by setting the error allowance δ_{position} for reaching a waypoint to the desired value. Additionally, the default noise of the simulated barometer, GNSS and IMU in Gazebo contributed to modelling the position error and provided a more realistic environment. These measurements were fused by an Extended Kalman Filter (EKF) within the PX4 Autopilot. The inspection position error and medoids' position error were evaluated by averaging the results of 5 different flight simulations.

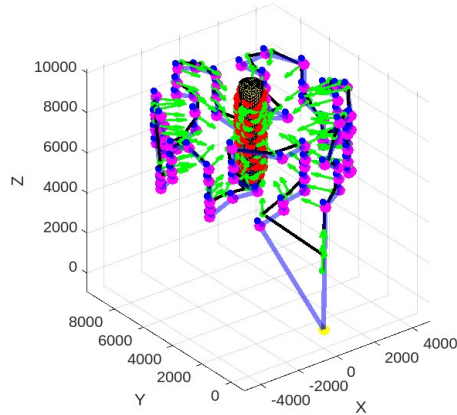


Fig. 2. Post-processed visualisation of a cylindrical structure. Algorithm outputs include: medoids (red), waypoints (magenta), and viewpoints (blue). Simulation outputs include: viewpoints around the structure (green), medoids on the surface from ray tracing (green), and the UAV path (black). The yellow dot marks the UAV take-off and landing location.

3.2 Benchmark Structures

Wall The VG results for a wall structure are shown in Table 2. The method from [19] has been implemented and is referred to as “ [19] reproduced”. It issued 272 viewpoints (Figure 3.a) while the results from the paper reported 285 viewpoints. The difference can be explained by different reasons such as the triangular polygons size in the mesh. Moreover, clustering may not provide the exact same results each time the algorithm is run. Nevertheless, the implemented method from [19] is considered similar enough for comparison. As seen in Table 2, the proposed method performs faster with a smaller overlap and a smaller number of viewpoints (Figure 3.b). The proposed algorithm issued 196 waypoints in 12 seconds compared with 272 waypoints in 1 hour and 13 minutes. This resulted in a 99.7% reduction in computing time and a 28% reduction in the number of waypoints.

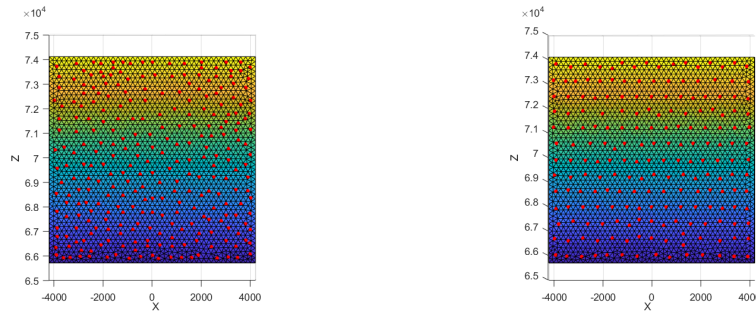


Fig. 3. VG results on a 73 m² flat wall: left (a) — reproduced method from [19]; right (b) — proposed method. Red dots represent the medoids. Inspection poses are computed from medoid positions, surface normals, and inspection distance. The proposed method produces a sparser and more uniform medoid distribution. The colors simply refer to the wall height

Table 2. VG Results for a Wall

Criteria	[19]	[19] reproduced	Proposed
Overlap	N/A	89%	71%
Computing Time	N/A	1h 13min	12s
Number of Viewpoints	285	272	196
Coverage	100%	100%	100%

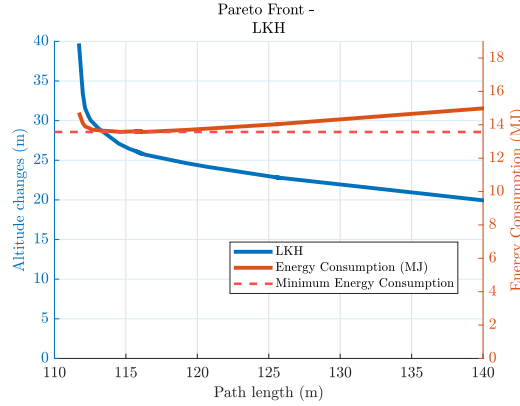
The PG results for inspecting a flat wall are summarised in Table 3 for both optimisation scenarios. In [19], the path that ensured the highest coverage was 153.4 metres long while ACO and LKH issued paths of 128.8 metres and 111.7 metres long respectively.

Table 3. PG Results for a Wall

Metric	Scenario 1			Scenario 2
	ACO	LKH	[19]	LKH
Path Length (m)	128.8	111.7	153.4	114.6
Altitude Change (m)	48.0	39.7	Not provided	27.1
Energy Consumption (MJ)	17.3	14.7	Not provided	13.6
Computational Time	82.2 s	8.47 s	Not provided	12 min (all weights)

Regarding Scenario 2, the Pareto optimal front is obtained as Figure 4, showing a strong correlation between an initial reduction of the altitude changes and a reduction in the energy consumption. The minimum energy consumption is obtained for weights of 0.475 and 0.500, inducing an altitude change of 27.1 m and a path length of 114.6 m. Then, as the altitude change is minimised, the path length increases greatly, causing an increase in energy consumption.

The last column on the right in Table 3 shows the results for the lowest energy consumption. The algorithm shows a decrease in energy consumption of 7.87% while the path length increased by 2.60%. Only the LKH solver was run as it provided the best results.

**Fig. 4.** Pareto front illustrating the trade-off between altitude changes and path length, considering energy consumption

Cylinder Table 4 compares the VG results from [19] and the proposed method on a cylinder (height = 5 m, diameter = 1 m). It is shown that the computing time drastically increases on large or non-flat structures using the existing method [19]. Most importantly, the number of viewpoints increases when handling structures with multiple faces.

Table 4. VG Results for a Cylinder

Criteria	[19] reproduced	Proposed
Overlap	17.8%	9.45%
Computing Time	232s	10s
Viewpoint Number	107	80
Coverage	100%	100%

The VG results under the existence of position uncertainty are shown in Table 5. The algorithm achieved 100% coverage for up to 0.6 m in position error. Above this, for the camera and GSD specification, the inspection range was too small to be considered usable as it was smaller than the size the triangular polygons of the mesh. Such results are attained by reducing the inspection distance and inspection area as described in [19].

Table 5. Evaluation of the coverage under position uncertainty

Position uncertainty (m)	Inspection position error (m)	Medoids position error (m)	Coverage (%)	Inspection distance (m)	Inspection radius (m)
0.2	0.186	0.198	100	3.79	0.56
0.3	0.235	0.277	100	3.69	0.44
0.4	0.314	0.348	100	3.59	0.32
0.5	0.458	0.483	100	3.49	0.19
0.6	0.518	0.547	100	3.39	0.072

The mission duration minimisation (Scenario 1) results of the generated viewpoints around a cylinder with 118 waypoints are shown in Table 6. For Scenario 2, the energy consumed decreased by 5.1% (LKH results). Increasing the waypoints number to 269 to assess the scalability of the TSP solvers, Mixed-Integer Linear Programming (MILP) provides an optimal path in 48 minutes while LKH does the same in under 6 seconds. ACO also provides acceptable results while SA, GA and HPSO perform worse.

Table 6. PG results showing path length (m) and computing time (s) for a cylinder (118 waypoints)

	MILP	SA	GA	HPSO	ACO	LKH
Path length (m)	196	267	348	383	221	196
Computing time (s)	7.1	6	17	8.2	7.5	5

3.3 Wind Turbine

The VG results of a real-size wind turbine (height = 75 m) are shown in Table 7. The viewpoints for the tower and nacelle were obtained in 43 minutes while the ones for the blades were obtained in 34 minutes. The results are visualised in Figure 5. The computational time is considered acceptable in practice, as the viewpoints can be computed ahead of the inspection.

Table 7. VG Results for a Wind Turbine

Criteria	Tower and nacelle	Blades
Overlap	64%	79.1%
Computing time	43 min	34 min
Viewpoint Number	3972	3436
Coverage	100%	100%

The PG results are summarised in Table 8. ACO ran for 1 hour 22 minutes, performing 100 iterations, and generated a 3319-meter long path. In contrast, LKH with default parameters generated a 2171-meter long path in 35 minutes. The project’s aim was to achieve acceptable computing time for prompt inspection after the wind turbine stops. An arbitrary 15-minute limit was set for the LKH solver to compare its quality with default parameters. The results were identical, showing an optimal path in a feasible time frame. Even with a 1-minute limit, the path length only increased by a few meters in the worst case.

Table 8. Evaluation of the PG algorithm optimised for mission duration on a wind turbine

Metric	ACO	LKH	LKH	LKH
Tower and Nacelle				
Path Length (m)	3319	2171	2171	2171
Altitude Change (m)	1932	1104	1104	1094
Energy Consumption (MJ)	526.8	329.4	329.4	328.1
Computing Time	1h 22m	35 min (free)	15 min (fixed)	1 min (fixed)
Blades				
Path Length (m)	2475	1568	1568	1568
Altitude Change (m)	1370	887	887	893
Energy Consumption (MJ)	385.9	245.3	245.3	246.3
Computing Time	26 min	28 min (free)	15 min (fixed)	1 min (fixed)

The optimisation for lowest energy consumption was run with a time limit of 15 seconds per weight, for a total target computing time of approximately 15 minutes. The best results were obtained for a weight of 4.750 which shows a decrease of 7.63% and 8.02% of energy consumption for the tower and nacelle and the blades, respectively (Table 9).

On the full wind turbine, this represents a 7.76% decrease, increasing the path length by 6.98% and reducing the altitude changes by 30.25%.

Table 9. Evaluation of the PG algorithm optimised for energy consumption on a wind turbine

Metric	Tower and Nacelle	Blades
Path Length (m)	2313	1687
Altitude Change (m)	771	615
Energy Consumption (MJ)	304.3	225.7
Computing Time	15 min	13 min



Fig. 5. VG results for the wind turbine. The magenta dots are the generated view-points.

4 Conclusion

This paper presents an algorithm capable of planning an inspection path for the complete coverage of known structures using a UAV. The algorithm improves upon an existing VG algorithm by reducing computing time and the number of viewpoints. Several TSP solvers are compared, with the option of optimising either mission duration or energy consumption. Tests on a real-size wind turbine demonstrated the practicality of the proposed method, achieving short computing times in both scenarios. The simulations proved the algorithm to be robust to position uncertainties. Validation carried out in ROS2 and PX4 SITL allows for streamlined tests and easy deployment on a real platform. Future work may include simulation testing of energy consumption gains, collision-free path planning and test flights on a real platform to assess the algorithm's coverage capabilities.

Acknowledgments. This work was supported by Marshall Futureworx.

References

1. Abeywickrama, H.V., Jayawickrama, B.A., He, Y., Dutkiewicz, E.: Comprehensive energy consumption model for unmanned aerial vehicles, based on empirical studies of battery performance. *IEEE Access* **6**, 58383–58394 (2018). <https://doi.org/10.1109/ACCESS.2018.2875040>
2. Almadhoun, R., Taha, T., Gan, D., Dias, J., Zweiri, Y., Seneviratne, L.: Coverage path planning with adaptive viewpoint sampling to construct 3D models of complex structures for the purpose of inspection. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 7047–7054. Madrid, Spain (October 2018). <https://doi.org/10.1109/IRoS.2018.8593719>
3. Asa, K., Funabara, Y., Doki, S., Doki, K.: Evaluation in real world of the measuring position determination for visual inspection using UAV. In: *44th Annual Conference of the IEEE Industrial Electronics Society*. pp. 2711–2716. Washington, DC, USA (October 2018). <https://doi.org/10.1109/IECON.2018.8591270>
4. Englot, B.J.: Sampling-Based Coverage Path Planning for Complex 3D Structures. Ph.D. thesis, Massachusetts Institute of Technology (2012)
5. Garlapati, V., Dasgupta, P.: A complete coverage algorithm for 3D structural inspection using an autonomous unmanned aerial vehicle. In: *Thirty-First International Florida Artificial Intelligence Research Society Conference*. pp. 342–347. Melbourne, FL, USA (May 2018)
6. Gospodnetić, P., Mosbach, D., Rauhut, M., Hagen, H.: Viewpoint placement for inspection planning. *Machine Vision and Applications* **33** (2022). <https://doi.org/10.1007/s00138-021-01252-z>
7. Gronle, M., Osten, W.: View and sensor planning for multi-sensor surface inspection. *Surface Topography: Metrology and Properties* **4**(2) (2016). <https://doi.org/10.1088/2051-672X/4/2/024009>
8. Jing, W., Deng, D., Wu, Y., Shimada, K.: Multi-UAV coverage path planning for the inspection of large and complex structures. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 1480–1486. Las Vegas, NV, USA (October 2020). <https://doi.org/10.1109/IRoS45743.2020.9341089>
9. Jung, S., Song, S., Youn, P., Myung, H.: Multi-layer coverage path planner for autonomous structural inspection of high-rise structures. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 1–9. Madrid, Spain (October 2018). <https://doi.org/10.1109/IRoS.2018.8593537>
10. Khoufi, I., Laouiti, A., Adjih, C.: A survey of recent extended variants of the traveling salesman and vehicle routing problems for unmanned aerial vehicles. *Drones* **3**(3) (2019). <https://doi.org/10.3390/drones3030066>
11. Laguerre, C.: Uav_structural_inspection (2025). <https://doi.org/10.5281/zenodo.15556065>, <https://doi.org/10.5281/zenodo.15556065>, accessed: 2025-05-30
12. MATHWORKS: intlinprog. <https://fr.mathworks.com/help/optim/ug/intlinprog.html> (2025), accessed: 2025-05-27
13. Mosbach, D., Gospodnetić, P., Rauhut, M., Hamann, B., Hagen, H.: Feature-driven viewpoint placement for model-based surface inspection. *Machine Vision and Applications* **32** (2021). <https://doi.org/10.1007/s00138-020-01116-y>
14. Rinaldi, G., Thies, P.R., Johanning, L.: Current status and future trends in the operation and maintenance of offshore wind turbines: A review. *Energies* **14**(9) (2021). <https://doi.org/10.3390/en14092484>

15. Scott, W.R.: Viewpoint placement for inspection planning. *Machine Vision and Applications* **20**, 47–69 (2009). <https://doi.org/10.1007/s00138-007-0110-2>
16. Staderini, V., Glück, T., Schneider, P., Mecca, R., Kugi, A.: Surface sampling for optimal viewpoint generation. In: 13th IEEE International Conference on Pattern Recognition Systems. pp. 1–7. Guayaquil, Ecuador (July 2023). <https://doi.org/10.1109/ICPRS58416.2023.10179043>
17. Thibbotuwawa, A., Nielsen, P., Zbigniew, B., Bocewicz, G.: Energy consumption in unmanned aerial vehicles: A review of energy consumption models and their relation to the UAV routing. In: 39th International Conference on Information Systems Architecture and Technology. pp. 173–184. Nysa, Poland (September 2018)
18. Tong, H.W., Li, B., Huang, H., Wen, C.: UAV path planning for complete structural inspection using mixed viewpoint generation. In: 17th International Conference on Control, Automation, Robotics and Vision. pp. 727–732. Singapore (December 2022). <https://doi.org/10.1109/ICARCV57592.2022.10004359>
19. Wu, W., Funabara, Y., Doki, S., Doki, K., Yoshikawa, S., Mitsuda, T., Xiang, J.: Evaluation and enhancement of resolution-aware coverage path planning method for surface inspection using unmanned aerial vehicles. *IEEE Access* **12**, 16753–16766 (2024). <https://doi.org/10.1109/ACCESS.2024.3359056>
20. Yacef, F., Rizoug, N., Bouhali, O., Hamerlain, M.: Optimization of energy consumption for quadrotor UAV. In: 9th International Micro Air Vehicle Conference and Flight Competition 2017. pp. 215–222. Toulouse, France (September 2017)
21. Yacef, F., Rizoug, N., Degaa, L., Hamerlain, M.: Energy-efficiency path planning for quadrotor UAV under wind conditions. In: 7th International Conference on Control, Decision and Information Technologies. pp. 1133–1138. Prague, Czech Republic (June–July 2020). <https://doi.org/10.1109/CoDIT49905.2020.9263968>
22. Zhang, K., Pakrashi, V., Murphy, J., Hao, G.: Inspection of floating offshore wind turbines using multi-rotor unmanned aerial vehicles: Literature review and trends. *Sensors* **24**(3) (2024). <https://doi.org/10.3390/s24030911>



DSSPACE

<https://dspace.org/>

Coverage path planning for structural inspections with UAV

Laguerre, Clément; Lee, Hae-In; Cho, Namhoon; Tsourdos, Antonios

2025-08-13

Springer

https://doi.org/10.1007/978-3-032-01486-3_31

Laguerre C, Lee H-I, Cho N, Tsourdos A. (2026) Coverage path planning for structural inspections with UAV. In: Towards Autonomous Robotic Systems (TAROS 25), 26th Annual Conference, TAROS 2025, 20-22 August 2025, York, UK, Lecture Notes in Computer Science (LNAI) Volume 16045