2016 International Conference on
Unmanned Aircraft Systems (ICUAS)
June 7-10, 2016. Arlington, VA USA

Distributed Infrastructure Inspection Path Planning for Aerial Robotics subject to Time Constraints

Christos Papachristos[1], Kostas Alexis[1], Luis Rodolfo Garcia Carrillo[1] and Anthony Tzes[2]

***Abstract*— Within this paper, the problem of 3D inspection path planning for distributed infrastructure using aerial robots that are subject to time constraints is addressed. The proposed algorithm handles varying spatial properties of the infrastructure facilities, accounts for their different importance and exploration function and computes an overall inspection path of high inspection reward while respecting the robot endurance or mission time constraints, as well as the vehicle dynamics and sensor limitations. To achieve its goal, it employs an iterative, 3–step optimization strategy within which it first randomly samples a set of possible structures to visit, subsequently solves the derived traveling salesman problem and computes the travel costs, while finally it randomly assigns inspection times to each structure, and evaluates the total inspection reward. For the derivation of the inspection paths per each independent facility, it interfaces a path planner dedicated to the 3D coverage of single structures. The resulting algorithm properties, computational performance and path quality are evaluated using simulation studies as well as an experimental test–case employing a multirotor micro aerial vehicle.**

I. INTRODUCTION

Aerial robotics are getting integrated into a wide variety of critical applications. Among the most promising ones, is that of infrastructure inspection and maintenance operations [1–9]. Aerial robots correspond to a safety–ensuring, efficiency–improving and cost–saving asset that can revolutionize this field. Within such missions, one of the most challenging problems – alongside those of perception, estimation and control – is that of being able to autonomously derive the inspection path such that this handles the structural and spatial distribution properties of the infrastructure, is efficient and respects the often tight endurance limitations of the vehicle as well as any sensor or kinematic constraints that apply. Especially in the case of distributed infrastructure –such as wind farms, solar panels, oil rigs or the power network– the complexity of this problem is particularly high.

The work presented in this paper deals exactly with the problem of distributed infrastructure inspection path planning for aerial robotics that are subject to time constraints. The new algorithm considers the fact that, given a set of spatially distributed *Infrastructure Facilities of Interest* (IFIs) and a robot with time constraints, intelligent inspection path planning should aim to derive the best *possible* path and not be constrained on a potentially infeasible attempt to find a full–coverage solution. Therefore, the proposed algorithm considers an *exploration function* (EF) for each IFI, as well as an *importance weight* (IW), and aims to compute the path that maximizes the totally collected *Inspection Rewards* (IRs) (combination of IWs and EFs) while respecting the vehicle endurance or other mission time constraints as well as the vehicle motion and sensor limitations. It is highlighted that the algorithm is free to select the partial structural inspection of a subset of the spatially distributed IFIs, as long as this maximizes the totally collected IR given that the imposed time constraints do not allow complete inspection of all IFIs. To the authors best knowledge, there is a lack of research activities to address this problem in an autonomous manner despite the notable progress in the efforts to extend the endurance of small aerial robots.

Fig. 1: Photo of an instant of a distributed structural facilities inspection mission.

Subject to specific limiting assumptions, this problem shares certain similarities with the Orienteering Problem (OP) class. Considering constant, time independent, inspection rewards it can be solved using methods for OP [10–12]. Assuming a time–only dependent –and nondecreasing– learning curve, the problem shares similarities with the Traveling Tourist Problem (TTP) and the planning of optimal tourist itineraries given trip time constraints [13–16]. However, in practice, the exploration function for a structure is not a single function of time since two equal duration inspection paths that start at different points or follow different directions can lead to largely different coverage percentages. This fact further increases the complexity of the problem. The proposed algorithm is designed to also be able to deal with this situation. It employs an iterative 3–step optimization paradigm, within which it randomy samples a subset of the facilities that should be visited and partially or fully inspected *(first step)*, subsequently solves the derived Traveling Salesman Problem (TSP) and

The authors are with the University of Nevada, Reno, 1664 N. Virginia Str., 80557, Reno, NV, US, email: kalexis@unr.edu

[2]The author is with the University of Patras, Eratosthenous 6 Str. Patras, Greece

978-1-4673-9333-1/16/$31.00 ©2016 IEEE

computes the travel costs *(second step)*, and finally *(third step)* samples and assigns inspection times and subsets of the full–coverage path of each IFI that respect the time constraints, and then evaluates the overall gained inspection reward. The computation of the full–coverage path per IFI is achieved by interfacing another inspection path planner that is capable of computing optimized inspection paths for a single structure. Any relevant path planner can be employed for this task, while within the framework of this work, an algorithm previously proposed and open–sourced by one of the authors is employed [7].

The proposed approach is thoroughly analyzed using both simulations, as well as an experimental study that employs the autonomous aerial robot shown in Figure 1. Computational analysis is provided with a detailed description of the breakdown of the computational cost for each main algorithmical step. The dataset collected from a large family of simulation results, is publicly released in order to allow easier future comparison with other methods and strategies proposed by the research community [17]. This algorithm will also be open–sourced, including the interfaces to the employed single structure inspection planner, with the aim to release a public and complete infrastructure inspection path–planning framework.

The distributed structural inspection path planning problem is defined in Section II, followed by the detailed description of the proposed approach in Section III. Finally, evaluation studies are presented in Section IV, while conclusions are drawn in Section V.

II. PROBLEM DEFINITION

The problem of distributed infrastructure structural inspection path planning subject to time constraints, as considered in this paper, consists of a) a set of spatially distributed Infrastructure Facilities of Interest (IFIs) that are modeled using 3D meshes, and each of them is associated with an Inspection Reward (IR) that is computed as the multiplication of the Importance Weight (IW) and the Exploration Function (EF), b) a time constraint (either due to the endurance of the robot or mission–specific limits), c) dynamic constraints of the vehicle as well as d) sensor Field–of–View (FoV) limitations. Assuming that for each IFI an *admissible* full–coverage path exists, and can be computed using a dedicated Single Structure Inspection Path–planner (SSIP), the goal is to maximize the total collected IRs by finding the best combination of a subset of IFIs to be visited, associated inspection times for each IFI and corresponding subsets of its full–coverage inspection path as well as the tour among the selected IFIs. Note again, that the problem of SSIP is decoupled and addressed by interfacing a relevant solver such as the one previously proposed by one of the authors [7]. In the following, the addressed problem is defined more formally, while the basic notation is also introduced.

Let $\mathcal{G} = \{\mathcal{V}, \mathcal{A}\}$ be a graph, where $\mathcal{V} = \{v_1, v_2, ..., v_N\}$ is the set of vertices, each one corresponding to the couple of entry and exit pose configurations of the coverage path of one IFI 3D structure $\mathcal{S}_i \in \mathcal{S}$, and $\mathcal{A}$ is the corresponding arc set. Note that incoming connections to a vertex v_i are attached to the entry point of the coverage path of the IFI $\mathcal{S}_i$, while outbound connections start from the exit point of the coverage path. Furthermore, $\mathcal{R} = \{r_i(\mathbf{p}_i^{\alpha,T_i,\mathcal{S}_i})\}, \forall\ \mathcal{S}_i \in \mathcal{S},\ r_i(\mathbf{p}_i^{\alpha,T_i,\mathcal{S}_i}) = w_i f_i(T_i, p_i^\alpha)$ is the set of IR functions for each of the IFIs vertices with w_i being the IW and $f_i(T_i, p_i^\alpha)$ is the EF as a function of the inspection time T_i and starting point p_i^α in the full–coveregate path $\mathbf{p}_i(\mathcal{S}_i)$ of $\mathcal{S}_i$, and t_{ij} is the transition time associated with each arc $a_{ij} \in \mathcal{A}$. The subset of the full–coverage path $\mathbf{p}_i(\mathcal{S}_i)$, that starts from p_i^α and has a duration T_i for a maximum travel speed v_T, is denoted as $\mathbf{p}_i^{\alpha,T_i,\mathcal{S}_i}$. Now let a feasible solution to the problem being the 5–tuple $\mathcal{D} = \{\mathcal{V}', \mathcal{A}', \mathcal{S}', \mathcal{P}', \mathcal{R}'\}$, where $\mathcal{G}' = \{\mathcal{V}', \mathcal{A}'\}$ denotes the hamiltonian path among the sampled vertices $\mathcal{V}' \subseteq \mathcal{V}$ associated with the sampled IFIs set $\mathcal{S}' \subseteq \mathcal{S}$ and the arc set $\mathcal{A}'$, $\mathcal{P}' = \{\mathbf{p}_i^{\alpha,T_i,\mathcal{S}_i}\},\ \forall \mathcal{S}_i \in \mathcal{S}'$ are the sampled subsets of the full–coverage paths for the selected IFIs and $\mathcal{R}' = \{r_i(\mathbf{p}_i^{\alpha,T_i,\mathcal{S}_i})\},\ \forall \mathcal{S}_i \in \mathcal{S}'$ is the relevant set of inspection rewards. The optimal solution to the problem is the 5–tuple $\mathcal{D}_{OPT} = \{\mathcal{V}'_{OPT}, \mathcal{A}'_{OPT}, \mathcal{S}'_{OPT}, \mathcal{P}'_{OPT}, \mathcal{R}'_{OPT}\}$ such that:

$$\mathbf{max} R_{TOT},\quad R_{TOT} = \sum_{i:\mathcal{S}_i \in \mathcal{S}'} r_i(\mathbf{p}_i^{\alpha,T_i,\mathcal{S}_i}) \tag{1}$$

$$\textbf{s.t. } T_{TOT} \leq T_{\max},\quad T_{TOT} = \sum_{i:\mathcal{S}_i \in \mathcal{S}'} (T_i) + \sum_{i,j:\mathcal{S}_i,\mathcal{S}_j \in \mathcal{S}'} t_{ij} \tag{2}$$

Note that, not only the aforementioned time constraint of the mission should be respected, but also the dynamic constraints of the vehicle and the limitations of the sensor as part of the selected SSIP calculations and the Boundary Value Solver (BVS) that is employed for the vehicle dynamics and will be explained in the next section.

III. PROPOSED APPROACH

The proposed Distributed Infrastructure Structural Inspection Planner *(DISIP)* is overviewed within this section. It relies on an iterative, 3–step optimization strategy that allows it to compute inspection paths that improve the gained inspection reward while respecting the imposed time constraints. Within each iteration, the three steps executed are: a) random sampling of a subset of the possible IFIs to be inspected, b) solution of the derived TSP problem and finally c) randomized assignment of inspection times (up to the limit of the time constraint) and corresponding subsets of the full–coverage path for each sampled IFI. As this procedure runs iteratively, the algorithm manages to provide improved solutions over the course of time, while a first solution is available from the first run. These steps are summarized in Algorithm 1 while the most important functions are detailed in the subsequent sections.

A. Computation of the SSIP paths

The DISIP algorithm utilizes and interfaces a Single Structural Inspection Path–planner (SSIP) that is capable of computing a complete coverage path for a given connected structure as well as a sensor model and vehicle kinematic

Algorithm 1 Distributed Infrastructure Structural Inspection Planner (DISIP)

$k \leftarrow 0$
Computation of SSIP paths for all IFIs (Section III-A)
Inspection Structures Set Sampling (Section III-B)
Cost matrix computation (Section III-C)
Solve the TSP problem using the LKH to obtain initial tour (Section III-D)
Inspection Times SSIP paths Sampling (Section III-E)
Inspection Rewards computation (Section III-F)
while running **do**
    Resample Inspection Structures Set (Section III-B)
    Recompute the cost matrix (Section III-C)
    Recompute the best tour among the sampled structures (Section III-D)
    Resample Inspection Times and SSIP paths (Section III-E)
    Recompute Inspection Rewards and update best tour reward R_{best} and best solution $\mathcal{D}_{best} = \{\mathcal{V}'_k, \mathcal{A}'_k, \mathcal{S}'_k, \mathcal{P}'_k, \mathcal{R}'_k\}$ is applicable (Section III-F)
    $k \leftarrow k+1$
end while
Assemble the best DISIP path $\mathbf{d}_{best}$ (Section III-G)
return $\mathcal{D}_{best}$, $\mathbf{d}_{best}$, R_{best}

constraints. Given the maximum travel speed for inspection v_T^I and max yaw rate $\dot{\psi}_{\max}$, the computed full coverage path is timed at each step. Different SSIP algorithms can be used, while within this work the algorithm previously proposed by one of the authors [7] is employed both due to its performance as well as due to its open–source availability. The algorithm is executed once per IFI type, considers a triangular mesh representation of the structure and computes the path via an optimization method that alternates between two steps, namely a step that samples and computes new viewpoints such as to reduce the cost–to–travel between each viewpoint and its neighbors, and a step that computes the optimal connecting tour for the current iteration. The specific planner, supports both rotorcraft as well as fixed–wing aerial vehicles, and accounts for their different motion model. Furthermore, its computational cost is particularly small due to the convexification of the viewpoint computation problem and the utilization of the Lin–Kernighan–Helsgaun (LKH) [18] solver for the tour computation step. Computational analysis is included in the relevant publication [7], while demo scenarios and experimental datasets are available online.

B. Inspection Structures Set Sampling

Within the IFIs set $\mathcal{S}$ and the associated graph vertices $\mathcal{V}$, the subsets $\mathcal{S}'_k \subseteq \mathcal{S}$, $\mathcal{V}'_k \subseteq \mathcal{V}$ are randomly sampled at each iteration k. The rest of the steps of the DISIP algorithm are now executed based on this set of IFIs and the corresponding graph $\mathcal{G}_k = \{\mathcal{V}'_k, \mathcal{A}_k\}$, where $\mathcal{A}_k$ is the set of all possible connections between the members of $\mathcal{V}'_k$.

C. Cost Matrix Computation

The paths among all $v_i \in \mathcal{V}'_k$ are computed via the utilization of a two state BVS applicable to rotorcraft aerial robots. This BVS is employed to connect two v_i, v_j based on the start and end points of the corresponding sampled inspection path of the two IFIs $\mathcal{S}_i, \mathcal{S}_j$ respectively (asymmetric connections). It is noted that, the employed BVS of a rotorcraft aerial robot consists of position as well as yaw $\xi = \{x, y, z, \psi\}$, while as long as low speeds are considered, roll and pitch are approximated to be zero. Consequently, the path from configuration ξ_i to ξ_j is given by $\xi(s) = s\xi_1 + (1-s)\xi_0$, where $s \in [0,1]$. The two limitations considered are the maximum translational velocity constraint v_T and the yaw rate constraint $\dot{\psi}_{\max}$. The resulting execution time is $t_{ij} = \mathbf{max}(\delta/v_T, \|\psi_j - \psi_i\|/\dot{\psi}_{\max})$ where δ denotes the Euclidean distance. The cost of a path segment within the DISIP algorithm execution corresponds to its travel–execution time t_{ex}. This calculation is conducted for all possible connections between the sampled IFIs of any iteration and the corresponding cost matrix $\mathcal{C}_k$ is derived. As a remark, it is noted that a local collision–free point–to–point navigation algorithm such as RRT* may be used to avoid collision with the IFI from which a connection departs, arrives or other IFIs in between.

D. TSP Tour Solution

Given the cost matrix $\mathcal{C}_k$, the next goal is to compute the hamiltonian path $\mathcal{G}'_k = \{\mathcal{V}'_k, \mathcal{A}'_k\}$, where $\mathcal{A}'_k$ is the arc set of the hamiltonian path. Essentially, at this step the algorithm derives the solution to the (possibly asymmetric) TSP problem of visiting the sampled vertices $\mathcal{V}'_k$ of the sampled IFIs set $\mathcal{S}'_k$. Among the multiple algorithms that have been proposed for the TSP, the approach of Lin–Kernighan [19] and its methodology of implementation by Keld Helsgaun (LKH solver) [18] corresponds to the best known local search algorithm solution. The LKH solver relies on the concept of λ–optimality, according to which a tour is said to be λ–optimal (or simply λ–opt) if it is impossible to obtain a shorter tour by replacing any λ of its links by any other set of λ links. Similarly, a λ–opt neighborhood for tour χ, $\mathcal{N}_k^\lambda(\chi)$, consists of all tours which can be constructed by deleting and adding λ edges. Based on the observation that two hamiltonian cycles only differ in λ edges ($2 \leq \lambda \leq N_k$) (N_k the sampled number of IFIs at the k–th iteration), i.e. $\chi \in \mathcal{N}_k^\lambda(\chi),\ \forall\chi$ and in order to address the problem that λ–optimality can be tested in $\mathcal{O}(N_k^\lambda)$ while λ is unknown, LKH employs the alternative of choosing an *efficient searchable* neighborhood such that λ can be chosen dynamically. This is the concept of *sequential* λ–opt *moves*. More formally, a λ–opt move is called sequential if it can be described by a path alternating process between deleted and added edges. The code implementation of the LKH solver is found online at [20]. Using such a solver, the derived TSP problem from the sampled graph $\mathcal{G}'_k$ is computed, leading to the tour τ_k (the solution to the problem of finding the hamiltonian path $\mathcal{G}'_k = \{\mathcal{V}'_k, \mathcal{A}'_k\}$) which represents the

sequence based on which the sampled IFIs should be visited as well as the travel costs of this route T^{τ_k}.

E. Inspection Times and SSIP paths Sampling

For each IFI $\mathcal{S}_i$, a SSIP path $\mathbf{p}_i(\mathcal{S}_i)$ is computed, has m_i number of points and $m_i - 1$ path segments, traveled in time based on the same BVS described above. Given the overall SSIP path duration T_i of each IFI, an inspection time $T_i^k \leq T_i$ for each IFI $\mathcal{S}_i \in \mathcal{S}'_k$ and DISIP iteration k is sampled while accounting that the overall assigned time $\sum_{i:\ \mathcal{S}_i \in \mathcal{S}'} T_i^k$ should be less than the available time $T_{\max} - T^{\tau_k}$. As by random sampling, a set of T_i^k that would overall lead to $\sum_i T_i^k = T_{\max} - T^{\tau_k}$ exactly is unlikely, an additional step of adjusting the assigned times by an amount proportional to the ratio of the sampled T_i^k and leads to the satisfaction of the aforementioned equation takes place. Based on the T_i^k values, the algorithm further samples an entry $p_i^{\alpha_k}$ point to each of the SSIP paths $\mathbf{p}_i(\mathcal{S}_i)$ as well as an exit point $p_i^{\beta_k}$ such that the corresponding subset of the SSIP path $\mathbf{p}_i^{\alpha_k, T_i^k, \mathcal{S}_i}$ has an overall duration time as close as possible (given the discrete points of the SSIP path) to T_i^k. Minor adjustments (interpolation) may take place afterwards to account for the discretized nature of the SSIP paths.

F. Inspection Reward Computation

For a SSIP path $\mathbf{p}_i(\mathcal{S}_i)$, each point corresponds to a viewpoint configuration that covers a specific subset of the IFI structure model (one of its triangular mesh faces). Therefore, for the sampled subset of each SSIP path $\mathbf{p}_i^{\alpha_k, T_i^k, \mathcal{S}_i}$, the overall covered area of the IFI $\mathcal{E}_i^k$ can be computed which then allows the computation of the coverage ratio $\gamma_i^k = \mathcal{E}_i^k / \mathcal{E}_i$ (where $\mathcal{E}_i$ the overall area of the i–th IFI) can be computed. Once this value is computed for all sampled inspection paths for the given set of IFIs within the k–th iteration, the inspection reward of each path is computed as $r_i^k(\mathbf{p}_i^{\alpha_k, T_i^k, \mathcal{S}_i}) = w_i \mathcal{E}_i^k$. Finally, the total inspection reward is computed as $R_{TOT}^k = \sum_{i:\mathcal{S}_i \in \mathcal{S}'_k} r_i^k(\mathbf{p}_i^{\alpha_k, T_i^k, \mathcal{S}_i})$. If this reward is higher than the previous best value R_{best}, then the k–th iteration of the algorithm is considered as the so–far best solution and the reward value gets updated $R_{best} \leftarrow R_{TOT}^k$. Furthermore, the elements of this best DISIP solution $\mathcal{D}_{best} = \{\mathcal{V}'_k, \mathcal{A}_k\prime, \mathcal{S}_k\prime, \mathcal{P}_k\prime, \mathcal{R}_k\prime\}$ are exported. Note that this 5–tuple contains the information for the best TSP tour $\tau_{best} \leftarrow \tau_k$ within the hamiltonian path $\mathcal{G}'_k = \{\mathcal{V}'_k, \mathcal{A}'_k\}$.

G. Assembly of the DISIP path

At this final step, which is performed after the execution of all the iterations of the DISIP algorithm, the SSIP paths $\mathcal{P}'_k$ and the connecting segments of the TSP tour τ_{best} are combined to form the overall DISIP path. This connected timed path $\mathbf{d}_{best}$ is the result used for the guidance of the aerial robot.

H. DISIP Performance

The proposed solution to the problem of distributed infrastructure structural inspection path planning subject to time constraints is motivated from the need to provide admissible inspection paths that respect all the constraints, are computed fast, while given that more time is available for the mission preparation, updated and improved solutions can be found. A fundamental design consideration of the algorithm was that we can safely assume that the amount of distributed facilities to be inspected by a single robot cannot be arbitrary large but can be of any geometrical shape and spatial distribution. As a response to that, the designed algorithm does not possess the feature of optimality, but produces high quality paths by randomized sampling of the search space. The derived paths always make use of all the time that is available to the robot – even from the first solution. As each iteration requires a very short computation time, its iterative execution quickly leads to paths of very high quality. Furthermore, the algorithm makes no constraining assumption on the nature of the exploration function and uses any possible full–coverage path for each of the distributed structures from which it extracts subsets based on the sampled time assignments. To the authors best knowledge this is the first attempt of its kind while the provided simulation and experimental studies further demonstrate its efficiency and suitability for the problem.

IV. EVALUATION STUDIES

In order to enable thorough evaluation of the properties and performance of the proposed distributed infrastructure inspection path planner, a wide set of evaluation test–cases both in simulation as well as using experimental studies were considered. Within those, the goal is to assess the quality of the inspection path for different spatial sparsity and distribution of the infrastructure as well as different robot endurances or mission time constraints and finally the computational properties of the algorithm. The experimental studies were conducted using a quadrotor MAV that is capable of autonomous navigation using a Visual–Inertial localization and mapping framework [21].

A. Studies in Simulation

In order to evaluate the properties of the DISIP algorithm and especially the quality of the inspection paths for different levels of spatial sparsity of the IFIs, we followed a process of creating multiple randomized distributed infrastructure scenarios. All scenarios consist of a combination of four (4) different infrastructure facilities and more specifically, a) a solar–panels park, b) a vessel tank, c) a turbine storage and a d) distribution network power transformer. These structures correspond to a variety of sizes and inspection complexities. Having these four building blocks at hand, 40 simulations were considered and specifically 10 for each case of $8, 16, 32, 64$ of these structures (randomly selected) at an overall area with planar dimensions $[\Delta x, \Delta y, \Delta z] = [200, 200, 50]$m (for 8 or 16 structures) or $[\Delta x, \Delta y, \Delta z] = [400, 400, 100]$m (for 32 or 64 structures). As the same size of overall area was used for different amount of structures, this also allowed to evaluate different infrastructure densities. Furthermore, it is pointed out that for the smaller scenarios, an aerial robot with an endurance of $T_{\max} = 1800$s, a

traveling speed $v_T = 3$m/s, and inspection speed $v_T^I = 1$m/s and a yaw rate constraint $\dot{\psi}^{\max} = 0.5$rad/s was considered (values achievable by a large multirotor), while for the larger scenarios the endurance was set to $T_{\max} = 5400$s (achievable by a gasoline–powered helicopter UAV). Regarding, the assumed on–board camera sensor, all scenarios were conducted assuming a FOV of $65°$ in both dimensions. Without loss of generality, all importance weight (IW) factors except those refering to the solar panel park are set to one, while for the solar panel farm it is set to 0.25.

The first step of the DISIP algorithm for all these scenarios is to interface SSIP and compute the full–coverage inspection paths for each structure present at the scenario. Figure 2 presents these coverage paths for all the considered structures possibly available in the aforementioned scenarios.

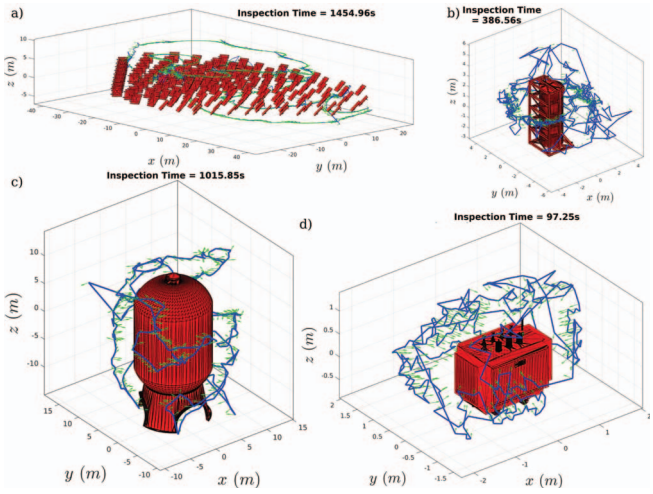

Fig. 2: Results of the execution of the SSIP algorithm for each structure. The DISIP algorithm interfaces SSIP and uses these full–coverage path to extract a subset of them based on the time assigned for each structure and the expected inspection reward.

Having the SSIP results at hand, DISIP can proceed to its further calculations as described in Section III. In the following, two simulation scenarios will be presented in detail, while summarized statistical data from all the 40 simulations are provided subsequently. The complete set of simulation results data may be found online [17].

The first result to be analytically presented, refers to the case of a robot with $T_{\max} = 1800$s endurance and its other parameters as previously defined, while an 8 IFIs scenario is considered. The best computed inspection result, after $k = 30$ DISIP iterations, is shown in Figure 3, while statistical analysis is shown in Figure 4.

The second result refers to the same area size and robot endurance but this time populated with 16 randomly distributed structures. This essentially corresponds to a much denser and more challenging scenario. The derived result, after $k = 30$ iterations of the algorithm, is shown in Figure 5. The relevant statistical analysis is shown in Figure 6.

To provide broader statistical insight on the computational capabilities of the algorithm, such simulations were conducted for 4 sets, each with 10 scenarios of $8, 16, 32, 64$ of

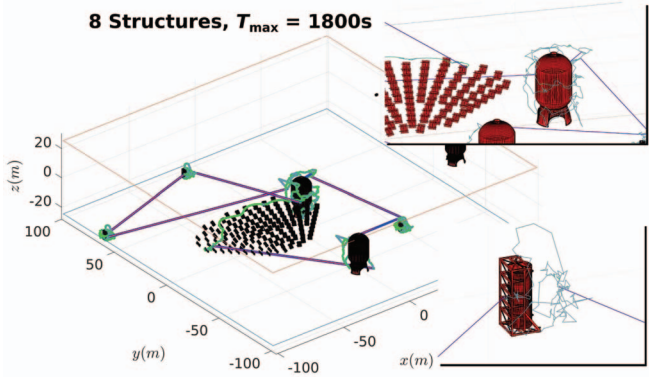

Fig. 3: Distributed infrastructure inspection result for the case of a large area with 8 structures and a robot endurance of $T_{\max} = 1800$s. The distribution of the structures is based on a randomized spatial generator.

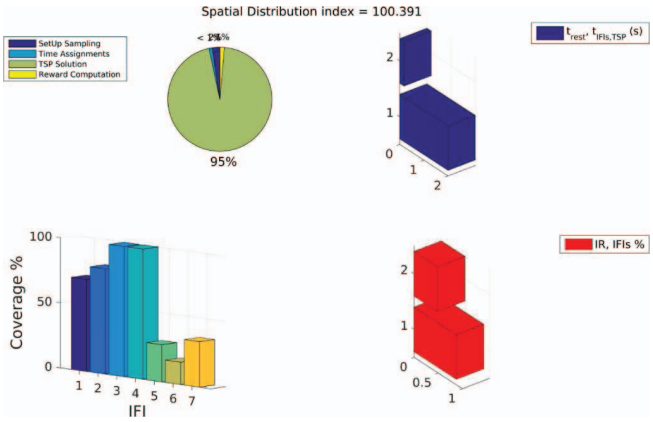

Fig. 4: Computational analysis of the a distributed infrastructure inspection result using the proposed algorithm for the case of a large area with 8 structures and a robot endurance of $T_{\max} = 1800$s. The term spatial distribution index denotes the average distance from IFI to IFI. The upper right plot denotes the time for the IFI sampling and TSP computations (lower bar) and all the rest (upper bar). The bottom right plot denotes the percentage of IFIs visited (lower bar) and the percentage of awards collected (upper bar). The upper left pie denotes the percentage of the basic algorithmical step and as shown the TSP solution consumes the majority of the time. Finally, the bottom left plot denotes the percentage of coverage for each of the sampled IFIs.

these structures in smaller and larger areas. The extracted averaged statistical results are presented and discussed in Figure 7, while the complete set of results is available online [17]. Note that in this plot the depicted times are per iteration of the DISIP algorithm.

B. Experimental Studies

The proposed DISIP algorithm was additionally evaluated experimentally for the case of a simplified, downsized set–up consisting of four structures, namely a power transformer mock–up and assemblies of boxes with different geometries. The distributed inspection mission is conducted using a small aerial robot capable of flying autonomously in GPS–denied environments by relying on its perception module. More specifically, this custom–designed robot is equipped with a

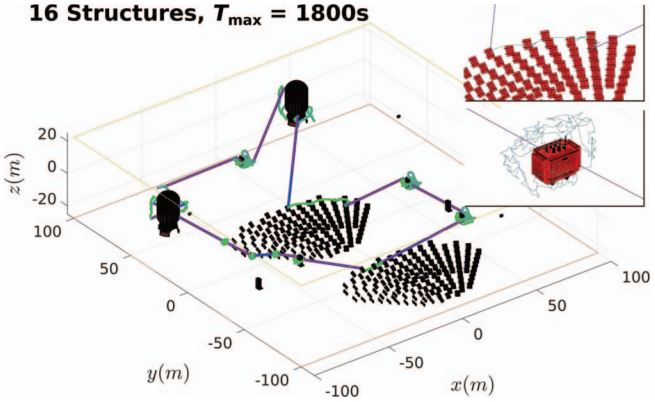

Fig. 5: Distributed infrastructure inspection result for the case of a large area with 16 structures and a robot endurance of $T_{\max} = 1800$s. The distribution of the structures is based on a randomized spatial generator.

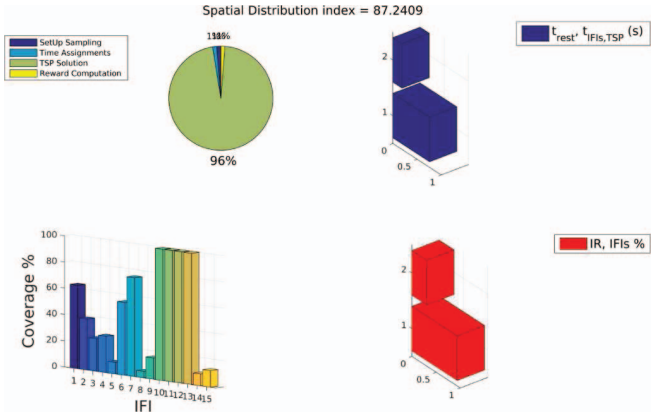

Fig. 6: Computational analysis of the a distributed infrastructure inspection result using the proposed algorithm for the case of a large area with 16 structures and a robot endurance of $T_{\max} = 1800$s. The term spatial distribution index denotes the average distance from IFI to IFI. The upper right plot denotes the time for the IFI sampling and TSP computations (lower bar) and all the rest (upper bar). The bottom right plot denotes the percentage of IFIs visited (lower bar) and the percentage of awards collected (upper bar). The upper left pie denotes the percentage of the basic algorithmical step and as shown the TSP solution consumes the majority of the time. Finally, the bottom left plot denotes the percentage of coverage for each of the sampled IFIs.

Pixhawk ARM M4–based autopilot which is interfaced by an ODROID–U3 Quad–Core embedded computer runing a lightweight linux distribution and acting as a high-level unit. Apart from interfacing the Pixhawk autopilot, it receives the feeds of a ground–pointed PS3 Eye camera operating at 125Hz at a 320×240 resolution. Based on custom–developed Optical Flow and Homograpphy estimation algorithms running at 100Hz and with proper data fusion with the IMU updates, the UAV is capable of achieving 6–Degrees of Freedom pose estimation autonomously. Finally, at the position control level a Model Predictive Control Strategy is employed. With this lower–layer of autonomy deployed, this aerial robot further employs a stereo vision pair consisting of two PS3 Eye cameras mounted on a 3D–printed frame and

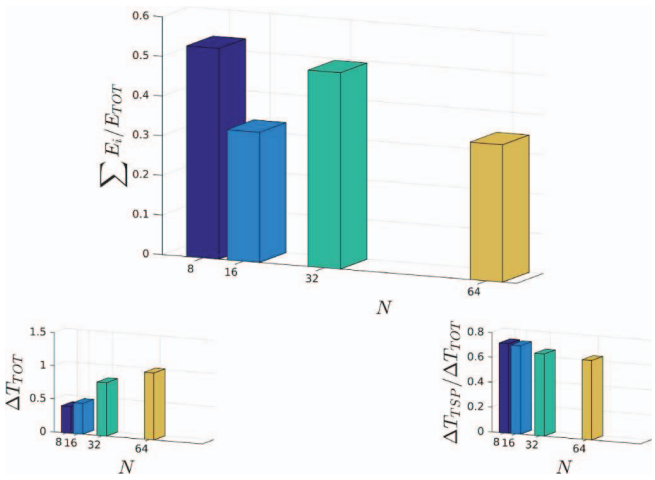

Fig. 7: Averaged statistics result on the percentage of the overall coverage (top plot), the total computation time per iteration (bottom left) and the ratio of computational load due to the TSP computations (bottom right). Interestingly, it is shown that for the same robot endurance, doubling the amount of IFIs leads to a similar drop of percentage coverage for the two cases of endurance considered. Furthermore, the increase in computational time is almost linear while the TSP computations hold a major and only slowly decreasing percentage of the computations as the amount of IFIs increase.

are time syncronized. Using an additional ODROID–XU3 Octa–Core single board computer dedicated to the execution of the simultaneous localization and mapping framework overviewed in [21], autonomous navigation is achieved.

Employing this aerial robot, a downsized distributed infrastructure inspection scenario is conducted. Assuming a limited endurance of $T_{\max} = 300$s and deploying the aforementioned structures on a limited space, DISIP was request to find an distributed inspection solution that leads to high inspection reward while respecting the endurance of the robot. Note that for this run, the FoV of the PS3 Eye camera ($[40, 40]$deg) is respected while a maximum inspection and traveling velocity $v_T = v_T^I = 0.1$m/s is set in combination with a maximum yaw rate $\dot{\psi}_{\max} = 5$deg/s. Figure 8 depicts the considered scenario alongside with the experimentally recorded trajectory of the robot. In addition, Figure 9 shows instances of the online reconstructed octomap voxed–based representation of the environment as well as different views of the dense point cloud derived using postprocessing methods [22]. As shown, the real–time computed octomap, which relies on the visual–inertial pipeline of the robot, provides a representation of the environment that facilitates autonomous and collision–free navigation, while the offline reconstructed dense point cloud indicates the sufficient quality of the inspection result for the subsets of the full–coverage path that the DISIP algorithm employed. It is acknowledged that this experimental scenario is not considered to reflect the full–scale of the overall capabilities of the DISIP algorithm due to its small scale, the size of the environment and the endurance of the robot. However, it indicates the applicability of such strategies in terms of inspection quality and handling the

vehicle and sensor limitations and models. Furthermore, it indicates the potential provided by autonomously navigating aerial robots when these are combined with powerful path planning algorithms.

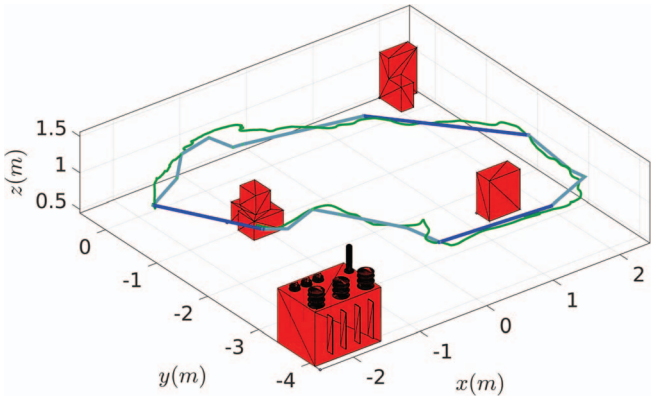

Fig. 8: Experimental downsized distributed infrastructure inspection scenario with three structurs (a power transformer mockup and three sets of boxes) conducted using an autonomous quadrotor navigating in a GPS–denied environment.

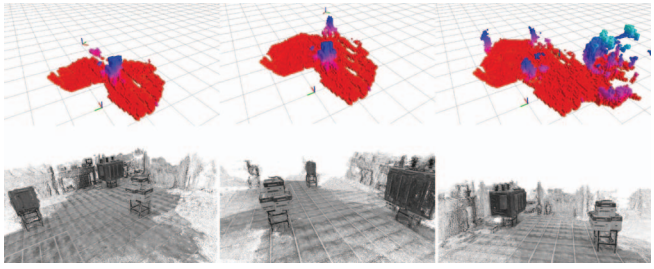

Fig. 9: Instances of a) the progressively–built octomap of the environment which is executed online to facilitate autonomous collision–free navigation and b) of the offline reconstructed dense point cloud of the inspection scenario and its structures.

V. CONCLUSIONS

A practically–oriented algorithm that addresses the problem of distributed infrastructure structural inspection path planning subject to time constraints was proposed within this paper. To derive its solution, the proposed approach relies on an iterative, 3–step optimization paradigm via which it computes first feasible solutions very fast while given that more time is available the solution quality increases. As shown, the algorithm respects the imposed time constraints as well as the vehicle dynamics and sensor limitations. Via an extensive set of simulation scenarios, the path quality, low computational load and scalability regarding the size of the problem and the density of the infrastructure are demonstrated and statistically evaluated. Furthermore, an experimental study using an autonomous quadrotor illustrates the fact that the algorithm fits to the tight constraints of a small aerial robot and performs with high efficiency. A public dataset accompanies this submission, while upon possible acceptance of the paper, the code will be released to the public for further use and development from the community.

REFERENCES

[1] N. Metni and T. Hamel, "A uav for bridge inspection: Visual servoing control law with orientation limits," *Automation in Construction*, vol. 17, no. 1, pp. 3 – 10, 2007.

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

[3] K. Alexis, G. Darivianakis, M. Burri, and R. Siegwart, "Aerial robotic contact-based inspection: planning and control," *Autonomous Robots*, pp. 1–25, 2015. [Online]. Available: http://dx.doi.org/10.1007/s10514-015-9485-5

[4] T. Merz and F. Kendoul, "Beyond visual range obstacle avoidance and infrastructure inspection by an autonomous helicopter," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, 2011, pp. 4953–4960.

[5] S. Rathinam, Z. Kim, and R. Sengupta, "Vision-based monitoring of locally linear structures using an unmanned aerial vehicle1," *Journal of Infrastructure Systems*, vol. 14, no. 1, pp. 52–63, 2008.

[6] G. Darivianakis, K. Alexis, M. Burri, and R. Siegwart, "Hybrid predictive control for aerial robotic physical interaction towards inspection operations," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, May 2014, pp. 53–58.

[7] A. Bircher, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel and R. Siegwart, "Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 6423–6430. [Online]. Available: https://github.com/ethz-asl/StructuralInspectionPlanner

[8] P. Oettershagen, T. Stastny, T. Mantel, A. Melzer, K. Rudin, G. Agamennoni, K. Alexis, and R. Siegwart, "Long-endurance sensing and mapping using a hand-launchable solar-powered uav," June 2015.

[9] Papachristos, Christos and Alexis, Kostas and Tzes, Anthony, "Technical activities execution with a tiltrotor uas employing explicit model predictive control," *International Federation of Automatic Control, Proceedings of the 19th IFAC World Congress*, 2014.

[10] P. Vansteenwegen, W. Souffriau, and D. Van Oudheusden, "The orienteering problem: A survey," *European Journal of Operational Research*, vol. 209, no. 1, pp. 1–10, 2011.

[11] T. Tsiligirides, "Heuristic methods applied to orienteering," *Journal of the Operational Research Society*, pp. 797–809, 1984.

[12] A. Blum, S. Chawla, D. R. Karger, T. Lane, A. Meyerson, and M. Minkoff, "Approximation algorithms for orienteering and discounted-reward tsp," *SIAM Journal on Computing*, vol. 37, no. 2, pp. 653–670, 2007.

[13] H. E. Stanley and S. V. Buldyrev, "Statistical physics: The salesman and the tourist," *Nature*, vol. 413, no. 6854, pp. 373–374, 2001.

[14] D. Gavalas, C. Konstantopoulos, K. Mastakas, G. Pantziou, and N. Vathis, "Heuristics for the time dependent team orienteering problem: Application to tourist route planning," *Computers & Operations Research*, vol. 62, pp. 36–50, 2015.

[15] J. Yu, J. Aslam, S. Karaman, and D. Rus, "Optimal tourist problem and anytime planning of trip itineraries," *arXiv preprint arXiv:1409.8536*, 2014.

[16] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," *Algorithmica*, vol. 20, no. 4, pp. 374–387, 1998. [Online]. Available: http://dx.doi.org/10.1007/PL00009201

[17] K. Alexis, C. Papachristos, "Distributed Infrastructure Structural Inspection Planner Dataset." [Online]. Available: https://disip.wikispaces.com/

[18] 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.

[19] 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.

[20] Keld Helsgaun, "LKH Version 2.07: an effective implementation of the Lin-Kernighan heuristic for solving the traveling salesman problem ." [Online]. Available: http://www.akira.ruc.dk/~keld/research/LKH/

[21] C. Papachristos, D. Tzoumanikas, and A. Tzes, "Aerial robotic tracking of a generalized mobile target employing visual and spatio—temporal dynamic subject perception," in *Intelligent Robots and Systems, 2015. IROS 2015. IEEE/RSJ International Conference on*, Sept–Oct 2015.

[22] Pix4D, "http://pix4d.com/."