

## Multi-UAV trajectory planning for 3D visual inspection of complex structures

Stefan Ivić<sup>a</sup>, Bojan Crnković<sup>b</sup>, Luka Grbčić<sup>a,d</sup>, Lea Matleković<sup>c,\*</sup>

<sup>a</sup> Faculty of Engineering, University of Rijeka, Vukovarska 58, Rijeka, 51000, Croatia

<sup>b</sup> Department of Mathematics, University of Rijeka, Slavka Krautzeka 75, Rijeka, 51000, Croatia

<sup>c</sup> Department of Mathematics and Computer Science, University of Southern Denmark, Campusvej 55, Odense, 5230, Denmark

<sup>d</sup> Center for Advanced Computing and Modelling, University of Rijeka, Radmila Matejcic 52, Rijeka, 51000, Croatia

### ARTICLE INFO

**Keywords:**

Autonomous UAVs

Trajectory planning

3D domain

CPP

Coverage

Inspection

### ABSTRACT

The application of autonomous UAVs to infrastructure inspection tasks provides benefits in terms of operation time reduction, safety, and cost-effectiveness. This paper presents trajectory planning for three-dimensional autonomous multi-UAV volume coverage and visual inspection of infrastructure based on the Heat Equation Driven Area Coverage (HEDAC) algorithm. The method generates trajectories using a potential field and implements distance fields to prevent collisions and to determine UAVs' camera orientation. It successfully achieves coverage during the visual inspection of complex structures such as a wind turbine and a bridge, outperforming a state-of-the-art method by allowing more surface area to be inspected under the same conditions. The presented trajectory planning method offers flexibility in various setup parameters and is applicable to real-world inspection tasks. Conclusively, the proposed methodology could potentially be applied to different autonomous UAV tasks, or even utilized as a UAV motion control method if its computational efficiency is improved.

### 1. Introduction

In recent years and with the constant demand for the automatization of processes, there has been a rising interest in research of autonomous Unmanned Aerial Vehicles (UAVs) and autonomous robotic systems in general. Known for their versatile movements and ability to explore the unstructured environment, UAVs are being used for numerous civil applications such as transportation [1,2], monitoring and surveying [3, 4], search and rescue [5,6] as well as infrastructure inspection [7]. With an increase in the UAV's autonomy, UAV applications supporting construction and infrastructure inspection are expected to dominate the total UAV market with a share of around 45% [8]. Manual UAV inspections have already enhanced safety as inspectors are not challenged with working at high altitudes and with heavy equipment anymore. However, inspection tasks still require skilled operating teams and can take a long time to coordinate and execute. By raising the level of autonomy, UAVs have high potential to reduce inspection time and human labor, therefore decreasing the total inspection costs. Deploying autonomous UAVs could further improve safety as with faster and cheaper inspections, the civil infrastructure could be better maintained resulting in the prevention of accidents.

Tackling the challenge of making UAVs autonomous requires an interdisciplinary approach, collaboration, and knowledge distribution

across the fields. The agents must be equipped with accurate sensor technologies that enable the control system to direct a vehicle according to the developed path or trajectory planning strategy. Path and trajectory planning problems are crucial to solve in order to fly autonomously. Solving them imposes additional complexities as the approach usually depends on a specific application. Path planning refers to finding a geometric, collision-free path that achieves the mission's objectives, while trajectory planning refers to assigning a time law to the geometric path, therefore, providing control inputs for desired movements. In most cases, path planning precedes trajectory planning; however, these two phases are not necessarily distinct [9] as in the case where trajectories are generated from the initial to the final position and both problems are solved at the same time. Although many studies have been published concerning path and trajectory planning for autonomous vehicles, there is a lack of research attention to multi-UAV cooperation for construction and infrastructure inspection applications [8]. Cooperation involves the distribution of inspection tasks and collaborative, collision-free path and trajectory planning. Cooperative planning could provide wider inspection scope, higher error tolerance, and faster completion time, but increases algorithmic complexity. Current research regarding inspection path and trajectory planning focuses on solving the Coverage Path Planning

\* Corresponding author.

E-mail addresses: [stefan.ivic@riteh.hr](mailto:stefan.ivic@riteh.hr) (S. Ivić), [bojan.crnkovic@uniri.hr](mailto:bojan.crnkovic@uniri.hr) (B. Crnković), [luka.grbcic@riteh.hr](mailto:luka.grbcic@riteh.hr) (L. Grbčić), [matlekovic@imada.sdu.dk](mailto:matlekovic@imada.sdu.dk) (L. Matleković).

(CPP) problem which aims at determining a geometric, collision-free path covering the area or volume of interest [10]. Afterwards, paths are usually endowed with simple point-to-point trajectories. Numerous existing publications tackle two-dimensional coverage problems, but there has been a growing need for solutions to three-dimensional coverage problems.

This paper aims to fill in the research gap within multi-UAV cooperation for infrastructure inspection by presenting a novel multi-UAV trajectory planning algorithm for three-dimensional coverage as well as applying it for the visual inspection of complex structures that usually require frequent maintenance. The algorithm takes into account vehicle parameters and environmental constraints producing somewhat smooth, collision-free trajectories. It is based on a potential field approach, designing the field using a modified Heat Equation Driven Area Coverage (HEDAC) algorithm [11] which minimizes the difference between the desired and achieved coverage, therefore leading agents through less covered space. The most important aspect when visually inspecting infrastructure employing autonomous UAV is assuring that all surfaces of the structure are included in the sensor's field of view at least once. Therefore, we upgrade the HEDAC trajectory generation with a camera direction control which always directs the camera to the closest point of the structure. As trajectories are produced in the potential field, we leave out the geometric path determination and solve both coverage path and trajectory planning problems at the same time. When accompanied by a robust trajectory-following control onboard a UAV, HEDAC can be applied to real-world infrastructure inspection. Due to the prolonged computational time, the algorithm is unable to run onboard in real time and serve as a motion controller.

This article is organized as follows. Section 2 presents an overview of the state-of-the-art research concerning three-dimensional coverage path and trajectory planning as well as the HEDAC algorithm foundation and applications to two-dimensional coverage problems. In Section 3, the adaptation of the HEDAC algorithm for three-dimensional domains is presented with the implementation based on the finite element method (FEM) and applied to a 3D unit cube ergodic coverage scenario. In Section 4, we adapt the algorithm for 3D visual inspection by introducing the distance field and the camera orientation. Furthermore, we provide a simplified surface inspection evaluation and present simulation results for the inspection of a portal, wind turbine, and bridge structures. A thorough analysis of the performance, limitations, and validation of the HEDAC approach has been conducted and presented in Section 5. Finally, we conclude in Section 6 by summarizing the achievements and limitations as well as proposing future improvements.

## 2. State-of-the-art

An autonomous infrastructure inspection requires a UAV to visually capture all elements of the structure of interest. To accomplish that, vehicles need to be equipped with an appropriate camera and traverse paths that allow the camera's field of view to cover the whole structure, ensuring desired visibility. Such paths are found by solving the CPP problem. Many algorithms have been proposed for covering two-dimensional areas for inspection and monitoring purposes, making it a well-researched topic [12]. In recent years, more and more attention is being dedicated to three-dimensional CPP. CPP approaches are categorized as either model-based or non-model-based [13]. The non-model-based approaches do not require a known reference model, therefore, are applied for the unknown environment exploration or 3D model reconstruction. The model-based approaches use a reference model to generate coverage paths, making them applicable for structural inspection [14].

This article focuses on producing feasible, three-dimensional coverage trajectories based on known models for structural inspections. As our approach solves the CPP problem, we present state-of-the-art achievements in the model-based 3D CPP for infrastructure inspection purposes. The literature review is divided according to the

number of agents employed for inspection. Furthermore, we provide an overview of the existing research regarding the HEDAC algorithm and its applications in two-dimensional domains.

### 2.1. Single-agent coverage path planning

In the attempt to solve the CPP problem for three-dimensional structure inspection, many proposed solutions generate viewpoints around a known 3D structure and employ different techniques to find coverage paths. Viewpoints are usually generated using sampling-based techniques or grid decomposition and the coverage path is found by solving the Traveling Salesman Problem (TSP) or the Vehicle Routing Problem (VRP). Multiple optimizations are often proposed for improving the overall inspection performance. A co-optimal CPP method that simultaneously optimizes the UAV path and quality of captured images and reduces the computational complexity of the solver, all while adhering to safety and inspection requirements is presented in [15]. The path optimization algorithm utilizes a Particle Swarm Optimization (PSO) framework which iteratively optimizes the coverage paths based on sampled viewpoints. The core of the method consists of a cost function that measures the quality and efficiency of a coverage inspection path and the greedy heuristic for optimization enhancement by aggressively exploring the search space. Flight trajectories are generated using global B-spline curve interpolation and the method is successfully applied for the inspection of a building, a statue, and a solar plant. Another approach where a variant of PSO is used for solving TSP is presented in [16]. Inspection viewpoints are generated based on a grid decomposition and performance is improved by using deterministic initialization, random mutation, and edge exchange. The focus of the article is on reducing the computation time by taking advantage of parallel computing on a GPU-based framework. Flight trajectories are not proposed, however, based on the visuals given for room and bridge inspection applications, a point-to-point controller could be employed to traverse determined paths. An alternative approach proposed in [17] solves TSP for sampled viewpoints by using Google OR-Tools at two different levels to provide a path for aircraft and bridge inspections. Firstly, a high-level algorithm separates the environment into multiple subspaces at different resolutions and solves the global TSP to determine the order of visiting subspaces. Secondly, a low-level TSP solution finds paths within the subspaces for detailed coverage and resolves collisions by adding new viewpoints. A comparison with greedy TSP solver is presented in [18] where a traversal algorithm is developed. Authors uniformly sample a point cloud model and create a bounding box around it. Viewpoints are generated by projecting sampled points onto the planes of the bounding box and a traversal path search algorithm is proposed to find paths. Paths are optimized in terms of flight distance and smoothness while considering safety constraints and sensor limitations. The proposed method is evaluated for use cases of bridge and power pylon inspection where it outperformed greedy TSP by finding simpler paths in less time. A similar approach is presented in [19]. The authors divide a structure's volumetric map into several layers, and in each layer, a set of normal vectors of each voxel's center point is calculated. The opposing vectors are used as viewpoints and the TSP is solved using the Lin-Kernighan Heuristic (LKH) solver. Afterward, all paths in each layer are combined to form a complete path. To quantify the inspection performance, the authors propose a coverage measure as a percentage of voxels included in the camera's field of view, achieving the result of 99.8%. Most of the other authors fail to provide success metrics for the determined coverage paths.

Some single-agent CPP approaches do not have a clear boundary between viewpoint sampling and path determination. Instead, these two steps are interchangeably applied to optimize both coverage and paths. A practical application of a two-step optimization approach has been demonstrated in [20] by using a trolley mesh model. The algorithm initially samples viewpoints based on the triangles in the mesh model and solves TSP to obtain an initial tour. Afterward, viewpoints

are resampled such that final paths are shorter and full coverage is provided. Full coverage is proven by reconstructing a 3D model of the structure of interest. Authors later extend their work [21] by planning the coverage path for the wind turbine, staircase, mountain, and cone models. As a part of their work on an online exploration path planner, the same authors developed an inspection CPP algorithm based on the receding horizon approach [22]. When employed for the inspection, the algorithm samples the next best view in a random geometric tree intending to include the structure in the camera's field of view. Feasibility is demonstrated in a bridge inspection case and practical applicability in a real-life scenario of a simple object inspection. In Section 5, we compare our approach for the bridge inspection with the receding horizon planner.

The main limitation of the described existing approaches for three-dimensional coverage path planning is the number of agents deployed for the inspection. They have considered only a single-agent inspection with a focus on the computational time to compute paths. Taking into account that these model-based planners compute paths offline, decreasing the computation time would not highly affect the inspection time. Actual inspection time seems to have more impact in achieving faster and more frequent infrastructure inspections. We intend to focus on collaborative multi-agent inspection path planning as a higher number of agents is expected to decrease the inspection time. With our approach, it is still possible to plan a single-agent trajectory in case the structure is simple enough and there is no need to reduce the inspection time.

## 2.2. Multi-agent coverage path planning

The problem of coverage path planning is enhanced by involving multiple agents which are expected to accelerate the mission accomplishment. Many known problems such as TSP or VRP are being modified and adapted for multi-agent coverage problems [13], but there is still a lack of solutions for specific applications such as 3D infrastructure inspection. Some existing solutions for multi-agent 3D area coverage such as [23], which divides the area in hexagons and produces a lawnmower pattern to cover it, cannot be adapted for structural inspection as the focus is on covering only the top part of the 3D map. These solutions do not consider complex structures which have to be covered from all sides. A similar approach of dividing a three-dimensional model in order to determine coverage paths for multiple UAVs is presented in [24]. Flight waypoints are generated based on the horizontally sliced 3D model and converted into position-velocity-yaw trajectories. Although the approach is applied for the inspection of a wind turbine and an outdoor structure, it would not be directly applicable for more complex structures as the slicing approach would not produce appropriate paths. An example of the infrastructure inspection solution which follows the pattern of most of the previously presented single-agent approaches is found in [25]. Authors use incremental sampling to create a roadmap with information on topology, coverage, and path lengths. The multi-agent CPP is formulated as a combination of two NP-hard problems: the Set Covering Problem and the VRP problem. The problem is solved by a modified Biased Random Key Genetic Algorithm which optimizes the inspection paths for multiple agents. The issue of this and similar approaches solving a variant of a TSP is that trajectories endowed through calculated waypoints are not continuous, but rather interrupted with sharp turns. Such solutions would require UAVs to fly from point to point, stopping and turning each time a path segment is accomplished, which would result in prolonged inspection time. Alternatively, if one attempts to generate trajectories through the determined viewpoints, there is a risk that no feasible solutions for a selected vehicle model would be found. The complexity of the solution drastically increases when constraints for obstacle avoidance are introduced. Our solution addresses both CPP and trajectory planning problems at the same time by producing continuous and feasible trajectories based on a potential field approach instead of solving a multi-TSP. Furthermore, the approach is highly adaptable to different, complex structures, which we demonstrate in the upcoming sections.

## 2.3. HEDAC multi-agent 2D trajectory planning

The ergodic multi-agent area coverage method HEDAC is presented for the first time in [11]. The algorithm designs a potential field based on a steady-state heat equation using a source term that depends on the difference between the given goal density and the current coverage density. The agents' movements are directed by the gradient of that potential field, leading them to the area of interest and producing paths to achieve the goal coverage density while avoiding collisions between agents using a built-in local cooling mechanism. The algorithm is demonstrated for two-dimensional test cases and compared with the Spectral Multiscale Coverage (SMC) algorithm showing superiority in achieved coverage and computational time. Simply stated, the algorithm leads agents to attractive areas that become less attractive as the area is covered, making the approach applicable for monitoring, surveillance, and inspection purposes.

The HEDAC algorithm application is further demonstrated for autonomous non-uniform multi-agent spraying tasks and presented in [26]. The algorithm combines the spraying and Dubins motion models to produce the path covering the area with the desired spraying density. The method is tested in comparison to the Lawnmower and SMC algorithms for simple geometries, as well as a realistic crop spraying case, outperforming both methods in convergence time while producing spraying density of satisfying accuracy and using less spraying media when compared with conventional spraying.

The adaptation of the HEDAC algorithm for multi-agent area search in uncertain conditions is presented in [27]. The undetected target probability-density field is computed from the initial probability field, already achieved agents' trajectories, diverse motion, and sensing parameters of each agent. The HEDAC method directs agents toward regions with a higher concentration of undetected targets, resulting in the maximization of the target detection rate. The target search using HEDAC was simulated and compared with Lawnmower, SMC, and Receding Horizon Control (RHC) approaches, demonstrating a shorter search time to achieve the same target detection rate.

Improvement of the HEDAC algorithm for multi-agent surveying constrained motion control in irregular domains is proposed in [28]. The previous implementation has considered rectangular domains due to using the finite difference method for solving a partial differential equation regulating the potential. The improved solution uses the finite element method to enable a simple and elegant application of boundary conditions and modeling of static obstacles in the domain. Optimization for collision avoidance maneuvers and the ability to set path curvature constraints improved the HEDAC algorithm's suitability for real-world 2D area surveying applications. The algorithm is tested in a synthetic scenario, as well as two realistic scenarios, providing results with high computational efficiency that enables real-time execution.

Due to the excellent coverage properties, the HEDAC approach for two-dimensional domains has been thoroughly researched and applied to different problems. The multi-agent coverage control presented in [29] utilizes HEDAC independently on the multiple regions of a domain partitioned using Voronoi tessellation. By achieving successful coverage, this work indicates the possibility of decentralized use of HEDAC control. An experimental utilization of HEDAC control, using a 7-axis Franka Emika robot, is employed in [30] for the ergodic-based portrait drawing. Unlike in other HEDAC implementations, a non-stationary heat equation is used to produce the trajectories tracked by the robot-driven ink pen.

Within this article, we present the extension of the HEDAC and its application to a three-dimensional coverage problem, specifically infrastructure inspection.

### 3. Three-dimensional ergodic coverage

A multidimensional ergodic coverage with the HEDAC method has already been theoretically presented in [11], but all published improvements have been implemented and applied only to two-dimensional problems. In this section, we describe the trajectory planning approach using HEDAC and its implementation for a swarm of  $n$  UAVs operating in a three-dimensional domain  $\Omega \in \mathbb{R}^3$  encompassed with the boundary  $\Gamma$ . The algorithm designs a potential field  $\psi$  that attracts agents into the uncovered space until the target coverage density  $\mu_0$  is achieved. The resulting motion is described with collision-free trajectories  $\mathbf{y}_i : t \rightarrow \Omega$ , where  $t$  is time and  $i = 1, \dots, n$  are indices of UAVs. Finally, we demonstrate the approach in the crowded domain using 100 agents to successfully cover half of a unit cube while avoiding collisions.

#### 3.1. Space coverage

Three-dimensional coverage can be considered as a continuous action of a UAV along its trajectory. Mathematically, this can be defined as a convolution of instantaneous action  $\phi$  and the trajectory, which results in a field occupying the space around the path of a UAV. For practical reasons, we use a radial basis function, precisely the three-dimensional Gaussian function, as the instantaneous action. The smoothness of the Gaussian function allows better stability and smooth motion because the coverage is accumulated near the trajectory. The use of the Gaussian function in the HEDAC control has been proven in numerical experiments in two-dimensional coverage motion control applications [11,27,28]. This instantaneous action is defined as:

$$\phi_\sigma(r) = \frac{\Phi}{(\sigma\sqrt{2\pi})^3} \exp\left(-\frac{r^2}{2\sigma^2}\right) \quad (1)$$

where  $\Phi$  is the coverage action intensity,  $\sigma$  is the standard deviation or the scope of action function  $\phi$ , and  $r_i(t) = \|\mathbf{x} - \mathbf{y}_i(t)\|$  is the distance from the location in the domain  $\mathbf{x}$  to the agent's position  $\mathbf{y}$ . The intensity can be interpreted via a property of the instantaneous action function:  $\int_{\mathbb{R}^3} \phi d\mathbf{x} = \Phi$ . The coverage field  $\rho(\mathbf{x})$  is defined as a convolution integral:

$$\rho(\mathbf{x}, t) = \sum_{i=1}^n \int_0^t \phi_\sigma(r_i(\tau)) d\tau. \quad (2)$$

Note that, due to the simplicity and practicality, the same action function  $\phi_\sigma$  is utilized for all agents. Though, it has been shown in [27] that HEDAC can be utilized for governing multiple heterogeneous agents in both sensing (action) and motion characteristics.

The objective of the space coverage algorithm is to produce agents' trajectories that explore the domain according to a given density. Furthermore, we want to enable a continuous non-stopping coverage motion whenever the agents' safety constraints are not violated. According to the given target density  $\mu_0(\mathbf{x})$ , characterizing UAV visiting areas and frequency, one can define an exponential law to determine the remaining density  $\mu$  at the time  $t$  according to the achieved coverage  $\rho$ :

$$\mu(\mathbf{x}, t) = \mu_0(\mathbf{x}) \cdot \exp(-\rho(\mathbf{x}, t)). \quad (3)$$

The target density  $\mu_0$  is normalized to satisfy  $\int_{\Omega} \mu_0(\mathbf{x}) d\mathbf{x} = 1$ . This formulation is analogously used in [27] for defining the (undetected) target probability in multi-agent search motion control with uncertain target detection.

The objective of the ergodic exploration is to evenly fill the space with trajectories or to fill it according to a given density. In the proposed formulation (3), minimizing  $\mu$  does not lead to equalization of the target density  $\mu_0$  and coverage  $\rho$ . However, it is trivial to show that minimizing  $\mu$  results with  $\rho(\mathbf{x}) = \ln(\mu_0(\mathbf{x}))$ , i.e.  $\ln(\mu_0)$  can be considered the goal density used in [11,31]. This approach elegantly solves the problems of normalization and logarithm of zero-valued  $\mu_0$ , which can

arise in the conventional ergodic coverage formulation. As stated, in order to realize spatial coverage, the motion control needs to direct UAVs to accomplish:

$$\lim_{t \rightarrow \infty} \int_{\Omega} \mu(\mathbf{x}, t) d\mathbf{x} = 0. \quad (4)$$

It is suitable to define the measure of spatial density coverage:

$$\eta_V(t) = 1 - \int_{\Omega} \mu(\mathbf{x}, t) d\mathbf{x}, \quad (5)$$

which indicates a share of the covered space.

#### 3.2. Utilizing a potential field for directing UAVs

In order to minimize (4) in time, one needs to minimize the  $\mu$  spatially. HEDAC's main idea is to design a potential field that can facilitate the minimization of  $\mu$ . The Helmholtz partial differential equation, used for modeling the conductive heat transfer accompanied by convective cooling, is employed for obtaining the potential  $\psi$ :

$$k \cdot \Delta\psi(\mathbf{x}, t) - \psi(\mathbf{x}, t) + \mu(\mathbf{x}, t) = 0 \quad (6)$$

with Neumann boundary condition applied to the entire boundary:

$$\frac{\partial\psi}{\partial\mathbf{n}} \Big|_{\Gamma} = 0, \quad (7)$$

where  $\mathbf{n}$  is the boundary outward-pointing normal. The parameter  $k$  represents the coefficient of thermal conductivity and regulates global and local details of the resulting potential field. For the purpose of this search model, we assume this coefficient is dimensionless. Note that more parameters have been used in the previous HEDAC formulations, but the conduction coefficient is dominantly regulating the behavior of  $\psi$  hence it is reasonable to consider only this parameter of the HEDAC control.

The potential field  $\psi$  calculated by (6) is actually a smoothed field  $\mu$ , which is accomplished due to using the Laplacian operator  $\Delta\psi$ . This allows us to utilize the gradient of the potential in order to establish the direction to regions of higher potential and implicitly to regions of higher values of density  $\mu$ . We calculate the desired direction of each UAV motion as a unit gradient of the potential  $\psi$ :

$$\mathbf{u}_i(t) = \frac{\nabla\psi(\mathbf{y}_i, t)}{\|\nabla\psi(\mathbf{y}_i, t)\|}, \quad i = 1, \dots, n. \quad (8)$$

Finally, we define a motion model to close the control's feedback loop. The motion, and consequentially the trajectories, are an outcome of the 1st order control  $d\mathbf{y}_i/dt = v_i \cdot \mathbf{u}_i(t)$ , where the position is directly changed by the control-appointed direction. The complete motion model, including the collision avoidance mechanism, is defined in Section 3.4. This motion model is often called kinematic as it neglects the mass and inertial effects of the UAV motion. However, due to advanced low-level multi-rotor control, in practice, modern multi-rotors can make a turn in any direction almost instantaneously. Therefore, it is justified to use such a model for multi-rotor UAV inspection simulations.

Due to the Neumann boundary condition, the gradient of the potential  $\nabla\psi$  inherently prevents an agent from approaching the domain boundaries. However, there is no assurance that the agent will not collide with the boundary due to a chosen combination of numerical parameters for solving the partial differential Eq. (6) (such as numerical grid density), UAV motion time step, and UAV properties (such as velocity  $v$  and action range  $\sigma$ ). Furthermore, a possible collision between two or more UAVs also needs to be taken into account when directing multiple UAVs.

### 3.3. Collision avoidance

Several different approaches for boundary and inter-UAV collision avoidance are successfully used in two-dimensional HEDAC applications [11,26,28]. Compared to collision avoidance in two dimensions, the three-dimensional case has more degrees of freedom hence the probability of random collisions between UAVs is greatly reduced. However, they do eventually occur if no collision avoidance procedure is implemented. We have implemented a robust and computationally inexpensive collision avoidance mechanism that prevents collisions of agents with other agents and with the domain boundaries.

The possibility of collisions is checked at every time step for  $i$ th agent by measuring the distance  $d(i, j) = \|\mathbf{y}_j - \mathbf{y}_i\|$ ,  $i \neq j$  to all other agents indexed with  $j$ , where  $d(i, i) = \infty$ , and the minimal distance from the agent to the domain boundary  $d_b(\mathbf{y}_i)$ . If the minimum of all measured distances  $d(\mathbf{y}_i) = \min(d_b(\mathbf{y}_i), d(i, 1), \dots, d(i, n))$  is below initially prescribed threshold  $2\epsilon$ , the collision avoidance must intervene and correct the agent's direction vector. To ensure that the outlines of a UAV do not collide with obstacles, the safety distance must be larger than the maximum dimension from the center of the UAV  $M$  plus the distance that the UAV can reach in a time step. Safety distance parameter  $\epsilon > v_i \Delta t + M$  must be large enough to enable safe and continuous maneuvers when avoiding collisions and low enough to not interfere significantly with the primary goal. If the agent's distance to obstacles is below the safety distance, it must not be allowed to reduce this distance in the next step.

Let  $I_i = \{j | d(i, j) < 2\epsilon, i \neq j\}$  be a set of indices of agents for which the  $i$ th agent is in danger of collision and suppose that  $d_b(\mathbf{y}_i) < 2\epsilon$  i.e. the  $i$ th agent is also to close to the domain boundary. We would like to find a direction vector  $\mathbf{w}_i$  for the  $i$ th agent such that it closes an obtuse angle with vectors  $\mathbf{e}_{i,j} = \mathbf{y}_j - \mathbf{y}_i, j \in I_i$  and vector  $\nabla d_b(\mathbf{y}_i)$ . Furthermore, to make this vector unique, we can pick the one which is closest to the vector  $\mathbf{c} = \sum_{j \in I_i} \mathbf{e}_{i,j} / \|\mathbf{e}_{i,j}\| + \nabla d_b(\mathbf{y}_i) / \|\nabla d_b(\mathbf{y}_i)\| + \mathbf{u}_i$ .

If we are able to solve this problem, then this direction will lead UAVs away from the boundary and neighboring agents in the next step. This problem can be formulated as an optimization problem:

$$\text{maximize}_{\mathbf{w}_i} \quad \mathbf{c}^T \mathbf{w}_i \quad (9)$$

$$\text{subject to} \quad -\mathbf{A}\mathbf{w}_i \leq \mathbf{0} \quad (10)$$

$$\|\mathbf{w}_i\| = 1, \quad (11)$$

where

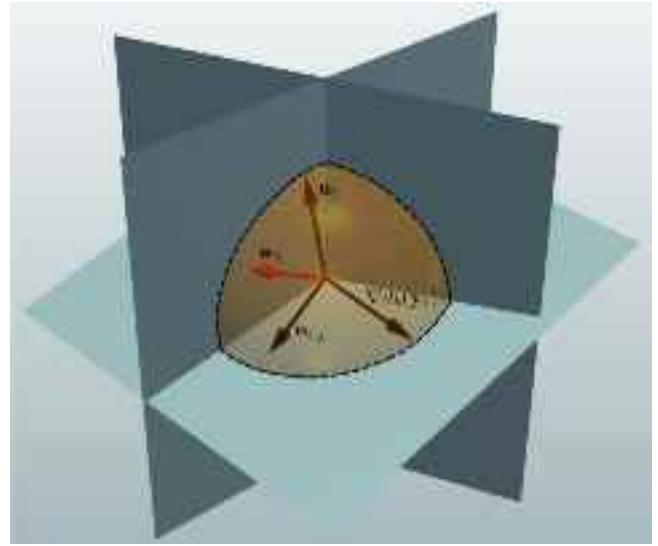
$$\mathbf{A} = \begin{bmatrix} \bar{\mathbf{A}} \\ \nabla d_b(\mathbf{y}_i) \\ \mathbf{u}_i \end{bmatrix}$$

$$\text{and } \bar{\mathbf{A}} = [\mathbf{e}_{i,j}], j \in I_i.$$

The solution of this linear objective function with nonlinear constraints can be solved efficiently using a nonlinear solver. Fig. 1 shows the feasible space for a matrix  $A$  with three rows where each row corresponds to a depicted black vector pointing towards the feasible region. The solution is a unit vector  $\mathbf{w}_i$  depicted in red which closes an acute angle with highlighted vectors.

It can be shown that the distance from the boundary  $d_b \in C^2$ , therefore,  $\nabla d_b(\mathbf{y}_i)$  always exists and points towards the boundary. We should underline that the matrix  $A$  has at least one row and  $\nabla d_b(\mathbf{y}_i)$  is not included if  $d_b(\mathbf{y}_i) \geq \epsilon$ . It is possible that the optimization problem does not have a solution because the feasible set is empty. In that case, the matrix  $A$  must have at least 6 rows (conditions) which is highly unlikely in practical situations. In the two-dimensional case, it would take at least 4 rows for this to occur. In the unlikely case that the optimization problem is not solved, the UAV should stand still for this time step and wait for the neighboring agents to clear its path. This maneuver is feasible for a multi-rotor UAV.

An initial approximate solution to the optimization problem is obtained using an approximate linear program that finds a good initial solution candidate in the feasible space. The initial approximate



**Fig. 1.** Feasible region for the optimal collision avoidance vector marked with a dashed sphere triangle. Solution  $\mathbf{w}_i$  of the optimization problem with nonlinear constraint  $\|\mathbf{w}_i\| = 1$  and linear constraints defined by normal vectors  $\mathbf{u}_i$ ,  $\mathbf{e}_{i,j}$  and  $\nabla d_b(\mathbf{y}_i)$ .

solution is used by a Trust-region optimization implemented in Scipy 1.7.3 [32] which is based on algorithms found in [33]. The solution to this optimization problem is computationally inexpensive and is invoked sparsely therefore it does not add significantly to the overall complexity of the algorithm.

### 3.4. Defining UAVs' trajectories

The motion of each agent in the fleet of UAVs is defined as:

$$\frac{d\mathbf{y}_i}{dt} = \begin{cases} v_i \cdot \mathbf{w}_i(t) & \text{if } d(\mathbf{y}_i) < \epsilon, \\ v_i \cdot \left( \left( 2 - \frac{d(\mathbf{y}_i)}{\epsilon} \right) \mathbf{w}_i(t) + \left( \frac{d(\mathbf{y}_i)}{\epsilon} - 1 \right) \mathbf{u}_i(t) \right) & \text{if } \epsilon \leq d(\mathbf{y}_i) < 2\epsilon, \\ v_i \cdot \mathbf{u}_i(t) & \text{otherwise,} \end{cases} \quad (12)$$

where  $v$  is a velocity magnitude equal for all UAVs, and  $\mathbf{w}_i$  is the direction obtained as a solution of the nonlinear optimization problem.

The collision avoidance can cause jitters in the trajectories because it is not applied smoothly. To fix this problem and smooth out trajectories, the correction of the direction vector is applied gradually when the minimal distance satisfies  $d(\mathbf{y}_i) \in [\epsilon, 2\epsilon]$ . In the unlikely case that the agent gets too close to an obstacle, the collision-avoidance direction vector  $\mathbf{w}_i$  is applied to move the agent toward a collision-safe part of the domain.

### 3.5. Numerical implementation using finite element method

The numerical solution of Eq. (6) together with boundary condition (7) can be complicated to solve, especially if the geometry of the boundary is complex. In practical applications of this model, we use a connected 3D domain with holes that can have complex geometry. Furthermore, for practical applications, we need a relatively large number of numerical grid points. FEM provides a simple implementation of boundary conditions, a fast solver, and a straightforward interpolation of results on any given grid.

The weak formulation of the presented problem is obtained by multiplying Eq. (6) by a smooth test function  $v \in H^1(\Omega)$ ,  $\Omega \subset \mathbb{R}^3$  and integrating over the domain. Using integration by parts, the following equation is obtained:

$$-k \int_{\Omega} \nabla \psi(\mathbf{x}, t) \nabla v(\mathbf{x}) d\Omega - \int_{\Omega} \psi(\mathbf{x}, t) v(\mathbf{x}) d\Omega$$

$$+ \int_{\Omega} \mu(\mathbf{x}, t)v(\mathbf{x})d\Omega + \sum_{j=0}^{n_0} \oint_{\Gamma_j} (\nabla \psi(\mathbf{x}, t) \cdot \mathbf{n})v(\mathbf{x})d\Gamma = 0.$$

The final form of the weak formulation is obtained after we apply the Neumann boundary for all parts of the boundary  $\Gamma_j \subset \Gamma$ :

$$\int_{\Omega} k \nabla \psi(\mathbf{x}, t) \nabla v(\mathbf{x}) + \psi(\mathbf{x}, t)v(\mathbf{x}) - \mu(\mathbf{x}, t)v(\mathbf{x})d\Omega = 0. \quad (13)$$

We use quadratic polynomials for the space of test functions and our representation of the solution. Let  $M = \{T_1, \dots, T_N\}$  be a partition of  $\Omega$  into  $N$  uniform non-overlapping triangles. The triangles with the geometry are described by the classical 3-node interpolation functions. The scalar field of the unknown variable and test functions over each  $n$ -node element  $T_i$  is approximated by

$$\psi_h = \sum_{j=1}^n N_j u_j$$

where  $N_j$  stands for the Lagrangian polynomial interpolation functions.

Because the domain and the triangulation do not change during the entire calculation, the linear system coefficient matrix is sparse and constant hence the solutions are obtained very efficiently. The interpolation of different scalar fields and calculation of the gradient  $\nabla \psi$  can be directly obtained using a finite element representation [34]. All results are obtained with the latest version of the finite element software NGsolve.

### 3.6. 3D coverage example

The proposed algorithm can be used to cover a three-dimensional space with a given density of trajectories. To demonstrate this application we use a simple unit cube domain and 100 agents initially distributed randomly in the center of the cube. The density of the trajectories must uniformly fill the lower part of the cube where  $0.05 < x, y < 0.95$  and  $0.05 < z < 0.45$ . This test demonstrates that this algorithm can avoid collisions inside a crowded domain without sacrificing the goal of space coverage. The safety distance is set to 0.025 m and the velocity is a constant of 0.1 m/s. Total simulation time is 2000 s with the time step  $\Delta t = 1$  s and the conduction parameter  $k = 0.1$ .

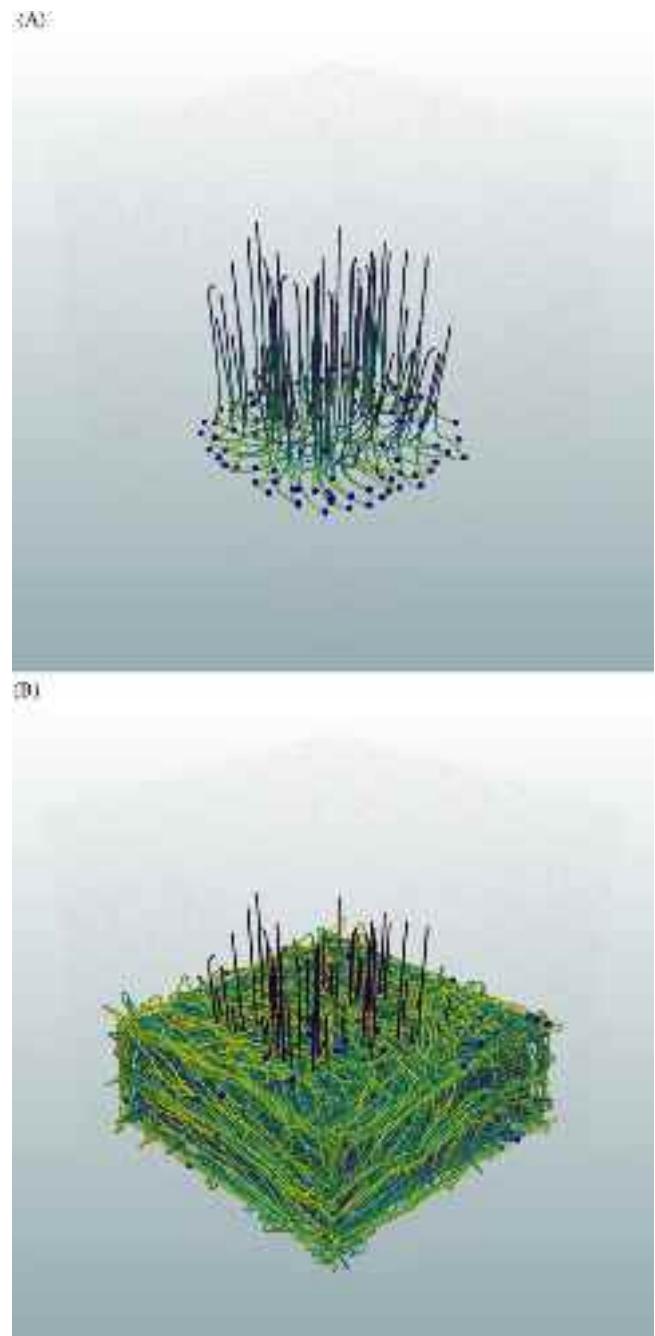
**Fig. 2** shows the final result after 50 s and 1000 s. One can observe that the trajectories form a space-filling curve and the agents keep inside the goal area except for a short initial part of the simulation. This domain is crowded with agents but the trajectories are quite smooth and no deadlocks or jitters in the produced trajectories can be observed.

**Fig. 3** clearly shows that the given domain is explored efficiently and the collision avoidance is working properly in this crowded domain. The agents safely navigate the domain and avoid other agents in the process. We must emphasize that the agents, in this case, are more in danger of colliding with each other than the domain boundary. This is a simple artificial test example, but the results demonstrate the algorithm is ready to be tested in a more realistic setting with complex geometries.

## 4. UAV visual inspection applications

The main task in the visual inspection is to capture all surfaces of the structure using a camera mounted on the UAV. This requires control of the direction of the camera depending on the position of the drone relative to the observed structure. We propose a rather simple idea that considers the control of camera orientation, along with obtained UAV trajectory, by directing the camera view to the nearest point of the observed structure.

In order to demonstrate the applicability of the proposed algorithm to infrastructure inspection, we conduct inspection simulations for three different test cases with varying complexity.



**Fig. 2.** Trajectories of 100 agents after 50 (A) and 1000 s (B) produced with HEDAC method for uniform coverage of lower half of a unit cube.

### 4.1. Area of interest and camera control for 3D structure inspection

We propose a camera direction control that can provide relatively good results if trajectories obtained by HEDAC suitably explore the region around the structure. Since the inspection is acting on faces and surfaces of the investigated structure, the scope of required trajectories cannot be correctly transferred to the three-dimensional domain  $\Omega$ . However, the proposed approximation is satisfactory as it is shown in the UAV inspection scenarios presented in this section and further investigated in Section 5.

Similar to the  $d_b$  field, we use  $d_s(\mathbf{x})$  as the distance of  $\mathbf{x}$  to the structure surface. Now, we can define a suitable region of interest around

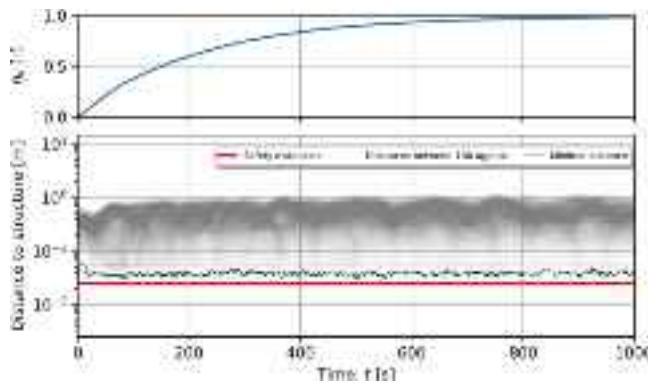


Fig. 3. Convergence of the spatial density coverage measure  $\eta_V$  and computed UAV distances during the inspection of a simple unit cube with 100 agents.

the observed structure using a three-dimensional Gaussian function:

$$\mu_0(\mathbf{x}) = \exp\left(-\frac{(d_s(\mathbf{x}) - d_m)^2}{2 \cdot d_\sigma^2}\right), \quad (14)$$

where  $d_m$  is the goal inspection distance at which the Gaussian function is centered and  $d_\sigma$  is the standard deviation, i.e. the broadness of the field  $\mu_0$  encompassing the structure. Using (14) allows us to construct a continuous non-negative field around the entire structure, with a peak exactly at the distance  $d_m$  from the surfaces to be inspected and gradually weakening both towards and away from the structure. Note that scaling constants are omitted from the Gaussian function in (14) since  $\mu_0$  is subject to normalization that produces compliant scaling constants.

Finally, we can easily find a camera orientation  $\mathbf{z}_i$  for each UAV, as the direction towards the nearest point on the structure, by employing the gradient of the field  $d_s$ :

$$\mathbf{z}_i(\mathbf{y}_i) = \frac{\nabla d_s(\mathbf{y}_i)}{\|\nabla d_s(\mathbf{y}_i)\|}. \quad (15)$$

Note that only camera orientation is subjected to the proposed control and provided by the unit vector  $\mathbf{z}$ . This approach does not acknowledge the field of view (FOV), camera focus, zoom, or other details regarding photographic equipment for visual inspection. Although not considered in the trajectory planning algorithm, a simplified field of view has been utilized for surface coverage assessment as described in the next subsection.

#### 4.2. Surface coverage assessment

In order to validate the correctness of solving the surface coverage task via ergodic exploration of the spatial field, we implement a simulation of the inspection camera view (Fig. 4). The camera's FOV is modeled as a cone whose top is positioned at the UAV's center  $\mathbf{y}_i$ . The cone orientation corresponds to camera orientation  $\mathbf{z}_i$  obtained by (15). The height of the cone  $C_H$  regulates the maximum acceptable distance at which inspection images are taken while  $C_R$  is the radius of the FOV at that distance. We only consider a circular area of the image, in contrast to the rectangular image obtainable by the camera. This can be justified by considering only inspection details at the center of the recorded image.

At each time step, all nodes on the structure's surface are tested if they are inside the cone. The candidate nodes (the ones inside the cone) are further tested to check if they are in the camera's line of sight. This is performed by a simple ray tracing technique: if there is no intersection between a ray shot from the camera and the structure's surface, then a candidate node is directly observed. The number of observations is tracked for all nodes during the entire inspection simulation. For

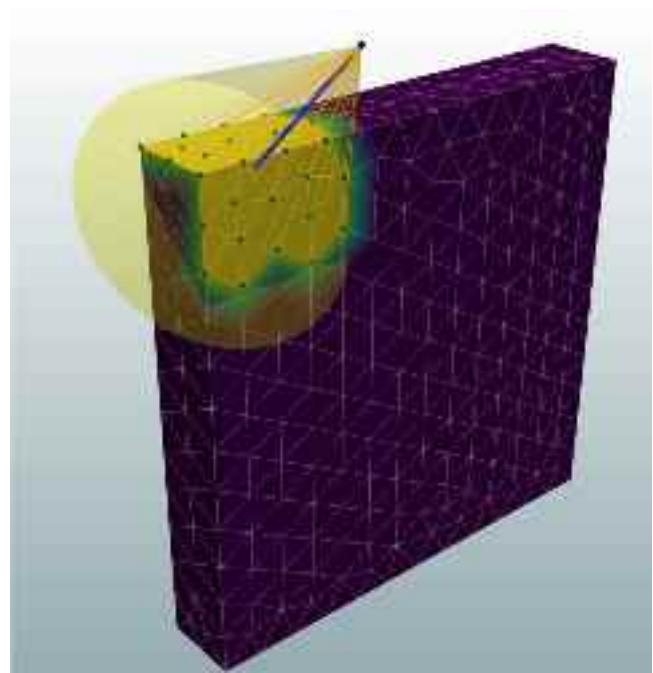


Fig. 4. Example of surface inspection evaluation via calculation of observable nodes (marked as green dots) in a conical field of view (semi-transparent yellow cone). Note that not all candidate nodes (nodes inside of the cone) are marked as observed. Nodes marked as red dots did not pass a "direct line of sight" test, i.e. their orange ray intersects the surface before landing into the candidate node. The blue dot represents the UAV/camera location while the thick blue line represents the camera orientation z i.e. the cone axis.

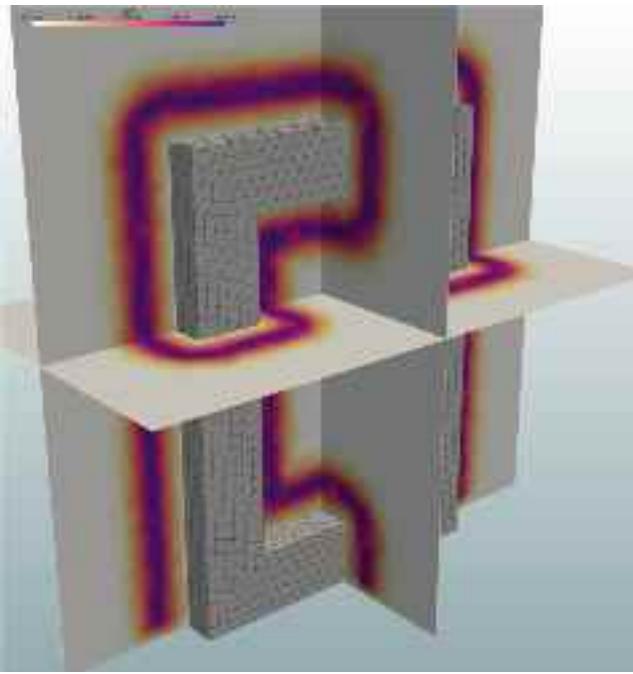
relatively uniform meshes, which are used in the presented test cases, the ratio between the number of inspected nodes and the total number of nodes on the structure's surface represents a share of the inspected surface  $\eta_A$ . Although simplified to ensure computational feasibility, this coverage assessment model provides a qualitatively satisfactory estimate of surface inspection.

#### 4.3. Portal test case

The portal test case is a scenario for visual inspection of a relatively simple synthetic three-dimensional structure. It can be described as a flattened box ( $50 \times 70 \times 10$  m) with a rectangular hole in it ( $30 \times 50$  m), and it is designed to provide a simple shape while requiring relatively complex maneuvers to achieve structure inspection. The target density field  $\mu_0$  (Fig. 5) is defined using the goal inspection distance  $d_m = 5$  m and the broadness  $d_\sigma = 2$  m. Other info and parameters used for the portal case simulation are presented in Table 1.

The proposed trajectory planning algorithm applied for the visual inspection of the portal structure is run with 3 UAVs in the duration of 1500 s. For the given  $\mu_0$ , the algorithm achieves convergence of the spatial density and the surface inspection coverage as shown in  $\eta_V$  and  $\eta_A$  plots in Fig. 6(A).

Plotted distance lines indicate that generated UAV paths around the structure are, on average, at distance  $d_m = 5$  m from the structure. However, it can be observed that distances slightly increase, on average, during the inspection. The target density  $\mu_0$  is computed to form a region at the offset from the structure's surface which is mostly convex. The resulting potential field  $\psi$  is positioned slightly closer to the structure than  $\mu_0$  on the convex part of the structure's surface, due to the smoothing effect of the Laplacian operator in the Helmholtz equation. Consequently, the outcome is UAV trajectories that are closer to the structure than appointed  $d_m$ . After initial rough passes around the structure, which are relatively close to it, the remaining density



**Fig. 5.** Target density field  $\mu_0$ , of thickness  $d_\sigma = 2$  m, encompasses the portal structure at the distance  $d_m = 5$  m.

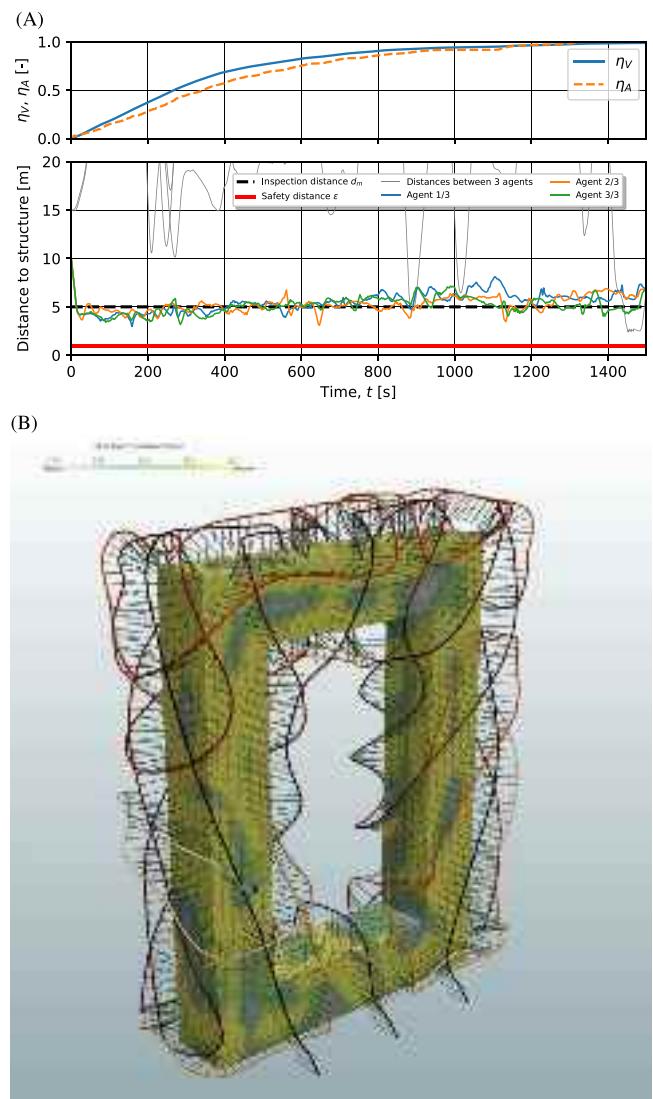
**Table 1**

Domain, structure and numerical mesh info, inspection task parameters, UAV parameters, and HEDAC parameters used in the portal test case.

Parameter	Value	Unit
Structure length	10	m
Structure width	50	m
Structure height	70	m
Domain length	50	m
Domain width	90	m
Domain height	90	m
Number of domain mesh nodes	34 161	
Number of domain mesh elements	167 495	
Number of structure mesh surface nodes	2315	
Number of structure mesh surface faces	4630	
Inspection distance $d_m$	5	m
Inspection distance broadness $d_\sigma$	2	m
FOV cone height $C_H$	8	m
FOV cone diameter $C_D$	10	m
Number of UAVs	3	
UAV velocity $v$	0.5	m/s
Safety distance $\epsilon$	1	m
Coverage action intensity $\Phi$	200	
Coverage action range $\sigma$	5	m
HEDAC conduction coefficient $k$	12	
Inspection duration	1500	s
Path planning time step $\Delta t$	1	s

$\mu$  is now more accumulated at a distance greater than  $d_m$  from the inspected surfaces. Thus, in the second part of the inspection operation, UAV trajectories are generated at a greater distance from the structure. The distances of the produced trajectories are safely kept above the minimal safety distance  $\epsilon$  during the entire operation. Due to the large proportions of structure and domain, distances between UAVs are practically negligible in the context of the minimum allowable spacing constraint.

The produced trajectories  $y_i$  and accompanying camera orientations  $z_i$  for all three UAVs are displayed in Fig. 6(B). Note that the unit vector of camera orientation is scaled to the length equal to the inspection distance  $d_m$ . The produced trajectories are fairly smooth and suitably



**Fig. 6.** (A) Convergence of the spatial density coverage measure  $\eta_V$ , surface coverage  $\eta_A$ , and computed UAV distances during the inspection. (B) shows trajectories of three UAVs performing inspection of the portal structure after 1500 s. Camera orientations along trajectories are displayed as magenta arrows of length equal to the inspection distance  $d_m$ . Surface inspection is indicated by the number of surface observations visualized as green-to-yellow shading of the portal structure surface.

distanced from the structure, complying with the given target density field  $\mu_0$  presented in Fig. 5.

Based on the camera orientation visualization and the surface observation shading shown in Fig. 6(B), all surfaces of the inspected structure are observed relatively uniformly which brings the conclusion that planned paths and camera orientations are suitable for inspection application.

#### 4.4. Wind turbine test case

Wind turbine inspection with UAV technology has become the norm in the past several years as it provides a lot of benefits over manual or ground-based inspection. The main advantages include human safety and high-quality inspection data with a noticeable reduction in data acquisition time. Furthermore, the given benefits of the UAV-based inspection approach enable a reduction in maintenance costs of wind turbine farms [35]. Due to the given rationale, a number of previous studies have included a wind turbine test case to assess the quality

**Table 2**

Parameters for the wind turbine inspection scenario containing domain, structure and numerical mesh info, inspection task parameters, UAV parameters, and HEDAC parameters.

Parameter	Value	Unit
Structure length	25.6	m
Structure width	122.8	m
Structure height	203.1	m
Domain length	65.9	m
Domain width	162.9	m
Domain height	223.1	m
Number of domain mesh nodes	124 979	
Number of domain mesh elements	711 003	
Number of structure mesh surface nodes	11 052	
Number of structure mesh surface faces	22 100	
Inspection distance $d_m$	6	m
Inspection distance broadness $d_\sigma$	1	m
FOV cone height $C_H$	8	m
FOV cone diameter $C_D$	6	m
Number of UAVs	2	
UAV velocity $v$	1.2	m/s
Safety distance $\epsilon$	1	m
Coverage action intensity $\Phi$	400	
Coverage action range $\sigma$	3	m
HEDAC conduction coefficient $k$	5	
Inspection duration	1200	s
Trajectory planning time step $\Delta t$	0.5	s

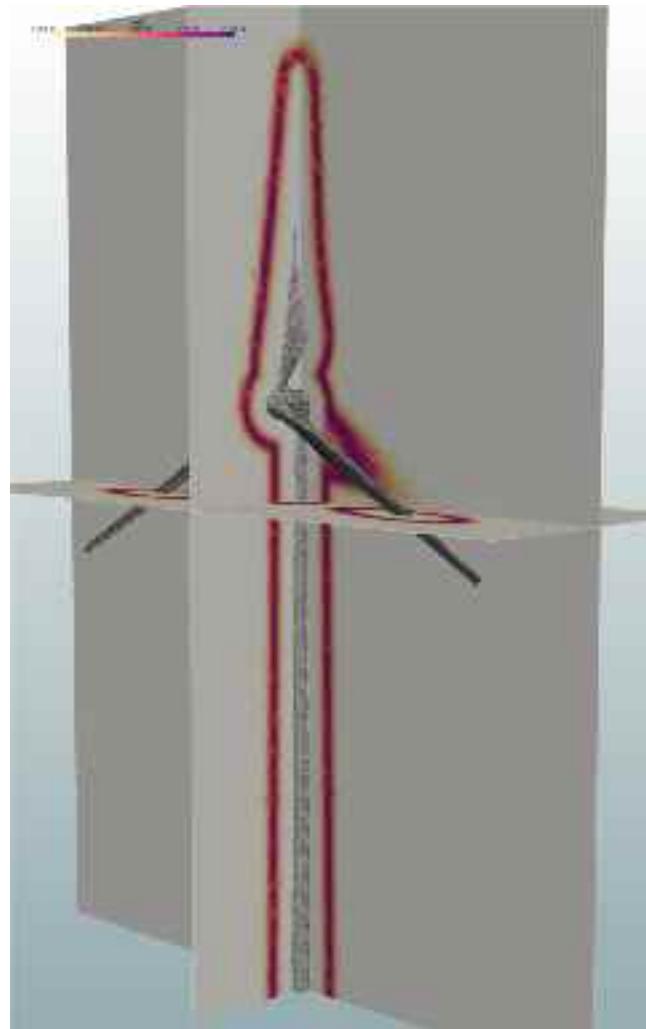
and efficiency of their respective visual inspection algorithms [21,24, 36,37], and hence this paper does the same.

For the wind turbine test case, a path planning for the inspection using two UAVs is prepared according to the parameters shown in [Table 2](#). The structure of the wind turbine consists of a 120 m high vertical column, a hub, and three attached blades, each 80 m long. Blades are rotated at a 60° angle, forming an “upside-down Y” shape, which is a typical orientation for inspection operations. Based on the shape of the wind turbine, the inspection distance  $d_m = 6$  m and the broadness  $d_\sigma = 1$  m, the target density field  $\mu_0$  is computed ([Fig. 7](#)). Note that the numerical domain used in this test case is significantly larger than the space needed for inspection flight in order to allow for the UAV to directly pass from the tip of one blade to the tip of another blade (or to the root of the column), though this possibility is not utilized in performed trajectory planning computations.

The wind turbine inspection typically takes about 40 min during which about 1000 photographs of the turbine surface are recorded. In this inspection scenario, two UAVs are used in order to speed up the inspection operation (the duration is 20 min) and to demonstrate the coordination between two UAVs on a relatively slender structure such as a wind turbine.

A slightly stepped convergence of the spatial density coverage  $\eta_V$  can be observed in [Fig. 8\(A\)](#) due to the UAVs’ inspection focus switching between the wind turbine’s column, hub, and blade. The Y-shaped configuration of the wind turbine is causing multi-pass inspections of the individual components, where each subsequent pass produces flatter change in  $\eta_V$  and finally, it results in stepped coverage performance. Analogous to the first example, the distances of the trajectories from the structure are on average equal to  $d_m = 6$  m and they have an increasing trend during the inspection. Since the trajectory planning is performed only for two UAVs and the domain is considerably large, the spacing between UAVs is easily kept at the safe distance  $\epsilon$ .

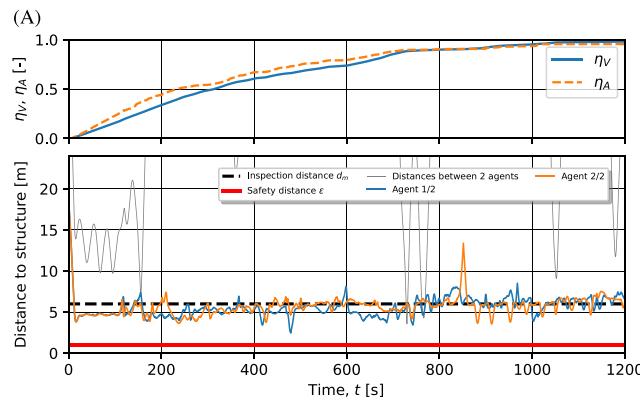
Realized trajectories, camera orientations, and surface observations for wind turbine inspection are shown in [Fig. 8\(B\)](#). It is interesting to observe a synchronized spiral motion caused by the interaction between UAVs’ coverage actions. All components of the turbine structure are observed from practically all directions and UAVs effectively collaborate in the inspection. The presented results imply that the proposed HEDAC trajectory planning algorithm is suitable for carrying out a multi-UAV inspection on real-world structures such as wind turbines.



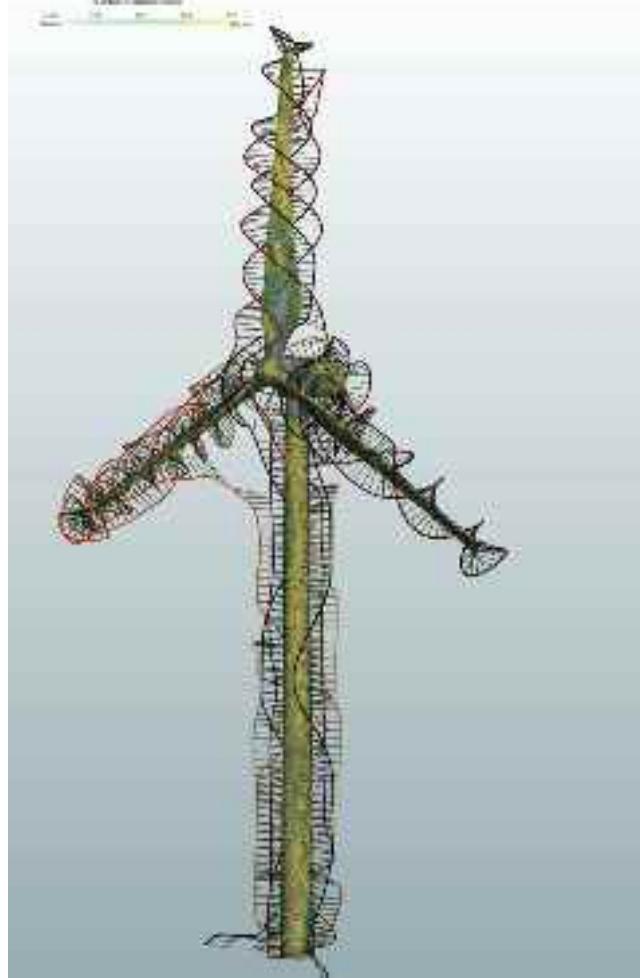
[Fig. 7.](#) Field  $\mu_0$  for the wind turbine test case defines the region aimed to be occupied by UAVs’ trajectories along which all turbine structure faces can be inspected.

#### 4.5. Bridge test case

Bridge inspection using UAVs is commonly conducted to visually detect damage as it lowers inspection costs and increases safety. Usually, UAVs are flown manually to visually inspect the structure using sensors and cameras. There has been a rising interest in research of UAV autonomous bridge inspection where previous studies have attempted to provide path planning and trajectory planning algorithms [18]. The most noticeable one is the receding horizon next-best-view planner [22] which is described in [Section 2](#). The authors employ the algorithm both for the exploration of a domain containing an unknown structure and for the visual inspection of a known structural model. For the inspection test case, the environment is represented as a volumetric occupancy grid map where segments are marked as inspected or uninspected based on the sensor’s readings within 10 m, and paths are planned accordingly. To achieve greater coverage and improve computational performance, the planner produces paths by taking into account sensor readings at a distance of 2 m. A colliding box around the agent assures a safe distance between the agent and the structure. Visual inspection of a bridge structure using a single UAV lasted for  $t_{tot} = 105.0$  min and achieved coverage of 99.1%. Within this section, we reconstruct the same bridge test case, adapt it for HEDAC, and calculate the inspection trajectories with proper camera orientations. All parameters used for the bridge test case and their analogies from [22] are provided



(B)



**Fig. 8.** Relatively non-smooth convergence of the coverage measures  $\eta_V$  and  $\eta_A$  displayed in plot (A) is caused by alternating inspection motion between relatively isolated parts (column, hub, and blades) of the wind turbine structure. The plot of distances shows that, on average, the inspection distance is ensured and the minimal spacing constraint is complied. Trajectories of two UAVs generated by the HEDAC method for the inspection of the wind turbine in a total duration of 1500 s are shown in (B). Camera orientations along trajectories are displayed as arrows of length equal to the inspection distance  $d_m = 6$  m. Green-to-yellow shading of the wind turbine surface indicates that almost all of it is observed between 5 to 20 times.

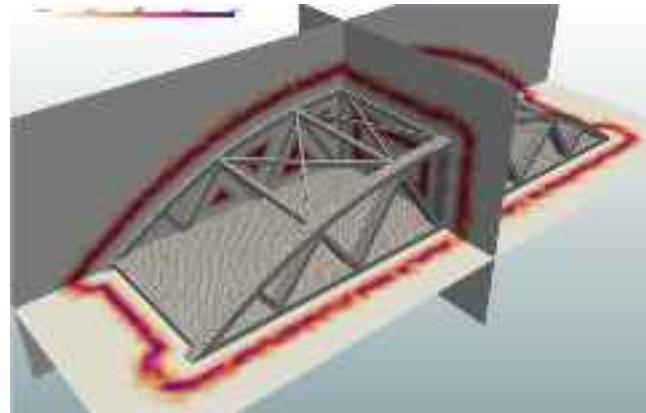
in Table 3. Note that parameter values marked with <sup>b</sup> are used for comparison purposes as presented in Section 5.3.

The structure consists of around 50 m long and 13 m wide bridge deck as well as two 9 m high arches, increasing the inspection complexity when compared to the portal or wind turbine cases. The inspection

**Table 3**

Geometry and numerical mesh info and parameters used for two variants of the bridge test case (values specific for inspection using five UAVs and single UAV are marked with <sup>a</sup> and <sup>b</sup> superscript, respectively) and corresponding or analogous parameters used in [22] (marked with \*). Values marked with <sup>b</sup> are used in Section 5.3 where HEDAC is compared with Receding Horizon approach.

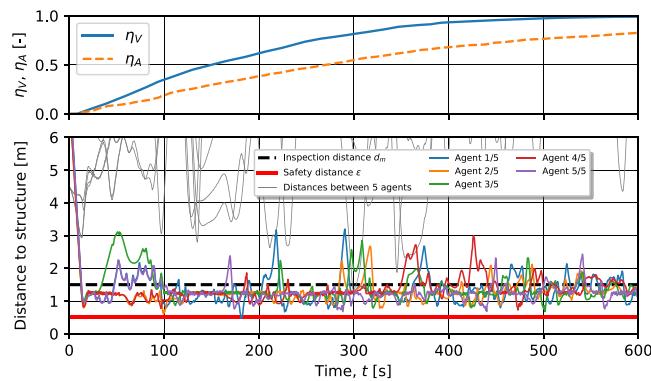
Parameter	Value	Unit
Structure length	47.5	m
Structure width	12.85	m
Structure height	9.0	m
Domain length	58.3, 50*	m
Domain width	22.85, 25*	m
Domain height	19.9, 14*	m
Number of domain mesh nodes	93 539	
Number of domain mesh elements	503 852	
- Based on volumetric map resolution [22]	1120 000*	
Number of structure mesh surface nodes	23 971	
Number of structure mesh surface faces	48 118	
- Based on inspection mesh resolution [22]	140 000*	
Inspection distance $d_m$	1.5	m
- $d_{\text{planner}, \max}$ [22]	2*	m
Inspection distance broadness $d_\sigma$	0.3	m
FOV cone height $C_H$	2 <sup>a</sup> , 10 <sup>b</sup>	m
FOV cone diameter $C_D$	3 <sup>a</sup> , 16 <sup>b</sup>	m
Number of UAVs	5 <sup>a</sup> , 1 <sup>b</sup>	
UAV velocity $v$	0.5, 0.5*	m/s
Safety distance $\epsilon$	0.5	m
- Colliding box [22]	0.5 × 0.5 × 0.3*	m
Coverage action intensity $\Phi$	0.4 <sup>a</sup> , 1 <sup>b</sup>	
Coverage action range $\sigma$	1	m
- $d_{\text{sensor}, \max}$ [22]	10*	m
HEDAC conduction coefficient $k$	20	
Inspection duration	1000 <sup>a</sup> , 3000 <sup>b</sup>	s
Path planning time step $\Delta t$	0.5	s



**Fig. 9.** Spatial regions for bridge inspection trajectory planning are defined using a detailed target density field  $\mu_0$ . The field is constructed around the structure at the distance  $d_m = 1.5$  m and with the inspection broadness  $d_\sigma = 0.3$  m.

domain is represented with the distance field inside the domain size volume. The distance field contains distances from the structure; therefore, leading agents near the structure to inspect. Based on the distance field  $d_s$ , we calculate the target density  $\mu_0$  that covers the space around all structural components of the bridge at the distance  $d_m$  with broadness  $d_\sigma$  as shown in Fig. 9. The implemented collision avoidance algorithm successfully prevents agent-to-agent as well as agent-to-boundary and agent-to-structure collisions.

The convergence of the coverage measure  $E$  and distances from agents to the structure are shown in Fig. 10. The inspection distance is difficult to maintain during the operation due to the complex structure and flight domain. However, the agents are restricted with the safety distance assuring collision avoidance. In a few brief moments, the safety distance is violated while inspecting the bridge with five agents due



**Fig. 10.** Top subplot shows the convergence of spatial and surface coverage for the bridge inspection scenario using five UAVs. Calculated distances are shown in the bottom subplot. It is clearly visible that the inspection distance  $d_m = 1.5$  m is hard to maintain during the operation due to the highly detailed and complex structure and flight domain. Although not as evident as in previous cases, the increasing trend of the UAV-to-structure distance is also present during the bridge inspection.



**Fig. 11.** Obtained trajectories for five UAV bridge inspection planning. Trajectories successfully explore the domain and avoid frame trusses of the bridge structure. The directions of cameras are suitably adjusted and directed towards almost all surfaces of the bridge structure. Shading of the bridge surfaces represents the number of camera inspection shots achieved by movement of UAVs along generated trajectories. Since in the simulation each camera recording is triggered after each time step, the number of observations is accumulated. Due to the crossing of trajectories, the surfaces of some regions are observed more than 20 times. Remaining uninspected areas can be inspected by increasing the inspection time.

to the crowded flight domain, complex geometry, and large time step which prevents the collision avoidance from earlier activation. It can be avoided by reducing the time step or employing fewer agents for the inspection.

Produced trajectories are visualized in Fig. 11, showing appropriate camera orientations along the trajectories and achieved surface coverage.

## 5. Performance and limitations analysis

The use of ergodic trajectory planning is well-suited for the UAV surface inspection tasks presented in this paper. Other approaches usually consider this as an optimization problem, which entails the potential of stalling in local optima and, in general, the limited effectiveness of the optimization itself.

Note that the parameter  $k$  governs the balance between global (exploration) and local (exploitation) coverage. Due to the ergodic nature of the proposed method, it is self-balancing and both global and local coverage are eventually achieved. In theory, the perfect

**Table 4**

Computational times for integral procedures of the proposed UAV trajectory planning algorithm. Procedures marked with \* are being run in each time step and the displayed computational times are calculated as averages of all steps in each inspection simulation. The computational benchmarks are executed using Python 3.8 on a Linux-based operating system, equipped with AMD Ryzen Threadripper 3970X 32-Core processor and 125,7 GiB of RAM.

Procedure	UAV inspection scenario		
	Portal	Wind turbine	Bridge
Initialization	19.7 s	114.2 s	237.5 s
Coverage $\rho$ calculation*	1.631 s	7.415 s	5.132 s
Potential $\psi$ calculation*	0.832 s	7.288 s	4.173 s
Collision avoidance*	0.003 s	0.002 s	0.094 s
Time step (all calculations)*	2.465 s	14.705 s	9.399 s
Entire trajectory planning	3717.2 s	35 406.2 s	19 035.5 s

coverage is achieved in  $\lim_{t \rightarrow \infty}$ , while in practical applications, a near-optimal coverage can be achieved in a finite time. The duration of this finite time window depends on the choice of UAV motion and vision parameters ( $\phi_\sigma$  or FOV) and on the scale and complexity of the inspected structure.

In the following subsections, we analyze the performance and shortcomings of the proposed method for UAV inspection trajectory planning. Validation of the method is carried out through comparison with a state-of-the-art approach on a standardized structure inspection scenario and is presented in Section 5.3.

### 5.1. Computational performance analysis

One of the main drawbacks of our proposed method is its computational inefficiency. Handling a relatively large three-dimensional numerical mesh, when calculating coverage and potential, is very demanding. In this subsection, we analyze the computational costs of the method in more detail.

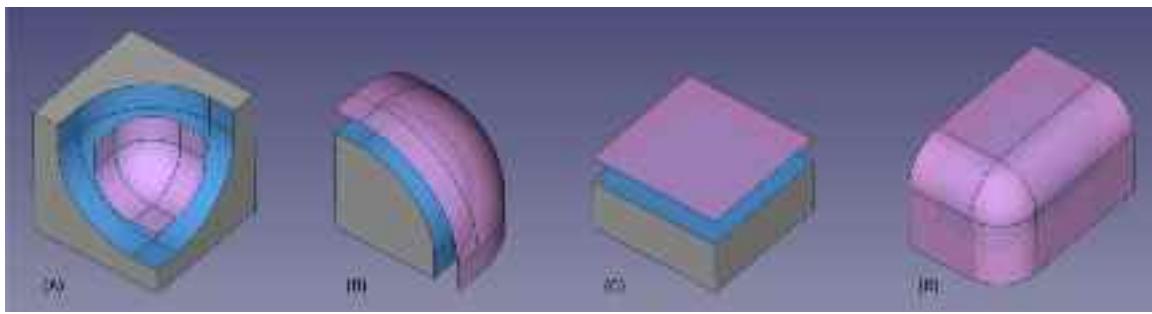
The initialization of the FEM in Table 4 is computationally demanding because the sparse matrix is inverted, but it can still be done on a PC. This ensures a low-computational cost of FEM run-time calculation which reduces to only one matrix–vector multiplication at each time step. The optimization problem related to the collision avoidance can be computationally very demanding, but in this application it is formulated as an almost linear programming problem and does not require significant resources. The cost of the collision avoidance algorithm in practice can be seen in Table 4. The cost obviously depends on the particular test case, but it is almost insignificant compared to other parts of the algorithm.

Although the proposed three-dimensional HEADC methodology is designed as a motion control algorithm, it is obvious that, due to excessive computational demands, it cannot deliver results in real time. That is why we present this method as trajectory planning instead of motion control.

### 5.2. Space coverage vs. surface coverage

The proposed trajectory planning method utilizes a space coverage technique in order to achieve coverage of two-dimensional surfaces. This approach is not justified in an obvious way, since space and surface coverage are not equivalent. However, as shown in the previous section, our approach can yield a suitable approximation, appropriate for real-world application. Nevertheless, the effect of this approximation needs to be addressed and assessed in pursuance of an overall evaluation of the proposed trajectory planning methodology.

The target density field  $\mu_0$  is obtained by convolution of the Gaussian action  $\phi_\sigma$  over the offsets of the structure's surfaces. The broadness of the target density field, regulated by the broadness  $d_\sigma$ , has an insignificant influence on the surface-to-space coverage mapping. However, the offset distance, regulated by the goal inspection distance  $d_m$ ,



**Fig. 12.** The offset of a structure's surface depends on the surface curvature: (A) offset area of a concave surface decreases, (B) offset area of a convex surface increases, (C) offset area of a planar surface remains the same and (D) offset area for sharp edges and corners (of zero area) consists of cylindrical and spherical surfaces (infinite area increase). Referent and offset surfaces are marked with blue and transparent pink colors, respectively.

plays a significant role. It is proportional to the surface-to-space mapping error. Using  $d_\sigma = 0$  (Gaussian function becomes Dirac function, providing infinitesimally thin goal density) and using offset  $d_m = 0$  would result in a spatial density  $\mu_0$  that is equivalent to the structure's surface. Compared to the referent surface, the area of the offset surface can generally be the same, larger or smaller, for planar, convex, or concave surfaces, respectively (Fig. 12). The influence of offsetting is not trivial [38], but it depends on surface curvature and offset distance.

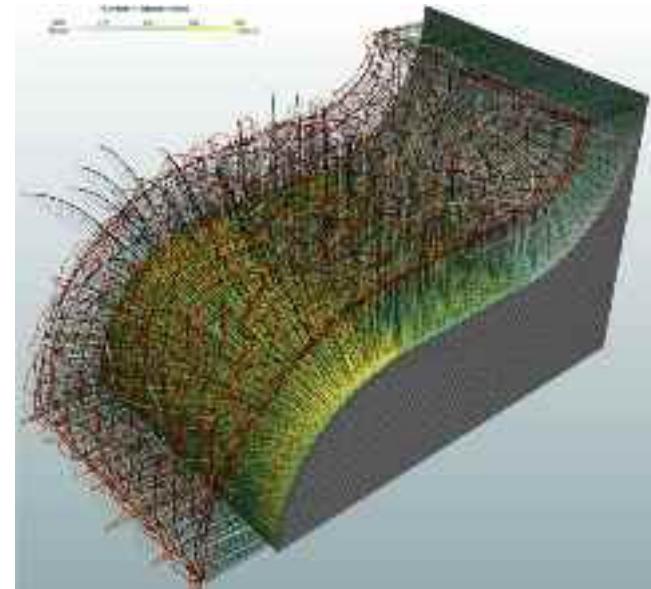
An inspection simulation on a relatively simple domain is prepared in order to demonstrate the effect of surface curvature on the inspection quality. The structure and the surrounding domain are designed to expose only one side of the S-shaped surface to the UAVs (Fig. 13). The S-shaped surface has concave, convex, and flat regions. We simulate the inspection conducted by 20 UAVs for a duration of 1000 s. One can easily recognize a different number of achieved observations in each of the three regions, although the spatial target density  $\mu_0$  is uniform (using the same distance  $d_m$  and broadness  $d_\sigma$ ) for all regions. These results confirm the considerations presented in the first part of this section.

Differences between spatial and surface coverage are strongly manifested in frequent and dense trajectories over certain regions. Since a relatively small surface coverage is required for inspection (only a few shots of the same point are sufficient), achieving spatial coverage is an adequate solution for achieving surface coverage in UAV inspection tasks.

### 5.3. Comparison with receding horizon

We utilize the receding horizon methodology for trajectory planning and the bridge test case, presented in [22], in order to evaluate the presented HEDAC trajectory generation approach. Since both compared approaches are designed differently, the parameters used in the trajectory planning procedures are different. In order to provide a scenario as equivalent as possible to the one in [22], we adjust the bridge case parameters (marked with <sup>b</sup> in Table 3). Additionally, non-identical but corresponding parameters used in the receding horizon approach are marked with \*.

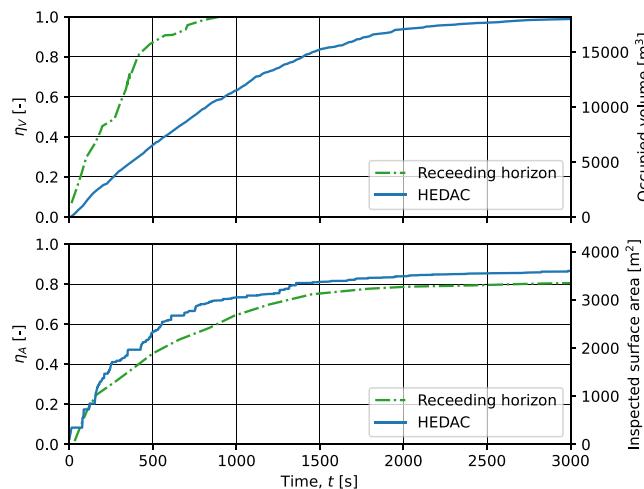
The upper plot in Fig. 14 shows the comparison of spatial target density vs. free volume achieved with spatial ergodic coverage using the HEDAC method and volumetric exploration using the receding horizon [22], respectively. It is noticeable that the HEDAC approach occupies less space than the receding horizon. This can be easily explained by the fact that HEDAC uses only a part of the available space defined by  $\mu_0 > 0$ , while the receding horizon considers the entire volume available for flying. Focusing on the space around the surfaces being observed utilizes less volume and produces a more efficient inspection. This is visible in the lower plot of Fig. 14, which shows the convergence of the area of inspected surfaces. Although it uses more conservative metrics (an effectively smaller FOV), the HEDAC approach outperforms the receding horizon for the bridge structure inspection presented in [22].



**Fig. 13.** Simulation of inspection of a surface with convex, concave and flat regions. All trajectories are located at the offset of the observed surface as defined by the target density  $\mu_0$ . Different numbers of recordings can be observed in each region of the plot, varying from 50 to 300. The non-linearity of delegating the surface coverage problem to space coverage is also indicated by the density of the camera direction vector (which coincides with the normal to the surface).

## 6. Conclusions

Due to the fast development of UAV technology and the potential cost benefits it could bring, autonomous tasks are prominent topics for contribution to the industries such as civil infrastructure inspection or search and rescue operations. To enable autonomous flight, safe paths need to be determined and provided to the UAVs assuring collision avoidance within the environment. In this paper, we have presented a new ergodic method for multiple UAV trajectory planning within a known three-dimensional environment as well as its application for visual infrastructure inspection. The method is an extension of the HEDAC algorithm, previously developed and tested for the exploration of two-dimensional domains. The presented algorithm calculates trajectories that cover three-dimensional space according to the given target density. The UAV motion is represented using a simple, first-order kinematic model allowing the calculation of trajectories feasible for a UAV. Collision avoidance is formulated as a non-linear optimization problem, successfully preventing collisions between UAVs as well as UAVs with domain boundaries by gradually redirecting a UAV when the



**Fig. 14.** Convergence of spatial coverage  $\eta_V$  (or free volume [22], marked on the right-hand y-axis) for the bridge scenario using the proposed HEDAC method and receding horizon [22] is shown on the upper plot. HEDAC occupies less volume by focusing on the space around the bridge surfaces allowing a more efficient surface inspection as shown in the lower plot. The lower plot shows surface coverage  $\eta_A$  (or inspected surface area [22], marked on the right-hand y-axis).

distance threshold is reached. We have validated the implementation of collision avoidance in a crowded, unit cube domain with 100 UAVs uniformly covering the lower half of the cube.

We have further adapted the algorithm for the visual inspection of three-dimensional structures using UAVs. In mathematical terms, the task is to explore two-dimensional manifolds embedded in a bounded three-dimensional volume using sensors mounted on a UAV, e.g., a camera. We have achieved this by constructing a three-dimensional field of interest (target density) as an offset from the structure's surface to be inspected. We have proposed a simple idea for handling the camera orientation. It relies on the gradient of the distance field  $d_s$  containing distances to the nearest surface of the inspected structure. This solution produces camera orientations that always point to the nearest point on the inspected structure.

The proposed method has been tested on three inspection applications. The first test involves a synthetic portal scenario with 3 UAVs inspecting the structure. For a given target density field, continuous trajectories, safely distanced from the structure, are produced and the coverage convergence is achieved. The surfaces of the structure are covered relatively uniformly, leading to the conclusion that planned trajectories and camera orientations are suitable for infrastructure inspection with appropriate camera equipment. The second scenario of a wind turbine inspection provides a realistic test case demonstrating the coordination of 2 UAVs inspecting a relatively slender structure. As all components of the wind turbine are observed, we can implictively conclude that the algorithm is suitable for conducting an autonomous UAV inspection of real-world structures. The third scenario of a bridge inspection has been successfully conducted with five UAVs inspecting the structure. The bridge scenario was also utilized in comparison with receding horizon inspection path planning. The results of the comparison indicate the competitiveness of the proposed approach, considering that the HEDAC method has achieved a better quality inspection.

Trajectories obtained with the proposed planner, though seemingly chaotic, are uniformly exploring the region around the 3D structure and allow the inspection of almost all surfaces of the structure in a limited time window. Although the differences between spatial and surface

coverage are highlighted, the proposed approach is adequate for non-exhaustive, single-pass UAV inspection. The convergence of the coverage measure is achieved for each scenario, with convergence depending on the complexity of the inspected structure, the number of UAVs inspecting the structure, and the available time window. The method is proven to be robust and stable, though it is not computationally efficient enough for real-time motion control applications.

Some ideas for further research related to this topic, such as better domain mesh generation, adaptive mesh refinement, solving potential using General-Purpose computing on Graphics Processing Units (GPGPU), or domain partitioning, could potentially rise computational efficiency enough for real-time motion control implementation. Possible applications of the proposed methodology are numerous, not only for the tasks of UAV inspection of complex structures, but also for some contact actions such as UAV spraying or surface cleaning. A more accurate surface coverage solution could be solved by more advanced mapping of the surface to the target spatial density  $\mu_0$  or by using a non-uniform motion model driven by the potential calculated directly on the surface of the structure using shell elements. The camera model is naive in the proposed inspection model (simplified FOV, zooming is not considered, and camera rotation is not limited), so for a real application, a suitable camera control should be designed in addition to UAV motion planning.

#### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Data availability

All parameters for reproducing the study are presented in the manuscript. The data needed to reproduce the presented UAV inspection scenarios and video animations are available on the Open Science Framework repository: <https://osf.io/bdrvn/>. The Python code needed to reproduce this research is available upon request.

#### Acknowledgments

This research is primarily supported by the Croatian Science Foundation under the project UIP-2020-02-5090. B.C.'s contribution is supported by Croatian Science Foundation under the project IP-2019-04-1239, L. M.'s contribution is supported by the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 861111, Drones4Safety.

#### References

- [1] M. Moshref-Javadi, M. Winkenbach, Applications and Research avenues for drone-based models in logistics: A classification and review, *Expert Syst. Appl.* 177 (2021) 114854, <http://dx.doi.org/10.1016/j.eswa.2021.114854>.
- [2] S. Biçici, M. Zeybek, An approach for the automated extraction of road surface distress from a UAV-derived point cloud, *Autom. Constr.* 122 (2021) 103475, <http://dx.doi.org/10.1016/j.autcon.2020.103475>.
- [3] S. Zhao, F. Kang, J. Li, C. Ma, Structural health monitoring and inspection of dams based on UAV photogrammetry with image 3D reconstruction, *Autom. Constr.* 130 (2021) 103832, <http://dx.doi.org/10.1016/j.autcon.2021.103832>.
- [4] D. Kim, M. Liu, S. Lee, V.R. Kamat, Remote proximity monitoring between mobile construction resources using camera-mounted UAVs, *Autom. Constr.* 99 (2019) 168–182, <http://dx.doi.org/10.1016/j.autcon.2018.12.014>.
- [5] I. Martinez-Alpiste, G. Golcarenarenji, Q. Wang, J.M. Alcaraz-Calero, Search and rescue operation using UAVs: A case study, *Expert Syst. Appl.* 178 (2021) 114937, <http://dx.doi.org/10.1016/j.eswa.2021.114937>.
- [6] S.W. Cho, H.J. Park, H. Lee, D.H. Shim, S.-Y. Kim, Coverage path planning for multiple unmanned aerial vehicles in maritime search and rescue operations, *Comput. Ind. Eng.* 161 (2021) 107612, <http://dx.doi.org/10.1016/j.cie.2021.107612>.

- [7] J.F. Falorca, J.P.N.D. Miraldes, J.C.G. Lanzinha, New trends in visual inspection of buildings and structures: Study for the use of drones, *Open Eng.* 11 (1) (2021) 734–743, <http://dx.doi.org/10.1515/eng-2021-0071>.
- [8] H. Shakhatreh, A.H. Sawalmech, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N.S. Othman, A. Khereishah, M. Guizani, Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges, *IEEE Access* 7 (2019) 48572–48634, <http://dx.doi.org/10.1109/ACCESS.2019.2909530>.
- [9] A. Gasparetto, P. Boscariol, A. Lanzutti, R. Vidoni, Path planning and trajectory planning algorithms: A general overview, in: G. Carbone, F. Gomez-Bravo (Eds.), *Motion and Operation Planning of Robotic Systems*, Vol. 29, in: *Mechanisms and Machine Science*, Springer International Publishing, Cham, 2015, pp. 3–27, [http://dx.doi.org/10.1007/978-3-319-14705-5\\_1](http://dx.doi.org/10.1007/978-3-319-14705-5_1).
- [10] E. Galceran, M. Carreras, A survey on coverage path planning for robotics, *Robot. Auton. Syst.* 61 (12) (2013) 1258–1276, <http://dx.doi.org/10.1016/j.robot.2013.09.004>.
- [11] S. Ivic, B. Crnkovic, I. Mezic, Ergodicity-based cooperative multiagent area coverage via a potential field, *IEEE Trans. Cybern.* 47 (8) (2017) 1983–1993, <http://dx.doi.org/10.1109/TCYB.2016.2634400>.
- [12] M. Torres, D.A. Pelta, J.L. Verdegay, J.C. Torres, Coverage path planning with unmanned aerial vehicles for 3D terrain reconstruction, *Expert Syst. Appl.* 55 (2016) 441–451, <http://dx.doi.org/10.1016/j.eswa.2016.02.007>.
- [13] R. Almadhoun, T. Taha, L. Seneviratne, Y. Zweiri, A survey on multi-robot coverage path planning for model reconstruction and mapping, *SN Appl. Sci.* 1 (8) (2019) 847, <http://dx.doi.org/10.1007/s42452-019-0872-y>.
- [14] Y. Tan, S. Li, H. Liu, P. Chen, Z. Zhou, Automatic inspection data collection of building surface based on BIM and UAV, *Autom. Constr.* 131 (2021) 103881, <http://dx.doi.org/10.1016/j.autcon.2021.103881>.
- [15] Z. Shang, J. Bradley, Z. Shen, A co-optimal coverage path planning method for aerial scanning of complex structures, *Expert Syst. Appl.* 158 (2020) 113535, <http://dx.doi.org/10.1016/j.eswa.2020.113535>.
- [16] M.D. Phung, C.H. Quach, T.H. Dinh, Q. Ha, Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection, *Autom. Constr.* 81 (2017) 25–33, <http://dx.doi.org/10.1016/j.autcon.2017.04.013>.
- [17] C. Cao, J. Zhang, M. Travers, H. Choset, Hierarchical coverage path planning in complex 3D environments, in: 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE, Paris, France, 2020, pp. 3206–3212, <http://dx.doi.org/10.1109/ICRA40945.2020.9196575>.
- [18] L. Shi, G. Mehrooz, R.H. Jacobsen, Inspection path planning for aerial vehicles via sampling-based sequential optimization, in: 2021 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, Athens, Greece, 2021, pp. 679–687, <http://dx.doi.org/10.1109/ICUAS51884.2021.9476784>.
- [19] S. Jung, S. Song, P. Youn, H. Myung, Multi-layer coverage path planner for autonomous structural inspection of high-rise structures, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Madrid, 2018, pp. 1–9, <http://dx.doi.org/10.1109/IROS.2018.8593537>.
- [20] A. Bircher, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, R. Siegwart, Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics, in: 2015 IEEE International Conference on Robotics and Automation (ICRA), IEEE, Seattle, WA, USA, 2015, pp. 6423–6430, <http://dx.doi.org/10.1109/ICRA.2015.7140101>.
- [21] A. Bircher, M. Kamel, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, R. Siegwart, Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots, *Auton. Robots* 40 (6) (2016) 1059–1078, <http://dx.doi.org/10.1007/s10514-015-9517-1>.
- [22] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, R. Siegwart, Receding horizon path planning for 3D exploration and surface inspection, *Auton. Robots* 42 (2) (2018) 291–306, <http://dx.doi.org/10.1007/s10514-016-9610-0>.
- [23] H.I.A. Perez-imaz, P.A.F. Rezeck, D.G. Macharet, M.F.M. Campos, Multi-robot 3D coverage path planning for first responders teams, in: 2016 IEEE International Conference on Automation Science and Engineering (CASE), IEEE, Fort Worth, TX, USA, 2016, pp. 1374–1379, <http://dx.doi.org/10.1109/COASE.2016.7743569>.
- [24] S.S. Mansouri, C. Kanellakis, E. Fresk, D. Kominak, G. Nikolakopoulos, Cooperative coverage path planning for visual inspection, *Control Eng. Pract.* 74 (2018) 118–131, <http://dx.doi.org/10.1016/j.conengprac.2018.03.002>.
- [25] W. Jing, D. Deng, Y. Wu, K. Shimada, Multi-UAV coverage path planning for the inspection of large and complex structures, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Las Vegas, NV, USA, 2020, pp. 1480–1486, <http://dx.doi.org/10.1109/IROS45743.2020.9341089>.
- [26] S. Ivić, A. Andreječuk, S. Družeta, Autonomous control for multi-agent non-uniform spraying, *Appl. Soft Comput.* 80 (2019) 742–760, <http://dx.doi.org/10.1016/j.asoc.2019.05.001>.
- [27] S. Ivić, Motion control for autonomous heterogeneous multiagent area search in uncertain conditions, *IEEE Trans. Cybern.* 52 (5) (2022) 3123–3135, <http://dx.doi.org/10.1109/TCYB.2020.3022952>.
- [28] S. Ivić, A. Sikirica, B. Crnković, Constrained multi-agent ergodic area surveying control based on finite element approximation of the potential field, *Eng. Appl. Artif. Intell.* 116 (2022) 105441, <http://dx.doi.org/10.1016/j.engappai.2022.105441>.
- [29] Y. Zheng, C. Zhai, Distributed coverage control of multi-agent systems in uncertain environments using heat transfer equations, 2022, <http://dx.doi.org/10.48550/ARXIV.2204.09289>, Publisher: arXiv Version Number: 1.
- [30] T. Low, J. Maceiras, S. Calinon, drozBot: Using ergodic control to draw portraits, *IEEE Robot. Autom. Lett.* 7 (4) (2022) 11728–11734, <http://dx.doi.org/10.1109/LRA.2022.3186735>.
- [31] G. Mathew, I. Mezić, Metrics for ergodicity and design of ergodic dynamics for multi-agent systems, *Physica D* 240 (4–5) (2011) 432–442, <http://dx.doi.org/10.1016/j.physd.2010.10.010>.
- [32] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, İ. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, A. Vijaykumar, A.P. Bardelli, A. Rothberg, A. Hilboll, A. Kloeckner, A. Scopatz, A. Lee, A. Rokem, C.N. Woods, C. Fulton, C. Masson, C. Häggström, C. Fitzgerald, D.A. Nicolson, D.R. Hagen, D.V. Pasechnik, E. Olivetti, E. Martin, E. Wieser, F. Silva, F. Lenders, F. Wilhelm, G. Young, G.A. Price, G.-L. Ingold, G.E. Allen, G.R. Lee, H. Audren, I. Probst, J.P. Dietrich, J. Silterra, J.T. Webber, J. Slavić, J. Nothman, J. Buchner, J. Kulick, J.L. Schönenberger, J.V. de Miranda Cardoso, J. Reimer, J. Harrington, J.L.C. Rodríguez, J. Nunez-Iglesias, J. Kuczynski, K. Tritz, M. Thoma, M. Newville, M. Kümmeler, M. Bolingbroke, M. Tartre, M. Pak, N.J. Smith, N. Nowaczyk, N. Shebanov, O. Pavlyk, P.A. Brodtkorb, P. Lee, R.T. McGibbon, R. Feldbauer, S. Lewis, S. Tygier, S. Sievert, S. Vigna, S. Peterson, S. More, T. Pudlak, T. Oshima, T.J. Pingel, T.P. Robitaille, T. Spura, T.R. Jones, T. Cera, T. Leslie, T. Zito, T. Krauss, U. Upadhyay, Y.O. Halchenko, Y. Vázquez-Baeza, SciPy 1.0: fundamental algorithms for scientific computing in Python, *Nature Methods* 17 (3) (2020) 261–272, <http://dx.doi.org/10.1038/s41592-019-0686-2>.
- [33] A.R. Conn, N.I.M. Gould, P.L. Toint, Trust Region Methods, Society for Industrial and Applied Mathematics, 2000, <http://dx.doi.org/10.1137/1.9780898719857>.
- [34] J. Schöberl, C++11 Implementation of Finite Elements in NGSolve, ASC Report No. 30/2014, 2014, <https://www.asc.tuwien.ac.at/preprint/2014/asc30x2014.pdf>, last accessed on 2022-05-02.
- [35] K. Kabbabe Poleo, W.J. Crowther, M. Barnes, Estimating the impact of drone-based inspection on the Levelised Cost of electricity for offshore wind farms, *Res. Eng.* 9 (2021) 100201, <http://dx.doi.org/10.1016/j.rinem.2021.100201>.
- [36] M. Stokkeland, K. Klausen, T.A. Johansen, Autonomous visual navigation of Unmanned Aerial Vehicle for wind turbine inspection, in: 2015 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, Denver, CO, USA, 2015, pp. 998–1007, <http://dx.doi.org/10.1109/ICUAS.2015.7152389>.
- [37] B.E. Schafer, D. Picchi, T. Engelhardt, D. Abel, Multicopter unmanned aerial vehicle for automated inspection of wind turbines, in: 2016 24th Mediterranean Conference on Control and Automation (MED), IEEE, Athens, Greece, 2016, pp. 244–249, <http://dx.doi.org/10.1109/MED.2016.7536055>.
- [38] W. Zhuo, J. Rossignac, Curvature-based offset distance: Implementations and applications, *Comput. Graph.* 36 (5) (2012) 445–454, <http://dx.doi.org/10.1016/j.cag.2012.03.013>.