ETH zürich

ETH Library

Structural Inspection Path Planning via Iterative Viewpoint Resampling with Application to Aerial Robotics

Conference Paper

Author(s):
Bircher, Andreas; Alexis, Kostas; Burri, Michael; Oettershagen, Philipp; Omari, Sammy; Mantel, Thomas A.

Publication date:
2015

Permanent link:
https://doi.org/10.3929/ethz-a-010466157

Rights / license:
In Copyright - Non-Commercial Use Permitted

Originally published in:
https://doi.org/10.1109/ICRA.2015.7140101

This page was generated automatically upon download from the ETH Zurich Research Collection.
For more information, please consult the Terms of use.

Structural Inspection Path Planning via Iterative Viewpoint Resampling with Application to Aerial Robotics

Andreas Bircher, Kostas Alexis, Michael Burri, Philipp Oettershagen,
Sammy Omari, Thomas Mantel and Roland Siegwart[1]

***Abstract*— Within this paper, a new fast algorithm that provides efficient solutions to the problem of inspection path planning for complex 3D structures is presented. The algorithm assumes a triangular mesh representation of the structure and employs an alternating two–step optimization paradigm to find good viewpoints that together provide full coverage and a connecting path that has low cost. In every iteration, the viewpoints are chosen such that the connection cost is reduced and, subsequently, the tour is optimized. Vehicle and sensor limitations are respected within both steps. Sample implementations are provided for rotorcraft and fixed–wing unmanned aerial systems. The resulting algorithm characteristics are evaluated using simulation studies as well as multiple real–world experimental test–cases with both vehicle types.**

I. Introduction

The ongoing boom in utilizing mobile robots for real–life applications sets new demands regarding their autonomy. In the particularly interesting field of inspection operations, aerial, maritime or ground robots are already utilized for critical tasks such as infrastructure surveillance, damage assessment or victim search. In such scenarios, the structure to be inspected may be given as a 3D model (typically a mesh from CAD software, Geographical Information System data or civil engineering instrumentation) and robots are employed either to derive an updated, possibly higher–fidelity model, or scan for risks and hazards (e.g. cracks).

To facilitate autonomous inspection planning capabilities, a robot must be equipped with algorithms that allow it to quickly compute efficient paths that result in full coverage of the structure to be inspected, while respecting any sensor limitations and motion constraints that may apply. Such a problem belongs to the general class of coverage planning and –despite the interest of the community– its inherent difficulties still limit the performance, efficiency and applicability of the proposed solutions. Furthermore, so far, only few works validated such algorithms in experimental studies.

Within this work, a novel fast iterative algorithm for structural inspection is proposed. The new algorithm employs an alternating two–step optimization paradigm to find good viewpoints that together provide full coverage and lead to a connecting path that is of low cost. In every iteration, each viewpoint is chosen such as to reduce the cost–to–travel between itself and its neighbours (*first step*) and subsequently the optimally connecting tour is recomputed (*second step*), while vehicle constraints and sensor limitations are respected in all phases. Extensive evaluation studies including 3D reconstruction experiments using a rotorcraft (hexarotor) Unmanned Aerial Vehicle (UAV), as well as a 5.6m wingspan glider fixed–wing UAV (both shown in Figure 1) reveal the high–performance properties of the algorithm in challenging scenarios and subject to real vehicle and sensor constraints. An open source implementation [17] as well as the point-clouds resulting from the experiments [2] are provided for further use and development by the comunity.

Fig. 1: Indicative inspection 3D reconstruction results using the proposed structural inspection planner and a rotorcraft as well as a fixed–wing UAV equipped with camera sensors.

A short overview of the related work and how our approach contributes further is presented in Section II, followed by the problem description in Section III. Afterwards, the proposed approach is presented in Section IV, while computational analysis takes place in Section V. Finally, evaluation test–cases in simulation and experiments are shown in Section VI, followed by conclusions in section VII.

II. Related Work

In the literature, many contributions have been made towards addressing the challenges of coverage planning. System and environment allowing, the space may be represented by a simplified discrete grid and paths can be computed using wavefront algorithms [23], spanning trees [7] or neural networks [16]. The work in [4] proposed a cellular decomposition of the planning space, covering each free cell with a sweeping pattern and an advanced algorithm following this

This work has received funding from the European Union's Horizon 2020 Research and Innovation Programme under the Grant Agreement No.644128, AEROWORKS as well as by the FP7-framework Grant No. 285417, ICARUS.

[1]All authors are with the Autonomous Systems Lab at ETH Zurich, Leonhardstrasse 21, 8092, Zurich, Switzerland. email: andreas.bircher@mavt.ethz.ch

concept was presented in [1]. Using a 2D planner, the authors in [11] approximate a 3D structure using multiple 2D layers.

Aiming towards real 3D structural inspection, advanced algorithms have recently been proposed. Within the most recent contributions, those that employ a two–step optimization scheme proved to be more versatile with respect to the inspection scenario. In a first step, such algorithms compute the minimal set of viewpoints that cover the whole structure which corresponds to solving an Art Gallery Problem (AGP). As a second step, the shortest connecting tour over all these viewpoints has to be computed, which is the Traveling Salesman Problem (TSP). Fast algorithms to approximately solve these two NP–hard, but well studied problems, are known, for example in [9, 18] for the AGP, and [5, 15] for the TSP. A recent application of these concepts, that allows some redundancy in the AGP such that it is able to improve the path in a post–processing step, was presented in [12]. This algorithm can deal with 3D scenarios and is demonstrated in experiments using underwater vehicles for ship hull inspection. Addressing the problem from a different perspective, the work in [21] concentrates on deriving close–to–optimal solutions at the inherently large cost of computational efficiency. A comprehensive survey of the existing coverage path planning methods may be found in [8].

The proposed fast inspection path planner retains a two–step optimization structure but contrary to trying to find a minimal set of guards in the AGP it rather tries to sample them such that the connecting path is short while ensuring full coverage. This is driven by the idea that with a continuously sensing sensor the number of viewpoints (and if this is minimal or not) is not necessarily important but mostly their configuration in space, which has to be such that short and full coverage paths are provided. As a result, this novel approach leads to full coverage paths that are of low cost and are computed quickly.

III. Problem Description

The problem of structural inspection path planning, as it is considered in this paper, consists of a 3D structure to be inspected, a system with its dynamics and constraints and an integrated sensor, the limitations of which have to be respected. The 3D structure to be inspected is represented by a triangular mesh, embedded in a bounded environment that may contain obstacle regions. The problem setup is to be such that for each triangle in the mesh, there exists an *admissible* viewpoint configuration – that is a viewpoint from which the triangle is visible for a specific sensor model. Then, for the given environment and with respect to the operational constraints, a path for the system has to be found that guarantees complete inspection of the 3D structure. Quality measures for paths are situation specific, depending on the system and mission objectives, e.g. time or distance.

As sample systems we consider a rotorcraft and a fixed–wing UAV, both equipped with a visual camera with a fixed orientation relative to the platform. Minor adaptations to the proposed path–planner enable its use for other common robot configurations such as underwater ROVs or wheeled robots.

IV. Proposed Approach

As the algorithm does not focus on minimizing the number of viewpoints, the proposed approach selects one (*admissible*) viewpoint for every triangle in the mesh of the structure to be inspected. In order to compute viewpoints that allow low–cost connections, an iterative resampling scheme is employed. Between each resampling, the best path for the current viewpoints is computed. The cost to connect to the current neighbours on the tour provides a metric for the quality of the viewpoint in the subsequent resampling. The initial selection of viewpoints for the first iteration is arbitrarily done such that full coverage is provided with non–optimized viewpoints. A fast implementation of the Lin-Kernighan-Helsgaun Heuristic (LKH) TSP solver [10] is employed to compute the best tour, while the cost of the interconnecting pieces of path is calculated by means of a boundary value solver (BVS). Algorithm 1 presents an overview of the proposed inspection planning procedure.

Algorithm 1 Inspection path planner

```
k ← 0
Sample initial viewpoint configurations
Compute cost matrix for the TSP solver (Section IV-A)
Solve the TSP problem to obtain initial tour
while running
    Resample viewpoint configurations (Section IV-B)
    Recompute the cost matrix (Section IV-A)
    Recompute best tour T_best using the LKH and update
        best tour cost c_best if applicable
    k ← k + 1
end while
return T_best, c_best
```

In the following, the formulations of the path computation and the viewpoint sampling for a rotorcraft UAV are given. Subsequently in Section IV-C, extensions and adaptations to enable planning for a fixed–wing UAV are discussed.

A. Path Computation and Cost Estimation

To find the best tour among the viewpoints, the TSP solver requires a cost matrix containing the connection cost of all pairs of viewpoints. The path generation and its cost estimation relie on a two state BVS. The BVS is either employed directly to connect the two viewpoints or as a component in a local planner, in case the direct connection is not feasible due to obstacles. In that case, our implementation makes use of the RRT*-planner [14] to find a collision–free connection. The proposed model for a rotorcraft UAV consists of position as well as yaw, $\xi = \{x, y, z, \psi\}$. Roll and pitch angles are considered to be near zero as slow maneuvering is desired to achieve increased accuracy. The path from configuration ξ_0 to ξ_1 is given by $\xi(s) = s\xi_1 + (1 - s)\xi_0$, where $s \in [0, 1]$. The single limitation considered is the speed limit. The translational limit is denoted by v_{max} while the rotational speed is limited by $\dot{\psi}_{max}$. Both values are small, such that the tracking of

paths with corners is sufficiently accurate. The resulting execution time is $t_{ex} = \max(d/v_{max}, \|\psi_1 - \psi_0\|/\dot{\psi}_{max})$, with d the Euclidean distance. The cost of a path segment corresponds to the execution time t_{ex}.

B. Viewpoint Sampling

For every triangle in the mesh, one viewpoint has to be sampled, the position and heading of which is determined sequentially in the proposed procedure while retaining visibility of the corresponding triangle. First, the position is optimized for distance to the neighbouring viewpoints using a convex problem formulation and only then, the heading is optimized. To guarantee a good result of this multistep optimization process, the position solution must be constrained such as to allow finding an orientation for which the triangle is visible.

Specifically, the constraints on the position $g = [x, y, z]$ consist of the inspection sensor limitations of minimum incidence angle, minimum and maximum range (d_{min}, d_{max}) constraints (depicted in Figure 2a). They are formulated as a set of planar constraints:

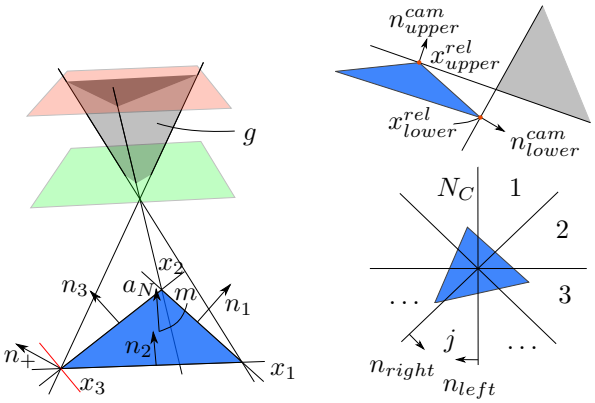
$$\begin{bmatrix} (g - x_i)^T n_i \\ (g - x_1)^T a_N \\ -(g - x_1)^T a_N \end{bmatrix} \succeq \begin{bmatrix} 0 \\ d_{min} \\ -d_{max} \end{bmatrix}, i = \{1, 2, 3\} \quad (1)$$

where x_i are the corners of the mesh triangle, a_N is the normalized triangle normal and n_i are the normals of the separating hyperplanes for the incidence angle constraints as shown in Figure 2a.

Further, the camera has a limited field of view (FoV) with a certain horizontal and vertical opening and is mounted to the system with a fixed pitch angle. The imposed constraint on the sampling space resulting from the vertical camera opening is not convex (a revoluted 2D-cone, the height of which is depending on the relevant corners of the triangle over the revolution). To approximate and convexify the problem, the space is divided in N_C equal convex pieces according to Figure 2b. The optimum is computed for every slice in order to find the globally best solution. The constraints for piece j are derived as follows: Left and right boundaries of the sampling space are the borders of the revolution segment and the cone top and bottom are represented by a single plane tangential to the centre of the slice. Angular camera constraints in horizontal direction are not encoded and instead d_{min} is chosen high enough to allow full visibility of the triangle. This leaves some space for variation in the sampling of the heading, where the horizontal constraints are enforced. Specifically, these constraints are:

$$\begin{bmatrix} (g - x^{rel}_{lower})^T n^{cam}_{lower} \\ (g - x^{rel}_{upper})^T n^{cam}_{upper} \\ (g - m)^T n_{right} \\ (g - m)^T n_{left} \end{bmatrix} \succeq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (2)$$

where $x^{rel}_{lower}, x^{rel}_{upper}$ are the respective relevant corners of the mesh triangle, m the middle of the triangle and n^{cam}_{lower}, n^{cam}_{upper}, n_{right} and n_{left} denote the normal of the respective separating hyperplanes.



(a) Incidence angle constraints on a triangular facet

(b) Camera constraints and convexification

Fig. 2: a) The figure depicts the three main planar angle of incidence constraints on all three sides of the triangle. For a finite number of such constraints the incidence angle is only enforced approximately. The red line (and n_+) demarks a sample orientation for a possible additional planar constraint at a corner. Minimum (green plane) and maximum (red plane) distance constraints are similar planar constraints on the sampling area. These constraints bound the sampling space, where g can be chosen, on all sides (gray area). b) The vertical camera angle constraints with the relevant corners of the triangle in red are depicted in the upper part, while beneath the partition of the space for convexification is depicted.

The optimization objective for the viewpoint sampling in iteration k, in the case of a rotorcraft UAV, is to minimize the sum of squared distances to the preceding viewpoint g_p^{k-1}, the subsequent viewpoint g_s^{k-1} and the current viewpoint in the old tour g^{k-1}. The former two parts potentially shorten the tour by moving the viewpoints closer together, while the latter limits the size of the improvement step, as g_p^{k-1} and g_s^{k-1} potentially move closer as well.

The resulting convex optimization problem is given below. Its structure as a Quadratic Program (QP) with linear constraints allows the use of an efficient solver [6].

$$\min_{g^k} \quad (g^k - g_p^{k-1})^T (g^k - g_p^{k-1}) + \quad (3)$$
$$(g^k - g_s^{k-1})^T (g^k - g_s^{k-1}) + (g^k - g^{k-1})^T (g^k - g^{k-1})$$

$$\text{s.t.} \quad \begin{bmatrix} n_1^T \\ n_2^T \\ n_3^T \\ a_N^T \\ -a_N^T \\ n^{cam\ T}_{lower} \\ n^{cam\ T}_{upper} \\ n_{right}^T \\ n_{left}^T \end{bmatrix} g^k \succeq \begin{bmatrix} n_1^T x_1 \\ n_2^T x_2 \\ n_3^T x_3 \\ a_N^T x_1 + d_{min} \\ -a_N^T x_1 - d_{max} \\ n^{cam\ T}_{lower} x^{rel}_{lower} \\ n^{cam\ T}_{upper} x^{rel}_{upper} \\ n_{right}^T m \\ n_{left}^T m \end{bmatrix} \quad (4)$$

For the computed optimal position, the heading is determined according to the criterion $\min_{\psi^k} = \left(\psi_p^{k-1} - \psi^k\right)^2 / d_p + \left(\psi_s^{k-1} - \psi^k\right)^2 / d_s$, s.t. $\mathbf{Visible}(g^k, \psi^k)$, where $\mathbf{Visible}(g^k, \psi^k)$ means that from the given configuration, g^k and ψ^k, the whole triangle is visible. d_p and d_s are the Euclidean distances from g^k to g_p^{k-1} and g_s^{k-1} respectively. For simple sensor setups establishing the boundaries on ψ^k

for $\mathbf{Visible}(g^k, \psi^k) = \text{TRUE}$ makes the solution explicit. Otherwise a grid search can be employed.

C. Extensions for Fixed-Wing UAVs

Fixed–Wing UAVs correspond to another excellent configuration for inspection operations. However, their advantages in aspects like long–endurance, come together with limitations on handling sharp turns, steep ascents or descents. Moreover, the direction of a fixed camera is related to the direction of travel. Accordingly, the implementation for the BVS and the viewpoint sampling have to be adapted.

Assuming that highly dynamic maneuvers are avoided for inspection flights, the minimum turn radius of the aircraft is constrained to be $r_{\min}$ while roll and pitch are considered to be near zero. For planning purposes, the xy–plane vehicle dynamics are captured using Dubins curves, thus minimizing the distance w.r.t. $r_{\min}$. Furthermore, in the vertical direction the path is constrained by a maximum climb and sink rate. Since these values are small, instantaneous changes are acceptable and the rate $\dot{z}$ is chosen to be constant along a path segment. If the maximum rate is exceeded, ascending/descending loitering circles are added at the end of the path segment to allow larger changes of height. In many practical cases such as flat landscape coverage, it makes sense to constrain the height of the path to a fixed value to avoid undesirable loitering circles. The fixed-wing UAV is assumed to travel with constant velocity v_{FW} and the path cost is the time $t_{ex} = l_{Path}/v_{FW}$, with l_{Path} the path length.

In contrast to the case of rotorcraft UAVs, where only the distance is minimized in the viewpoint position sampling step, the fixed-wing UAV sampler also aims to align the viewpoints on a as straight line as possible. This effectively avoids too many curly path segments and thus, together with the distance minimization tends to reduce the path length. The addition in the objective is therefore to minimize the squared distance d^2 to the straight line between the neighbouring viewpoints. Using its direction vector b, the distance is calculated as follows:

$$b = \frac{g_s^{k-1} - g_p^{k-1}}{\left\| g_s^{k-1} - g_p^{k-1} \right\|} \tag{5}$$

$$d = \left\| b \times (g^k - g_p^{k-1}) \right\| = \left\| \begin{bmatrix} 0 & -b_3 & b_2 \\ b_3 & 0 & -b_1 \\ -b_2 & b_1 & 0 \end{bmatrix} (g^k - g_p^{k-1}) \right\|$$

and with

$$\begin{bmatrix} 0 & -b_3 & b_2 \\ b_3 & 0 & -b_1 \\ -b_2 & b_1 & 0 \end{bmatrix} (g^k - g_p^{k-1}) = q \tag{6}$$

follows $d^2 = q^T q$. To avoid the insertion of unnecessary circles, the distance between the viewpoints has to be large enough according to their heading, the direction to the next viewpoint and $r_{\min}$. The bounds on that distance $l_i,\ i = \{p, s\}$ are derived geometrically and evaluated using numerical algorithms. The distance criterions are therefore $(g^k - g_i^{k-1})^T (g^k - g_i^{k-1}) \geq {l_i}^2,\ i = \{p, s\}$ which are non-convex. To convexify, the criterions are linearized around the old viewpoint. This adaptation is conservative by the exclusion of the non-convex part and attenuates the impact of the extrapolation error:

$$(g^k - g_i^{k-1})^T (g^{k-1} - g_i^{k-1}) \quad \geq \quad {l_i}^2,\ i = \{p, s\} \tag{7}$$

Wrapping all up in a single QP-formulation and adding the two slack variables ϵ_p and ϵ_s with constant C to allow occasional violation of the minimal distance criterion:

$$\min_{q, g^k} \quad \text{(objective in (4))} + q^T q + C(\epsilon_p + \epsilon_s) \tag{8}$$

$$\text{s.t.} \quad \begin{bmatrix} 0 & -b_3 & b_2 \\ b_3 & 0 & -b_1 \\ -b_2 & b_1 & 0 \end{bmatrix} (g^k - g_p^{k-1}) = q$$

$$\begin{bmatrix} \text{constraints in (4)} \\ (g^{k-1} - g_p^{k-1})^T \\ (g^{k-1} - g_s^{k-1})^T \end{bmatrix} g^k \succeq \begin{bmatrix} \text{constraints in (4)} \\ {l_p}^2 - \epsilon_p \\ {l_s}^2 - \epsilon_s \end{bmatrix} \tag{9}$$

$$\epsilon_p \geq 0$$

$$\epsilon_s \geq 0$$

The criterion of Equation (7) is inverted for the heading computation and applied as long as a feasible solution is found. The proposed approach also works efficiently in case of obstacles up to some complexity by dividing the sampling space in convex pieces that are evaluated individually. In our implementation obstacles are approximated with cuboids.

D. Additional Heuristic Concepts

Additional heuristic measures increase the quality of computed paths. These primarily concern the rotorcraft UAV path planning. In order to allow a faster and more rigorous ordering of the viewpoints, initial iterations of the algorithm consider not the nearest neighbour on the tour to minimize the distance to, but neighbours that are $N_{Neighbour}$ away on both sides. $N_{Neighbour}$ is then decremented in every iteration to finally reach 1. To further improve the viewpoint ordering, the allowable yaw rate is set lower in the initial iterations and then slowly increased to reach the maximally allowed $\dot{\psi}_{max}$.

V. Computational Analysis

In order to evaluate the capabilities of the proposed algorithm, a simple and scalable scenario is used. An array of equilateral triangles is arranged in a plane as shown in Figure 3, with a height of 1250m and a width of $\frac{\sqrt{3}}{2}$2500m. This corresponds to an area of 2.71km^2. This area is filled with a variable number of equilateral triangles, effectively corresponding to different mesh resolutions and thus numbers of viewpoints. The mesh resolution is varied to examine the impact on the quality of the resulting path, while the number of viewpoints is a meaningful parameter of the problem complexity and the time the algorithm needs for execution.

TABLE I: Scalable Inspection Scenario

N_{facets}	[100...3600]	∠incidence	30°
FoV	$[70, 70]^\circ$	Mouting pitch	25°
Range	unconstrained	Height	200m
r_{min}	60m	v_{FW}	9m/s
$v_{\max}$	5m/s	$\dot{\psi}_{\max}$	0.5rad/s

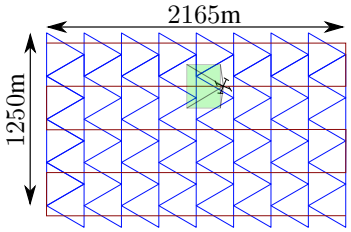
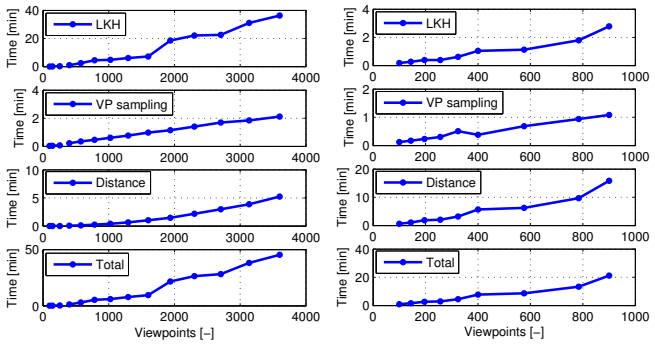


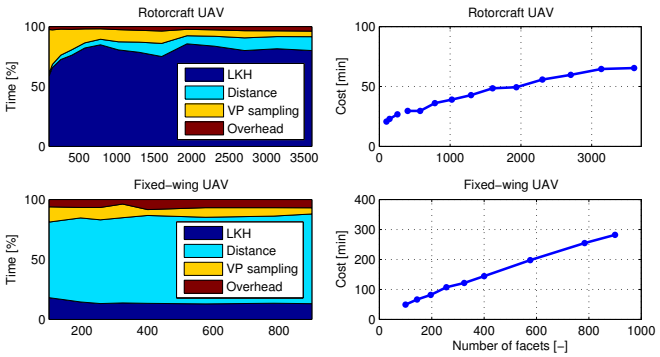
Fig. 3: Illustration of the triangular pattern that is used in different resolutions for the following analysis of the algorithm's characteristics. Overlayed in brown is a naive sweeping path with a certain base line (in this case the same as the triangle edge length).



(a) Rotorcraft UAV (b) Fixed-wing UAV

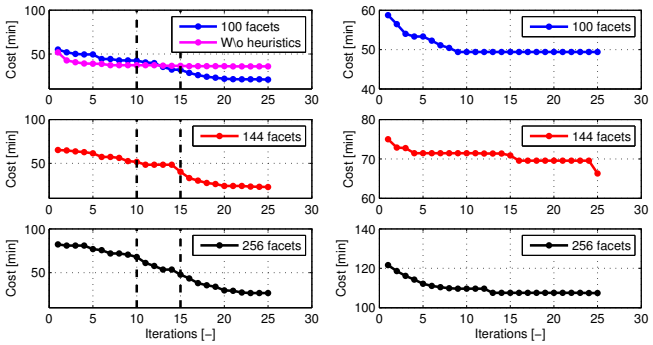
Fig. 4: Correlation of the number of viewpoints for both systems with the computational time consumption. The time consumption curve is given individually for different components of the algorithm.

Both for the rotorcraft and the fixed–wing UAV cases, the inspection is performed from a constant height of 200m above ground, with a minimum incidence angle of 30°. The employed camera is mounted with a pitch of 25°, the FoV is 70° in both vertical and horizontal directions, while a margin for robustness is deducted in the horizontal direction, thus leading to an effectively usable opening of 60°. This reduction from the nominal FoV is done in all test cases. The rotorcraft UAV is assumed to move with a maximum translational speed of 5m/s and a maximum yaw rate of 0.5rad/s, while the minimum turn radius for the fixed-wing is 60m. Simulations for both models, using triangular patterns with variable numbers of facets, were performed on a computer with a 1.73GHz processor running Ubuntu 14.04 and using a single-thread C++ implementation. The time consumption for the computation is depicted in Figure 4 for the rotorcraft and the fixed–wing UAV cases. It shows accumulated time consumptions of the different parts of the algorithm, as well as the total, while Figure 5a depicts the relative shares. The time complexity of the algorithm for large numbers of viewpoints is dominated by the LKH for which [10] gives a time complexity of $\mathcal{O}(N^{2.2})$, where N is the number of viewpoints. Less dominant is the viewpoint sampling, with a complexity of $\mathcal{O}(N)$, since the constant complexity method of sampling has to be repeated for every viewpoint. Lastly, the complexity of the distance computations is $\mathcal{O}(N^2)$, since



(a) Relative time consumption (b) Resolution dependent cost

Fig. 5: Figure 5a depicts the relative time consumption of different parts of the algorithm, while Figure 5b shows the cost of the computed paths for different amount of facets, corresponding to varying mesh resolution.



(a) Rotorcraft UAV (b) Fixed-wing UAV

Fig. 6: Improvement of the path cost over the course of 25 iterations for 100, 144 and 256 facets. The left top figure depicts a run without heuristics and one with the heuristic on the viewpoint neighbours (running for the first 10 iterations) and the heuristic on the viewpoint yaw (running for the first 15 iterations). As shown, for the fixed–wing UAV case a less smooth path improvement procedure takes place after the first iterations due to the additional complexity introduced by the nonholonomic constraints.

the distance has to be computed for all pairs of viewpoints. The predicted behaviour of the viewpoint sampling and distance computation complexities can be observed in the plots for the rotorcraft UAV case (Figure 4a), which contain scenarios for up to 3600 viewpoints. For lower numbers of viewpoints 25 iterations were performed, while for numbers of viewpoints above 900, 50 iterations were performed in order to find high quality paths. Consequently the rotorcraft graphs contain two seperate curves. The fixed–wing UAV graphs in Figure 4b were computed with 25 iterations for numbers of viewpoints up to 900. Overall, the relative time consumption plots in Figure 5a show an increasing share of computation time for the LKH for larger numbers of viewpoints, as could be expected from the theoretical complexities. Evidently, the distance computation, which timewise is insignificant for the rotorcraft UAV case, consumes a large part of the computation time for the fixed–wing case.

This is due to the fact, that the system constraints induce an asymetric TSP, which doubles the amount of viewpoints (corresponding to both directions of travel). While the TSP solver can efficiently handle this, the number of two state boundary value problems is quadrupled. As the number of facets increases, the path is more densely populated with viewpoints, thus attenuating the increase of cost through the larger amount of connections. This can be observed in Figure 6b for both the rotorcraft and the fixed–wing UAV, respectively. To further validate the proposed approach, comparison to a sweeping path as depicted in Figure 3 can be made. With an image base–line of approximately 190m 7 sweeps of 2165m length are necessary to cover the area, to which adds the translation of 6 times the base–line. With a travel speed of 5m/s the resulting path cost is 3259s, which is more than double the cost of what the proposed algorithm computes for a rotorcraft UAV when planning with e.g. 100 facets (1234.90s). This stands as an indication of the performance of the algorithm, the main strength of which is however related with complex 3D scenarios, as those presented in Section VI, where simplified approaches like the sweeping path lose their potential.

Finally, the path length over the course of 25 iterations is depiced in Figure 6 for 100, 144 and 256 facets. The effect of the additional heuristics on finding a better final solution for the rotorcraft UAV case is shown in the top plot of Figure 6a where the cost curve is compared to the one without the heuristics discussed in Section IV-D. Overall, quick progression towards lower–cost paths is achieved for all scenarios, while for the fixed–wing UAV case the curve flattens sooner due to the additional complexity introduced by the nonholonomic constraints.

VI. Evaluation Test–Cases

Within this section advanced evaluation test–cases in simulation and experimental studies are presented. Three challenging experiments are presented, one using a rotorcraft and two with a 5.6m wingspan fixed–wing UAV.

A. Complex 3D Simulation Test–Case

As a complex simulation test–case, a mesh model of the 405m high Central Radio & TV Tower in Beijing was used [3]. The employed mesh contains $N_{facets} = 1701$ triangular facets that model the real building. A rotorcraft vehicle is assumed which is subject to a maximum allowed linear velocity of $v_{\max} = 2$m/s and a maximum yaw rate $\dot{\psi}_{\max} = 0.5$rad/s while it carries a camera sensor mounted with 15° pitch and has a field of view $[120, 120]^\circ$ along the vertical and the horizontal axis respectively. Furthermore, it is enforced that the distance of the camera to the inspected structure is set between 10m and 25m to ensure safety but also close–inspection capable of revealing structural problems (e.g. cracks). The test–case parameters are summarized in Table II. The building and the derived inspection path are illustrated in Figure 7. Eventually, this complex test–study reveals the high–performance characteristics of the proposed inspection planner which are further evaluated in the experimental studies presented in subsections VI-B and VI-C.

TABLE II: Beijing Tower Inspection Scenario

N_{facets}	1701	$\angle$incidence	30°
FoV	$[120, 120]^\circ$	Mouting pitch	15°
$d_{\min}$	10m	$d_{\max}$	25m
$v_{\max}$	2m/s	$\dot{\psi}_{\max}$	0.5rad/s

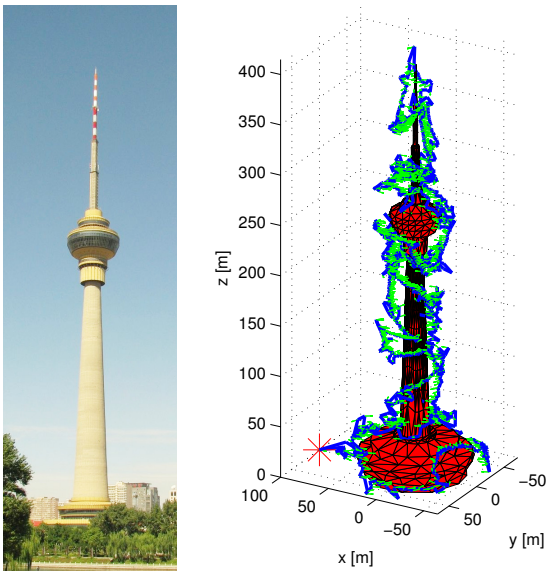


Fig. 7: Large scale structure to be inspected: The 405m high Central Radio & TV Tower in Beijing. The mesh used to compute the path contains 1701 triangular facets. After a computation time of 92s the cost for the inspection is 2997.44s with a maximal speed of 2m/s and a maximal yaw rate of 0.5rad/s. The red point denotes, start– and end–point of the inspection.

B. Rotorcraft UAV Inspection Operations

A first experimental test study was conducted using an AscTec Firefly Hexacopter MAV onboard of which the Visual–Inertial Sensor (VI–Sensor) developed by our lab and Skybotix AG was further integrated. The VI–Sensor integrates 2 HDR global shutter cameras (Aptina MT9V034) and an Analog Devices ADIS16448 IMU in a tightly aligned and synchronized way using an Artix^{TM}–7 FPGA, a Xilinx Zynq 7020 SoC module and an ATOM CPU running Linux. This integrated sensor system runs advanced image processing algorithms, provides complete pose estimates and builds a 3D map of the environment. Figure 8 shows the employed UAV equipped with the VI–Sensor.

The experimental setup refers to the inspection of the 3D structure shown in Figure 9 and consists of 106 facets capturing the structure in detail. The scenario is further complicated by a bounding box of $3\times3\times2.75$m and a sensing minimum range $d_{\min} = 1$m. This inspection structure was offline reconstructed from a set of terrestrial images and consequently a mesh was computed to be utilized by the proposed inspection path planner. Table III summarizes the experiment parameters. Using the proposed inspection path

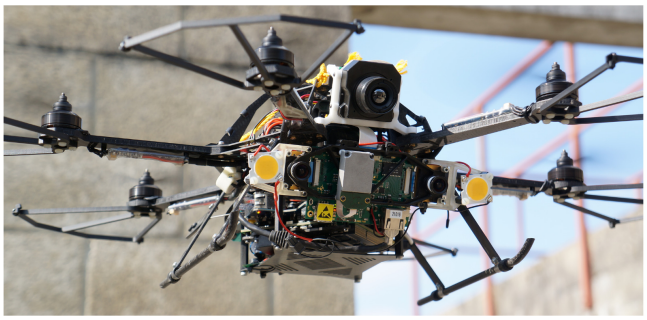


Fig. 8: The Firefly UAV equipped with the VI–Sensor.

TABLE III: Rotorcraft UAV Inspection Scenario

N_{facets}	106		
∠incidence	30°	Bounding box	3x3x2.75m
FoV	$[60, 90]°$	Mouting pitch	15°
$d_{\min}$	1m	$d_{\max}$	3m
$v_{\max}$	0.25m/s	$\dot{\psi}_{\max}$	0.5rad/s

planner, a path that guarantees complete coverage is derived and has a total length of 151.44s. Reconstruction results derived using pose–annotated (position and rotations) image sequences from one of the VI–Sensor cameras and the Pix4D software indicate excellent 3D reconstruction results, a fact that further increases confidence on the practical applicability of the proposed algorithm. The reference path and the recorded flight response along with the reconstruction results are shown in Figure 9. The arrows indicate the reference viewpoints proposed by the inspection planner.

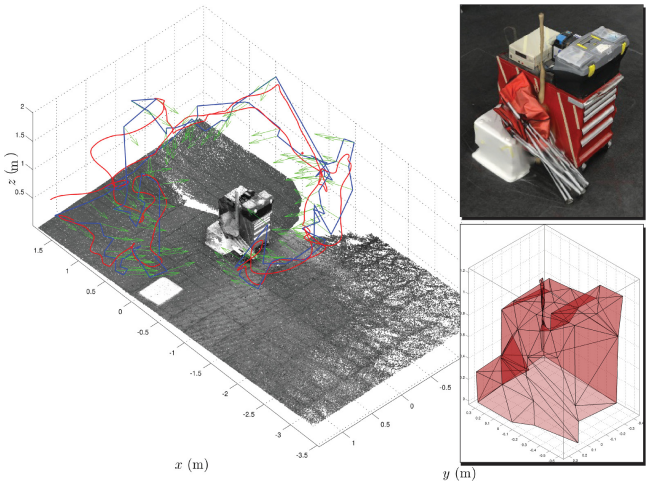


Fig. 9: Experimental study of the inspection of a trolley. The preliminary, terrestrial images–based, 3D reconstruction of the inspection structure is depicted and was used to derive a simplified mesh that was then employed by the inspection path planner to compute the inspection path shown in the Figure. The path cost is 151.44s for $v_{\max} = 0.25$m/s and $\dot{\psi}_{\max} = 0.5$rad/s .

C. Fixed-Wing UAV Inspection Operations

A second set of experiments was conducted using a long endurance fixed–wing UAV platform developed by our lab. The particular platform, AtlantikSolar [20], is a 5.6m wingspan, 7.5kg, solar–powered vehicle with robust state–estimation capabilities [22], automatic trajectory tracking control [19] and further integrates a) an advanced sensor pod with a monocular version of the aforementioned VI–Sensor with the Aptina MT9V034 camera mounted at a 50° front–down oblique view and every image is fully pose–annotated as well as b) a GPS–tagged Sony HDR-AS100VW camera. Figure 10 depicts the UAV as well as the sensor pod.

TABLE IV: Marche–en–Famenne Inspection Scenario

N_{facets}	8, 8	∠incidence	30°
v_{FW}	9m/s	$r_{\min}$	60m, 60m
FoV	$[90, 50]°$, $[120, 120]°$	Mouting pitch	50°, 90°
Range	unconstrained	Height	120m, 100m

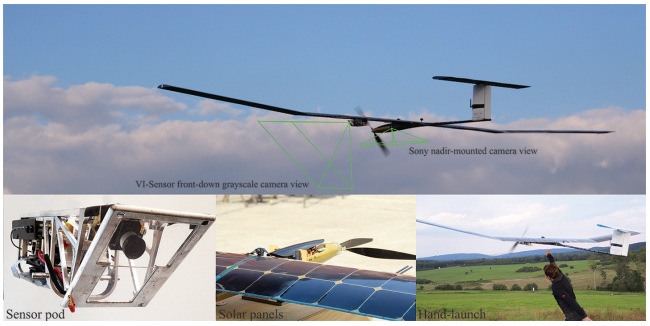


Fig. 10: The AtlantikSolar UAV with the sensor pod attached to its wings and further photos of the sensor pod, the solar cells and an instant of the hand–launching.

With this UAV corresponding to an excellent test–case for nonholonomic inspection path planning, two inspection missions were designed to be conducted within the framework of the ICARUS project field–trials in the area of Marche–en–Famenne in Belgium [13]. Geographical Information System (GIS) data were used to derive a first, rough, 8 facets ($N_{facets} = 8$) mesh of the area and subsequently two inspection paths were computed, one for the oblique view grayscale camera of the VI–Sensor with a fixed reference altitude set at an absolute value $z^r = 362$m (corresponding to 120m above the highest point to be inspected) and the other for the nadir–mounted Sony HDR-AS100VW with $z^r = 342$m, while the modelled minimum turning radius was $r_{\min} = 60$m. Table IV summarizes the parameters used for the two experiments. Figures 11 and 12 present the results for the two camera configurations. In both cases, the optimized reference inspection path, the real recorded UAV trajectory as well as an offline computed dense point–cloud of the inspection area are shown such that the completeness of coverage is visually assessed. The point clouds are derived using the pose–annotated images and the Pix4D software and are freely available online [2].

For both configurations the proposed inspection planner manages to provide short distance paths that guarantee complete coverage while accounting for the motion constraints of the fixed–wing UAV. The reconstructed point cloud is

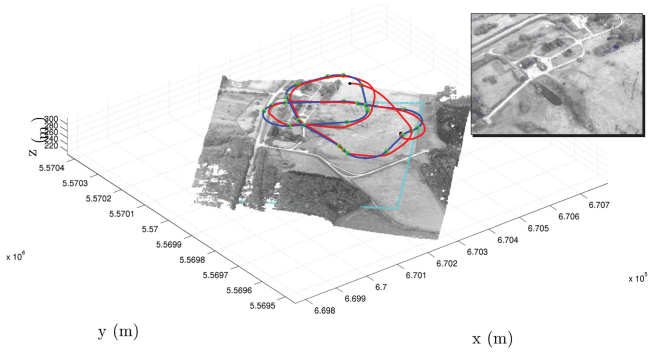


Fig. 11: Inspection path and point–cloud for 3D reconstruction purposes using the front–down mounted view grayscale camera of the VI–Sensor onboard AtlantikSolar. Blue line represents the reference path, green circles are used to indicate the actual waypoints loaded to the autopilot and red is used for the vehicle response. The planner commands the vehicle to navigate such that the camera covers the whole desired area marked with dashed cyan line. A UTM31N coordinate system is employed.

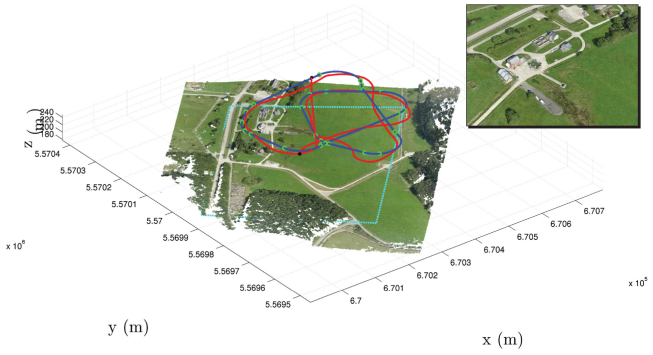


Fig. 12: Inspection path and 3D reconstruction results using the nadir mounted Sony HDR-AS100VW onboard AtlantikSolar. Blue line represents the reference path, green circles are used to indicate the actual waypoints loaded to the autopilot and red is used for the vehicle response. The planner commands the vehicle to navigate such that the camera covers the whole desired area marked with dashed cyan line. A UTM31N coordinate system is employed.

very dense which indicates that such short paths remain practically useful for high fidelity reconstruction purposes where significant overlap is typically required.

VII. Summary & Conclusions

Within this paper, a practically–oriented fast inspection path planning algorithm capable of computing efficient solutions for complex 3D structures represented by triangular meshes was presented. The method was first tested on a scalable scenario and the results were summarized both for the case of a rotorcraft as well as a fixed–wing UAV. Subsequently, the capabilities of the algorithm were demonstrated in real–world application scenarios and experimental studies using both UAV configurations. With the help of 3D–reconstruction software, the recorded inspection data were postprocessed to support the claim of finding full coverage paths and the point cloud datasets are released to enable evaluation of the inspection quality. An implementation of the presented algorithm is accessible [17] for further use and development by the community.

References

[1] E. U. Acar, H. Choset, and J. Y. Lee, "Sensor-based coverage with extended range detectors," *Robotics, IEEE Transactions on*, vol. 22, no. 1, pp. 189–198, 2006.

[2] Accompanying datasets with inspection results in Point-Cloud form. [Online]. Available: http://goo.gl/IUMWQH

[3] Beijing Central TV & Radio Tower Model, 3D Warehouse, "https://3dwarehouse.sketchup.com/."

[4] H. Choset and P. Pignon, "Coverage path planning: The boustrophedon cellular decomposition," in *Field and Service Robotics*. Springer, 1998, pp. 203–209.

[5] G. Dantzig, R. Fulkerson, and S. Johnson, "Solution of a large-scale traveling-salesman problem," *Journal of the operations research society of America*, vol. 2, no. 4, pp. 393–410, 1954.

[6] H. Ferreau, C. Kirches, A. Potschka, H. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.

[7] Y. Gabriely and E. Rimon, "Spiral-stc: An on-line coverage algorithm of grid environments by a mobile robot," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 1. IEEE, 2002, pp. 954–960.

[8] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

[9] H. González-Baños, "A randomized art-gallery algorithm for sensor placement," in *Proceedings of the seventeenth annual symposium on Computational geometry*. ACM, 2001, pp. 232–240.

[10] K. Helsgaun, "An effective implementation of the lin–kernighan traveling salesman heuristic," *European Journal of Operational Research*, vol. 126, no. 1, pp. 106–130, 2000.

[11] S. Hert, S. Tiwari, and V. Lumelsky, "A terrain-covering algorithm for an auv," in *Underwater Robots*. Springer, 1996, pp. 17–45.

[12] F. S. Hover, R. M. Eustice, A. Kim, B. Englot, H. Johannsson, M. Kaess, and J. J. Leonard, "Advanced perception, navigation and planning for autonomous in-water ship hull inspection," *The International Journal of Robotics Research*, vol. 31, 2012.

[13] ICARUS: Unmanned Search and Rescue, "http://www.fp7-icarus.eu/."

[14] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," vol. 30, no. 7, pp. 846–894, 2011.

[15] S. Lin and B. W. Kernighan, "An effective heuristic algorithm for the traveling-salesman problem," *Operations research*, vol. 21, no. 2, pp. 498–516, 1973.

[16] C. Luo, S. X. Yang, D. A. Stacey, and J. C. Jofriet, "A solution to vicinity problem of obstacles in complete coverage path planning," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 1. IEEE, 2002, pp. 612–617.

[17] Open source implementation of the presented algorithm as a ROS package. [Online]. Available: https://github.com/ethz-asl/StructuralInspectionPlanner

[18] J. O'rourke, *Art gallery theorems and algorithms*. Oxford University Press Oxford, 1987, vol. 57.

[19] P. Oettershagen, A. Melzer, S. Leutenegger, K. Alexis and R. Siegwart , "Explicit model predictive control and l1-navigation strategies for fixed–wing uav path tracking," in *22nd IEEE Mediterranean Control Conference*, Palermo, Italy, 2014.

[20] P. Oettershagen, A. Melzer, T. Mantel, K. Rudin, R. Lotz, D. Siebenmann, S. Leutenegger, K. Alexis and R. Siegwart, "A solar-powered hand-launchable uav for low-altitude multi-day continuous flight," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, May 2015, (accepted).

[21] G. Papadopoulos, H. Kurniawati, and N. M. Patrikalakis, "Asymptotically optimal inspection planning using systems with differential constraints," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 4126–4133.

[22] S. Leutenegger, A. Melzer, K. Alexis and R. Siegwart, "Robust state estimation for small unmanned airplanes," in *IEEE Multiconference on Systems and Control (MSC)*, Antibes, France, 2014.

[23] A. Zelinsky, R. A. Jarvis, J. Byrne, and S. Yuta, "Planning paths of complete coverage of an unstructured environment by a mobile robot," in *Proceedings of international conference on advanced robotics*, vol. 13, 1993, pp. 533–538.