



Optimized structural inspection path planning for automated unmanned aerial systems



Yuxiang Zhao^{a,*}, Benhao Lu^b, Mohamad Alipour^a

^a Dept. of Civil and Environmental Engineering, University of Illinois Urbana Champaign, United States of America

^b Siebel School of Computing and Data Science, University of Illinois Urbana Champaign, United States of America

ARTICLE INFO

Keywords:

Bridge inspection
Optimization
UAS
Path planning

ABSTRACT

Automation in Unmanned Aerial Systems (UAS)-based structural inspections has gained significant traction given the scale and complexity of infrastructure. A core problem in UAS-based inspection is electing an optimal flight path to achieve the mission objectives while minimizing flight time. This paper presents an effective two-stage method that guarantees coverage as a constraint to ensure damage detectability, while minimizing path length as an objective. A genetic algorithm first determines viewpoint positions, and a greedy algorithm calculates the camera poses, as opposed to directly optimizing all degrees of freedom (DOF) simultaneously. A sensitivity analysis demonstrates the range of applicability and superiority of this formulation over direct 5-DOF optimization by at least 30 % shorter path length. Applied examples, including focused and partial space inspections, are also presented, demonstrating the flexibility of the proposed method to meet real-world requirements. The results highlight the feasibility of the approach and contribute to incorporating automation into UAS-based structural inspections.

1. Background

A significant portion of the essential infrastructure that supports society today was constructed several decades ago, and some have already exceeded their intended lifespan. According to the 2021 America's Infrastructure Report Card [1], among all the 617,000 bridges in the U.S., 42 % were constructed more than 50 years ago. Considering the massive infrastructure network and expensive labor in the U.S., substantial investment in manual maintenance is required to ensure personal and property safety. Furthermore, given the challenging working conditions in infrastructure inspection, manual labor always carries the risk of accidents, including traffic incidents, falls, and exposure to environmental hazards. Although there are no specific statistics on the fatalities and injuries during inspection tasks, the growing inventory of aging infrastructure increases the risk of accidents during inspections, thus highlighting the importance of ensuring inspection safety. Numerous studies have demonstrated that the implementation of UAS technology contributes to the protection of workers' safety [2–4].

The application of robotic systems for conducting infrastructure inspections to achieve automated structural assessment has garnered significant attention from scholars [5–9]. Among different robotic systems,

Unmanned Aerial Systems (UAS) have emerged as a portable, cost-effective, and efficient solution for inspection [10]. Numerous studies have been conducted to investigate the potential of utilizing UAS for improving the inspection of infrastructure assets, including power lines [11,12], buildings [13–15], railways [16,17], and bridges [18–20]. In [21], the potential of applying UAS as a platform for passive optical remote sensing was discussed. In [22], a construction inspection framework was developed, utilizing BIM models as input and generating automated UAS inspection plans, which can be executed through mission planning software. In [23], a UAS-based visual inspection framework was implemented to enhance the degree of automation and quality of results for evaluating bridge deck surface distress. In [24], a UAS-based end-to-end bridge inspection pipeline has been proposed, which promotes the practical application of autonomous structural inspection with UAS. In [25], an autonomous point cloud based navigation method is developed for 3D data collection with UAS, using an object-detection-based potential vector field method for mission planning. The application of UAS significantly improves the efficiency and quality of the data collection phase, and computer vision and deep learning techniques have seen a rapid increase in popularity for subsequent automated data processing.

* Corresponding author.

E-mail addresses: zhao132@illinois.edu (Y. Zhao), benhao12@illinois.edu (B. Lu), alipour@illinois.edu (M. Alipour).

The studies mentioned above have demonstrated the contribution of UAS to improving inspection techniques. However, further automation in planning and controlling UAS flights is crucial to reduce manual labor and enhance robustness. While deep learning-based data processing methods promise to improve automated infrastructure inspections, the quality of the resulting inspections is highly sensitive to the selection of UAS viewpoints and poses. Additionally, given the large scale of civil infrastructures and the complexity of surrounding environments, effective path planning is essential for enhancing the practical value of UAS-based inspection. There have been multiple explorations on path planning for infrastructure inspection tasks, and the solutions can be categorized into classical approaches and artificial intelligence (AI) based approaches. Some studies regard the path planning task as a Coverage Path Planning (CPP) problem and solve it with various classical search algorithms such as greedy next-best-view [26], Monte Carlo Tree Search [27], Rapid Random Tree (RRT*) [28], and Probabilistic Roadmap (PRM) [29]. However, given the complex space configurations and strict inspection requirements, the classical search algorithms could be less efficient and may produce only locally optimal solutions [30,31]. Therefore, AI-based algorithms have recently gained rapidly rising tractions due to their advantages, such as a better balance between exploration and exploitation, the ability to avoid locally optimal solutions, and easier parallel programming to improve computational speed [32]. Many studies have attempted to solve path planning problems with various AI-based algorithms, including Ant Colony Optimization [33,34], Particle Swarm Optimization [35] [36] [37], and Genetic Algorithm [38–40]. These works mainly focus on topics other than structural inspections, including construction progress monitoring, ship inspection, geomagnetic navigation, and general robotic path planning algorithms, leaving a research gap in path planning for structural inspections.

2. Introduction

In engineering practice, structural inspection can be accomplished using various types of sensors, such as cameras, thermal cameras [41], LiDAR [42], Ground Penetrating Radar (GPR) [43], etc. However, the effective ranges of these sensors are often limited, which can impact the resolution and quality of inspection results. Therefore, selecting an optimal set of viewpoints is crucial for ensuring the comprehensiveness and reliability of the inspection process. The process of determining proper inspection paths is defined as *structural inspection path planning* (SIPP), which constitutes a specialized form of robotic path planning characterized by its alignment with structural inspection requirements and domain expertise.

In contrast with conventional path-planning endeavors, SIPP displays distinctive attributes. Primarily, SIPP is unique due to its intrinsic connection to structural inspection, where neglecting potential damage could lead to member failures and even the collapse of infrastructure, resulting in loss of life and property. Therefore, SIPP has stricter requirements for inspection coverage [44]. Unlike path planning for other tasks, which often involve striking a balance between coverage and efficiency, in SIPP, coverage should be treated as a stringent constraint to ensure the quality and reliability of the inspection results. Conversely, efficiency is regarded as the primary optimization objective. The optimization approach proposed in this study allows us to derive the optimal path while upholding a predetermined coverage guarantee.

Another distinctive attribute of SIPP is its acknowledgment of varying importance levels within various parts of the infrastructure [45]. Certain elements, like the connections, expansion/construction joints, and beam midspans, can be subject to higher deterioration or load demands and may, therefore, hold greater significance [46]. Consequently, uniform inspection may be inadequate, and SIPP necessitates the allocation of distinct weights to different sections to facilitate damage-guided inspection. These weights can be derived through a variety of methods:

- 1) Previous Inspection Records: Due to the gradual progression of damage in structures, the damage distribution gleaned from previous inspection or monitoring records can be utilized to accentuate areas where damage has already manifested and is, therefore, more likely to be present.
- 2) Structural Analysis: Analytical approaches or numerical simulations such as finite element analysis (FEA) can provide insights into which segments of the structure are critical and exhibit an elevated likelihood of sustaining damage. Leveraging this information, areas of heightened risk can be allocated higher weights during path planning.
- 3) Inspector Expertise: In instances where conducting FEA might not be cost-effective, inspectors can resort to a user interface to stipulate the importance factor for various elements, drawing upon their domain expertise. By aligning with this importance factor, SIPP can prioritize the predefined crucial elements, directing a greater emphasis on their inspection.

Safety constitutes another pivotal factor that underscores the distinctiveness of SIPP [47]. Infrastructures often feature intricate outlines, thereby presenting many obstacles within the 3D space that must be circumvented during inspection. Furthermore, in scenarios like bridge inspections, the presence of UAS in close vicinity to active traffic poses the risk of distracting drivers, potentially leading to traffic accidents [48]. Thus, incorporating functionality within SIPP that enables the delineation of exclusive zones, such as minimum distance thresholds to infrastructure surfaces or the area directly above a bridge deck, becomes necessary.

In summary, SIPP represents a unique path planning challenge within a complex 3D space, with specialized requirements and constraints, setting it apart from traditional path planning problems. The quality of SIPP determines the reliability and time efficiency of UAS-based infrastructure inspection; therefore, how to perform an efficient and safe SIPP is a significant research problem.

A number of explorations on planning inspections have been reported in the literature. A straightforward solution is navigating the UAS according to predetermined rules. Narazaki et al. [49] proposed a rule-based path planning method based on the visual recognition of structural members. However, rule-based methods are, by nature, more applicable and effective for predefined types of infrastructure and are challenging to apply to structures with unique shapes. Stringent constraints involving inspection quality, safety requirements, and inspection efficiency pose challenges in UAS navigation using predetermined rules, prompting the need for optimization-based methods.

Optimization algorithms have also been used for UAS-based infrastructure scanning mission planning based on geometric models, as summarized in [50]. Some studies [51,52] focus on selecting an optimal set of viewpoints that provide the desired 3D reconstruction quality and optimized efficiency. To further consider the spatial relationship of the viewpoints for inspection efficiency, instead of only the number of viewpoints, a common strategy is to first sample viewpoints with good performances and then connect these points with a Traveler Salesman Problem (TSP) solver. Huang et al. [53] proposed a fast viewpoint sampling method using a Next Best View (NBV) strategy, which enables iteratively updating the geometric model and NBV planning based on the model. Zhou et al. [54] presented a viewpoint sampling method with Max-Min optimization based on urban building models generated by 2D maps and satellite images, eliminating the need for a pre-scan to get initial models. Smith et al. [55] applied heuristics optimization methods for determining viewpoints considering multi-view stereo reconstruction quality. Yan et al. [56] proposed two types of viewpoint selection strategies, including a global viewpoint selection strategy to ensure coverage and a local viewpoint selection strategy to optimize inspection quality. After determining two categories of viewpoints based on the two strategies, an Ant Colony Optimization (ACO) algorithm was proved to be most efficient to traverse all these viewpoints. Peng et al. [57]

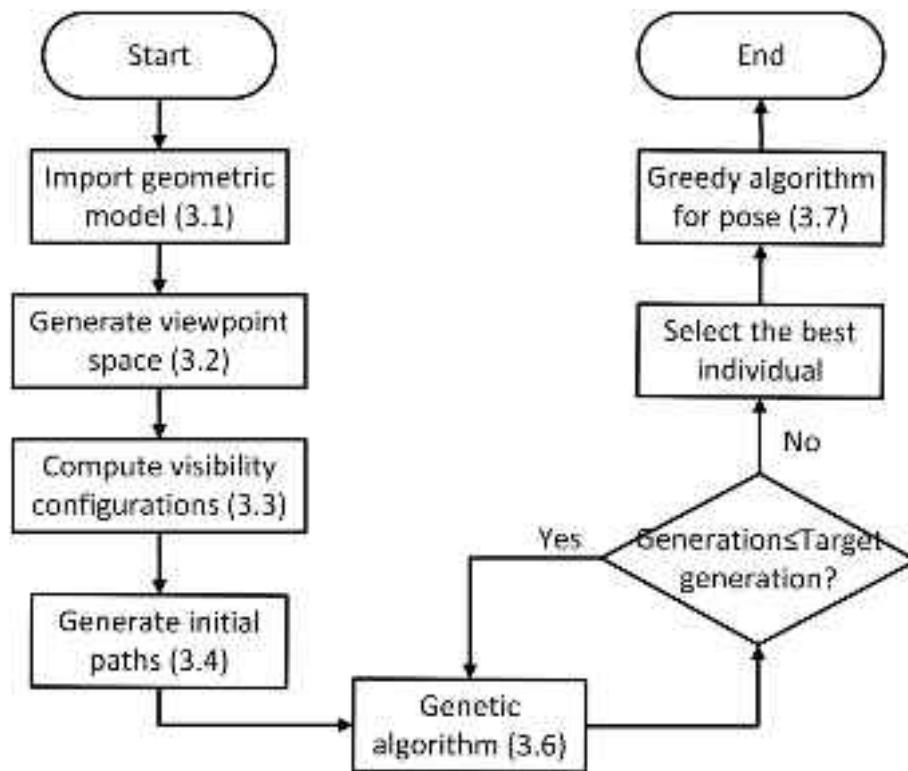


Fig. 1. Flowchart of the path planning method (the numbers refer to the corresponding section).

proposed a variant of TSP with Neighborhoods (TSPN) for coverage path planning in urban environments with multiple buildings. Sharma et al. [58] applied the genetic algorithm to determine the optimal viewpoint positions and poses and the Fast Marching Method for planning a collision-free path through the viewpoints. Tong et al. [59] proposed a convex hull-based sampling method and applied a TSP solver with Rapid-exploring Random Tree (RRT) collision avoidance. For this category of methods, since efficiency is not considered when selecting the viewpoints, the planned mission can be suboptimal even if the TSP solution provides the shortest path traversing the viewpoints. Additionally, the viewpoint sampling process can be challenging for infrastructures with complex shapes.

Another idea is generating a 2D surface around the target and solving path planning problems within this surface. Yu et al. [60] applied ant colony optimization to achieve path planning for construction safety inspections at a certain altitude. Tan et al. [61] used A* algorithm and genetic algorithm to conduct path planning on a plane around the target building for an automatic UAV-based crack detection method. Bolourian et al. [19] also integrated the A* and genetic algorithms to conduct bridge defects inspection path planning on a pre-determined plane considering the criticality levels. Bircher et al. [62,63] employed an optimization method to sample the viewpoints, with the objective of minimizing the path's length connecting these viewpoints, calculated using a Lin-Kernighan Heuristic (LKH) solver. However, applying the LKH solver for every optimization iteration can be computationally burdensome if the search dimension is too large. Biundini et al. [64] used a data filtering algorithm to generate the outline surface of the infrastructure, and a coverage path planning problem was formulated and solved on this surface. Janousek et al. [65] generated a motion planning roadmap based on triangular meshes and used a self-organizing neural network to solve the coverage path planning problem. Tao et al. [66] combined genetic algorithm and particle swarm optimization (PSO) to conduct wall surface inspection in a GPS-denied environment. Ibrahim et al. [67] applied a genetic algorithm for 5-degree-of-freedom (5DOF) path planning (3D location and camera pose,

including pitch and yaw) in building construction monitoring, providing an example of task-specific path planning in construction engineering. However, the outline of the inspection object was relatively simple (the exterior surface of a building), and the generated viewpoint space was relatively constrained (less than 1000 points). As a result, both positions and poses could be discretized into a few levels for simplicity of optimization, while the 5DOF optimization in a large-scale complex space that requires a fine viewpoint grid can be challenging to converge. In addition, coverage was regarded as an objective instead of a constraint. Thus, there was no guarantee for minimum coverage to ensure the comprehensiveness of the inspection.

For surface-based methods mentioned above, the distance from the surface to the drone is usually predetermined, which vastly reduces the flexibility of path planning and makes the generated paths less optimal. Additionally, generating a surface can be difficult for complex structures. There have been limited explorations in infrastructure mission planning in 3D space instead of the 2D surface around the target. Mansouri et al. [68] proposed a 3D mission planning method by slicing the 3D space into horizontal planes and leveraging convex hull to generate paths for multi-UAS cooperation. Hepp et al. [69] constructed the viewpoint space as a volumetric occupancy graph with observation quality measurements, which were calculated by a ray-casting method. An RRT-based trajectory optimization method was then applied for 3D mission planning, considering 3D reconstruction quality. Roberts et al. [70] leveraged submodularity to first find a path that maximizes the coverage and 3D reconstruction quality with battery constraint and then calculate the viewpoint orientations, achieving effective path planning in 3D space, which delivered high reconstruction quality. Ivić et al. [71] applied a Heat Equation Driven Area Coverage (HEDAC) algorithm for multi-UAV corporation in 3D visual inspection. Bircher et al. [72] proposed an online path planning method based on receding horizon strategy. Papachristos et al. [73] also applied receding horizon fashion in their path planning strategy, while uncertainty-optimization is taken in the method. Song et al. [74] addressed online inspection problem with a global-local strategy, which improved the completeness of

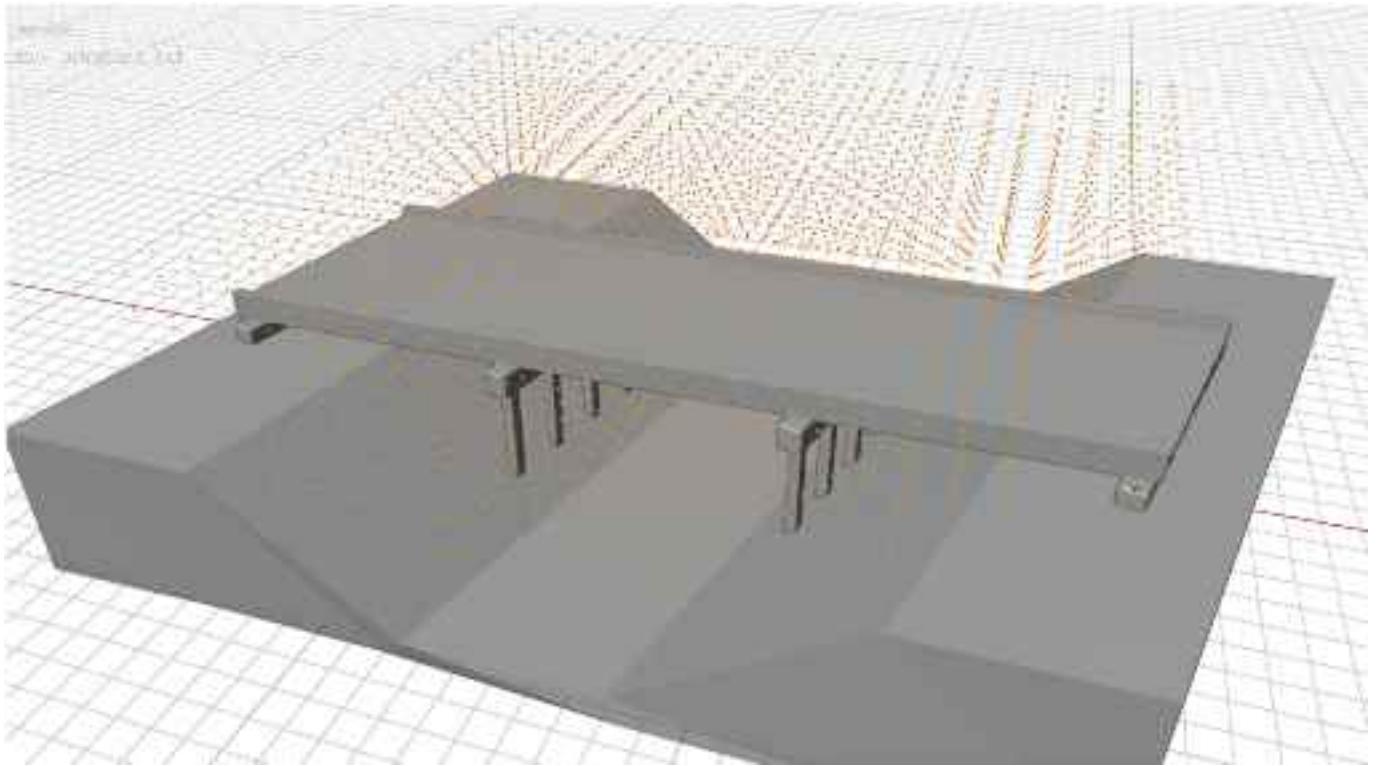


Fig. 2. Example of viewpoint space consisting of a regular grid with 1 m spacing.

surface coverage.

Expanding on the literature above, our study focuses on formulating SIPP as an optimization problem and proposing a comprehensive framework for SIPP that works for diverse types of infrastructures and tasks with high flexibility. To ensure inspection reliability, we regard coverage as a constraint that must exceed a high threshold. When calculating visibility, we applied visible distance and inclination angle constraints to ensure inspection quality. To enable optimization convergence for complex geometries with large viewpoint grids and variable spaces, we propose a two-stage optimization approach consisting of a genetic algorithm for determining viewpoint coordinates in 3D space around the target and a greedy algorithm for determining poses. This scheme effectively addresses the convergence challenges posed by large variable spaces by breaking the unknowns into two separate and smaller optimization steps. The proposed method allows UAS-based inspection to optimize efficiency while ensuring inspection safety and quality. In short, the primary contributions of this work include:

- 1) Proposing a two-stage optimization formulation that delivers desired coverage for computationally efficient structural inspection path planning.
- 2) Revealing the role and interaction of optimization parameters via a comprehensive sensitivity analysis to verify the applicability of the proposed method.
- 3) Demonstrating the flexibility of the proposed method for specialized inspection requirements such as focused inspection and no-fly zones.

3. Methodology

In the proposed path planning method, the input is a digital model from various sources, e.g., Computer-Aided Design (CAD) or a scanned 3D model generated using LiDAR or photogrammetry. Based on the model, the spatial configuration, including the viewpoint grid and visibility information, was first generated to define the planning problem.

Then, a two-step optimization process is performed. First, a genetic optimization algorithm is applied to identify the locations of points on the optimal path with the desired coverage and minimum length. The fundamental assumption at this first step is the presence of an omnidirectional camera at each point, leaving the camera pose estimation to the next step. Subsequently, a greedy algorithm is used to determine the camera poses of these viewpoints, given the angular specifications of the UAS camera. In the second stage, one viewpoint can have multiple viewing angles to satisfy the coverage requirement, especially for the corners or tight spaces. This formulation effectively splits the optimization problem into two smaller sub-problems. The process of the proposed path planning method is summarized in Fig. 1. In this process, after importing the geometric model, all the steps can be achieved automatically, except for initializing the paths, which can be semi-automated if rule-based path initialization is applied.

3.1. Modeling

The proposed framework is generic and compatible with various models, including CAD models, point clouds, and reconstructed meshes. The input model will be converted into a regular mesh for subsequent processing. The model is then divided into two parts: the inspection object and its surrounding environment (e.g., ground, trees, other obstacles, etc.). For visibility calculations, we exclusively use the mesh of the inspection object, while the surrounding environment is considered when determining the passable area of the UAS.

3.2. Grid-based graph configuration

To start the process, the bounding box accommodating all the faces of the model is automatically detected, and a predetermined clearance distance is added in each direction, denoting the region around the object available for inspection. The viewpoint space is then discretized with a 3D grid, on which candidate viewpoints can be generated. Next, we filter out the viewpoints inside the model (inspection object) or those

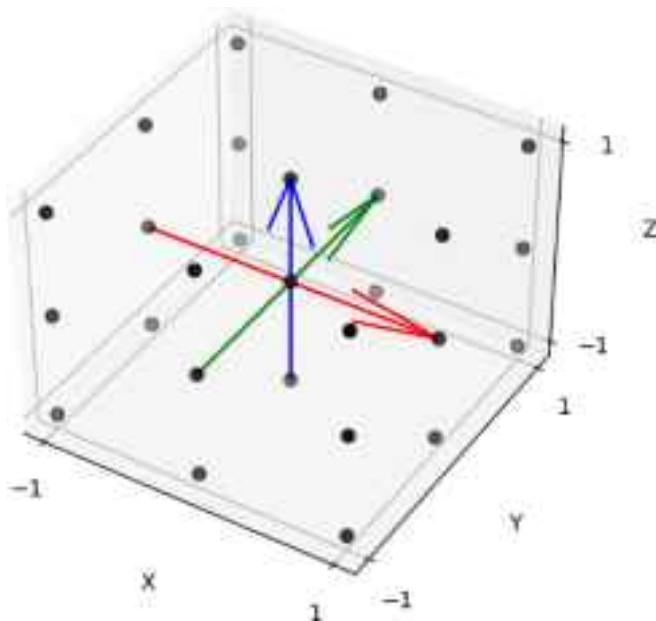


Fig. 3. 26-connectivity graph for the vertex at the center.

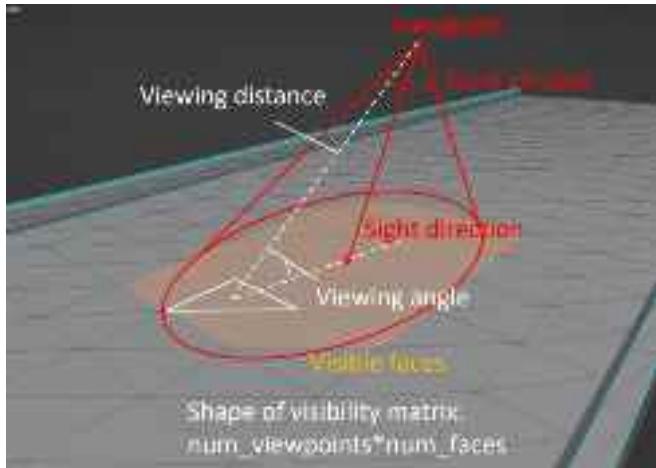


Fig. 4. Visibility of a face from a viewpoint.

with a distance to the model less than the safety threshold, addressing safety concerns. The viewpoint space is defined as a uniform rectilinear grid, and an example of the generated viewpoint space is shown in Fig. 2. Based on the grid, a 26-connectivity 3D graph can be generated. For each vertex, there are at most 26 edges connecting it with its neighbor points, which have a distance no more than the grid size on all three axes, as shown in Fig. 3. Among the 26 edges, six connect to the face midpoints (with a length of 1), 12 connect to the cube edge midpoints (with a length of $\sqrt{2}$ grid size), and eight connect to the corners (with a length of $\sqrt{3}$ grid size). For each edge, an intersection check with each face will be performed, and the edges intersecting with any face will be regarded as invalid to prevent collision. The movement of the UAS will be limited in the viewpoint space and along the valid edges, which ensures that the path will be continuous without the UAS crashing into obstacles.

3.3. Visibility configurations

To ascertain the coverage of a path, it is essential to establish the visibility of the geometric model from the 3D viewpoint space through

geometric relationships. As shown in Fig. 4, a spherical cone is defined with the sight direction as its axis, the visible distance as its radius, and the FOV as its angle, representing the space within the line of sight. All the faces inside this spherical cone that are not blocked by other faces are considered visible. The process of determining visibility is explained in Algorithm 1. Firstly, the line segment connecting each viewpoint and the center of every face is generated, and the inclination angle between the line segment and the face is calculated. Subsequently, the presence of any intervening face between each viewpoint and the respective face is detected. By considering both the obstacle face and inclination angle, visibility information can be precomputed and stored in memory. This enables direct access to visibility during optimization instead of recalculating it in each optimization epoch. The coverage of a path can then be easily determined based on the visibility of each face.

Algorithm 1: Calculate visibility matrix

```

Input:  $F\{f_i\}$ : the collection of faces forming the bridge model
 $Q\{q_j\}$ : the collection of all viewpoints
 $\delta$ : distance threshold to determine visibility
 $\alpha$ : angle threshold to determine visibility
Output:  $V\{v_{ij}\}$ : binary visibility matrix indicating the visibility of each face from each viewpoint
1. for  $q_j$  in  $Q\{q_j\}$  do
    // loop for each viewpoint in the viewpoint space
    2.   for  $f_i$  in  $F\{f_i\}$  do
        // loop for each face
        3.        $s_{ij} \leftarrow \text{segment}(q_j, f_i)$ 
        4.        $a_{ij} \leftarrow \text{angle}(s_{ij}, f_i) > \alpha$  and  $d_{ij} \leq \delta$  and  $s_{ij} \cap F \neq \emptyset$ 
        5.        $V_{ij} \leftarrow s_{ij} \cap F$ 
        // determine visibility based on angle and distance
    
```

3.4. Path initialization

For initialization, the path can be created by connecting a series of n consecutive (neighbor) viewpoints in the 3D space following the 26-connectivity graph shown in Fig. 2, where n denotes the number of viewpoints. Therefore, the positions of the viewpoints on the path can be encoded as an array with the shape of $(n, 3)$, representing the path to be optimized. In this study, two types of initializations are defined: random initialization and rule-based initialization. Random initialization can be fully automatic, while rule-based initialization needs manually segmenting the spans of the bridge, after which the rule-based initial paths can be generated.

3.4.1. Random initialization

For random initialization, the choice of the viewpoints on the path is determined by randomly selecting an edge connecting to the current viewpoint, and the path is formed by iteratively moving forward until its length reaches the desired value. It is worth noting that the choice of points is restricted to the viewpoint space, eliminating the possibility of the UAS colliding with obstacles. A reasonable initialization path will contribute to a faster convergence. At the same time, the optimization result is not overly sensitive to the initialization path length because the optimization process iteratively adjusts the path length to achieve the desired inspection outcome.

3.4.2. Rule-based initialization

Inspectors typically follow a rule-based progression of inspection tasks in manually controlled structural inspection tasks. For instance, they may fly the UAS along a grid path traversing every bridge span. While rule-based paths may not always be mathematically optimal, they are often reasonable starting points and can yield relatively satisfactory results. Moreover, once the rule is established, the paths can be automatically generated. It is hypothesized that applying rule-based initialization helps develop more realistic initial paths and improves the convergence speed. On the other hand, although most random initialized paths have poor performance compared to rule-based initialization,

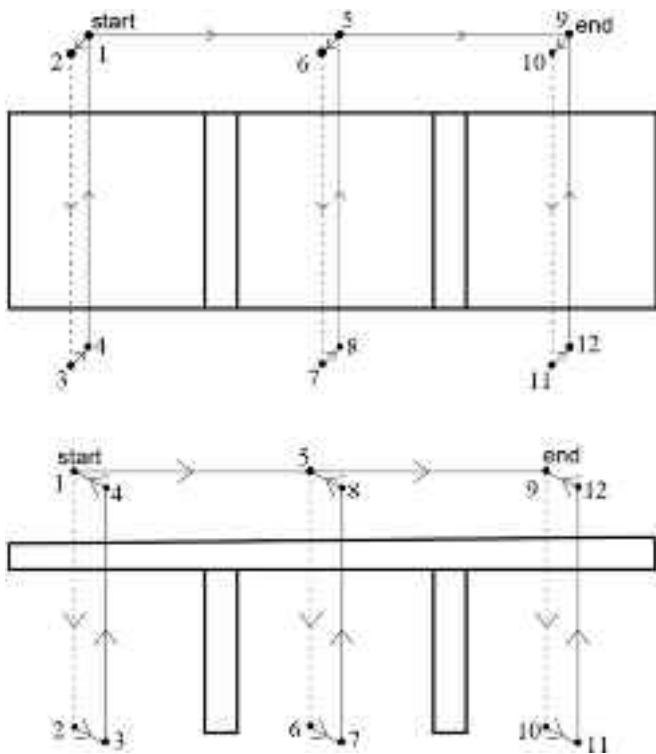


Fig. 5. Rule-based initialization demonstration.

their higher flexibility contributes to inspecting difficult-to-reach areas with rule-based initialized paths. Therefore, combining these two types of initializations is expected to help balance exploration and exploitation in the optimization process.

In this study, the impact of such a rule-based initialization is studied by generating a portion of the initial paths using a rule-based strategy. The space around the bridge is divided into three spans along the bridge length, and each span is divided into four regions on either side of the bridge and above and below the deck, resulting in a total of 12 sections. To create an initial rule, one point is selected in each of the 12 sections, which are then connected to each other consecutively to form three connected loops around the bridge. Fig. 5 provides a simple demonstration of an example path with one loop around each span to demonstrate the rule-based initialization. It needs to be noted that Fig. 5 is simplified for understandability. In practice, the points are randomly selected in each region, and a collision check is performed to ensure the path won't collide with the bridge. After inputting the height of bridge deck and locations of piers, the rule-based initialization can be performed automatically.

3.5. Optimization problem formulation

3.5.1. Optimization variables

The optimization variables are defined by the coordinates of the waypoints on the path in the 3D space, denoted by $P\{p_i \in \mathbb{Z}^3, i \in [0, n]\}$, in which each of the two adjacent points is connected, as shown in Fig. 3. Therefore, the search space dimension is d^n where d is the connectivity number (26 in this paper) and n is the number of waypoints on the path.

3.5.2. Objective function

The objective function is minimizing the path length, which is denoted by Eq. (1)

$$l^P = \sum_{p_i \in P} \|p_{i+1} - p_i\|_2 \quad (1)$$

In Eq. (1), l^P denotes the length of path P , p_i represents the waypoints

on the path P .

3.5.3. Constraint

With the goal of ensuring the reliability of the inspection, the constraint is defined by the coverage of the path, which can be denoted by Eq. (2).

$$c^P = \frac{\sum_{f_j \in F} \left[\left(\sum_{p_i \in P} v_{ij} \right) \geq w_j \right] \times a_j}{\sum_{f_j \in F} a_j} \geq c^t \quad (2)$$

In Eq. (2), c^P represents the coverage of path P , v_{ij} represents the visibility of face j from point i on the path, which is determined through Algorithm 1. w_j represents the importance weight of face j , which is always 1 for uniform weights. The effect of importance weight is discussed in 5.10. a_j represents the area of face j . c^t is the coverage requirement, $[]$ is the Iverson bracket, which returns 1 if the statement inside is true and returns 0 otherwise.

3.6. Genetic optimization algorithm

To produce optimal paths capable of achieving the desired coverage while maintaining high efficiency, this study employs a genetic optimization algorithm (GA). This iterative approach is inspired by the process of natural selection and evolution and aims to progressively enhance the quality of the solution (in our case, the performance of the inspection path). In GA, generation refers to specific iterations or steps in the evolution process. In each generation, the collections of waypoints on a path are encoded as chromosomes, and offspring are generated by applying mutation and crossover operators on the parents, which will be introduced in detail later. Individuals with better fitness have a higher probability of surviving and passing their features to their offspring so that their performance can be iteratively improved.

3.6.1. Overall optimization process

Fig. 6 shows the process of the optimization algorithm applied to solve the path planning problem. Firstly, paths are initialized based on the two strategies discussed in 3.4. A selection operator is then employed to identify offspring exhibiting robust performance as parents for crossover and mutation operators. Performance in this context is characterized by a measure referred to as fitness. These operators are applied to generate offspring with new and improved attributes, offering the potential to surpass their parental counterparts. In each generation, the coordinated efforts of the mutation, crossover, and selection operators drive the iterative enhancement of individuals, thereby refining the coverage and efficiency of paths. The specifics of these operators are detailed in the next section.

3.6.2. Fitness

In GA, fitness refers to a measure of how optimal a solution is or how well an individual performs with respect to the problem objectives. In general, we aim to achieve the shortest inspection path that produces visual coverage of the target structure. As we formulate the problem with coverage as a constraint, if the coverage of a path does not meet the constraint, improving coverage will be the primary objective. Otherwise, path length becomes the fitness value to be minimized. We define fitness as Eq. (1), where α is an extremely large value compared to path length. It is worth noting that fitness is always negative so that it can be maximized while path length is minimized.

$$\phi^P = \begin{cases} -\frac{\epsilon}{c^P} & \text{if } c^P < c^t \\ -l^P & \text{if } c^P \geq c^t \end{cases} \quad (3)$$

In Eq. (3), ϕ^P represents the fitness of path P , c^P is the coverage calculated by Eq. (2), c^t is the coverage requirement, l^P is the path length

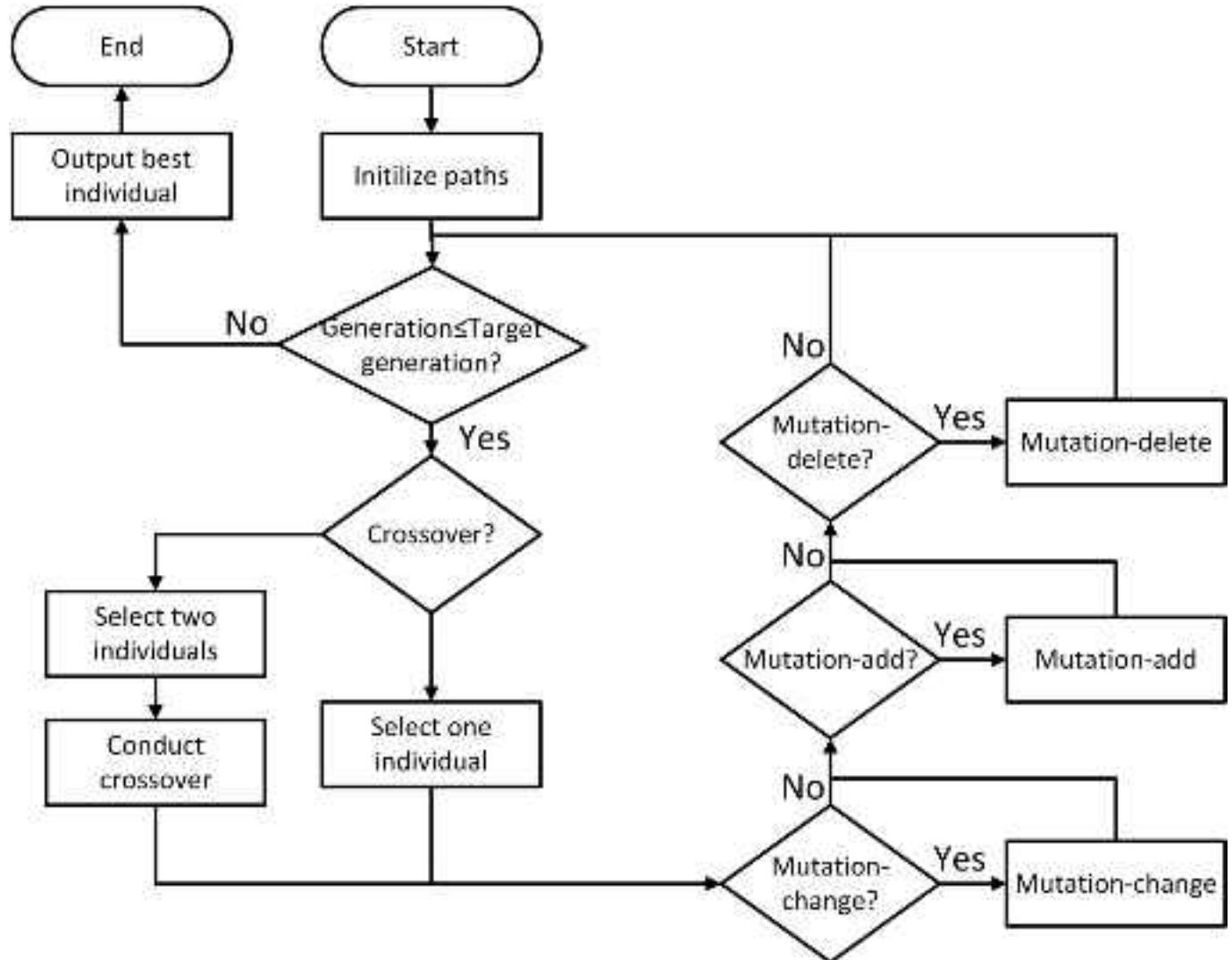


Fig. 6. Flowchart of the genetic optimization algorithm.

calculated by Eq. (1), and ϵ is an extremely large value compared to the path length. Given that ϵ is large enough, its value does not affect the optimization procedure.

3.6.3. Selection operator

The selection operator determines which individuals from the current population will undergo further genetic operations. In this study, a tournament selector is used, wherein for each selection, a fixed number of individuals will be randomly selected from the parental generation to form a tournament, and the one with the highest fitness will be selected for the next generation. This method helps maintain diversity in the population while ensuring that individuals with better fitness have a higher chance of survival. The number of individuals selected for the tournament, which is denoted as tournament size, is the parameter that controls the balance between exploration and exploitation. If tournament size equals population size, the best population will always be selected, eliminating diversity. At the opposite extreme, if the tournament size equals 1, the selection is totally random, thus curbing evolution and performance improvement. Therefore, a reasonable

tournament size should be selected for better optimization convergence.

3.6.4. Crossover operator

Fig. 7(a) demonstrates the crossover operator. When performing a crossover operator, two paths are initially selected with the selection operator. We define two points as neighboring points if they are connected on the 26-connectivity graph, as shown in Fig. 2, or have the same coordinates. With this definition, all pairs of neighboring points are computed, for which the two points are on different paths. Subsequently, we assume the UAS starts from one path, and on each pair of neighboring points, it will have the probability of switching to the other path. This will enable random swaps to occur between two paths to generate offspring inheriting the advantages of both parents.

3.6.5. Mutation operators

Fig. 7(b) demonstrates three types of mutation operators, namely the change, add, and delete operators. The change operator alters the coordinates of viewpoints. When applied to a viewpoint, the viewpoints before and after on the path will be selected first. Then, it calculates the

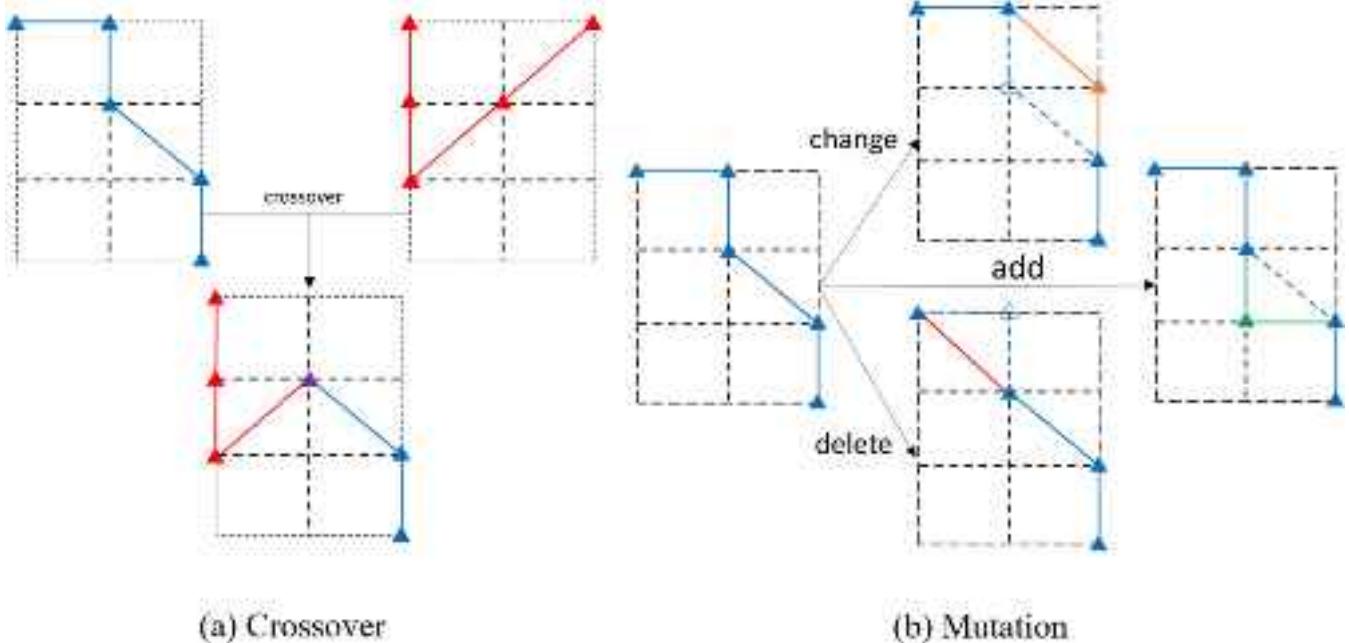


Fig. 7. Demonstration of crossover and mutation operators. Note that crossover and mutation happen in 3D, and the 2D examples are provided here for simplicity.

intersection of the two sets of neighbor viewpoints for these two points and randomly selects a new viewpoint from the intersection set to replace the original viewpoint. This ensures that the mutated path remains continuous. The add operator is similar to the change operator, with the key difference being that it examines the neighboring viewpoints of the current and next viewpoints. It randomly selects a point from the intersection set and inserts it after the current viewpoint. The delete operator creates new paths by removing points. Before deleting a viewpoint, it is necessary to verify whether the last viewpoint is among the neighboring viewpoints set of the following viewpoint to ensure continuity. If not, the delete operator will not be applied, ensuring continuity.

A parameter, denoted as the individual evolution rate (IER), signifies the probability of initiating a mutation or crossover event for a path. If a crossover event is triggered, two individuals are selected from the parental generation, upon which the crossover operator is applied. Otherwise, the child will inherit attributes from only one individual from the parental generation. Then, a similar decision-making process occurs for the three mutation operators successively to determine if mutation will take place. For each crossover or mutation operator applied to an individual, a gene evolution rate (GER) is employed to determine the probability of applying the operator to each viewpoint for the mutation operator or pairs of neighbor viewpoints for the crossover operator.

3.7. Greedy algorithm for pose calculation

When determining the optimal path using GA, it is assumed that the UAS has a complete view of its surroundings, and the pose and FOV of the camera are not considered. Based on the viewpoint positions generated by the optimization, a series of pose vectors can be generated, pointing from each viewpoint on the path to each face. We then use a greedy algorithm to select the pose vectors to enable the poses to meet the coverage constraint, as explained in Algorithm 2. In essence, this algorithm iteratively selects the pose that covers the largest number of unseen faces until all the faces visible from the path are covered.

Algorithm 2: Determine the poses with greedy algorithm

Input: $F\{f_i\}$: the collection of faces (coming the bridge mode);
 $P\{p_i\}$: the collection of viewpoints on the path;
 $V\{v_{ij}\}$: visibility matrix;
 β : field of view angle.

Output: S : the collection of poses on the path.

```

1 for  $p_i$  in  $P$  do
    // loop for each viewpoint on the path
2     for  $f_j$  in  $F$  do
        // loop for each target face
3          $s_{ij} \leftarrow \text{segment}(p_i, f_j)$ 
            // the sight from viewpoint to target face
4         for  $f_k$  in  $F$  do
            // loop for each face to determine visibility
5              $s_{jk} \leftarrow \text{segment}(p_i, f_k)$ 
                // the segment connecting such viewpoint and face to
                // determine visibility
6              $v_{jk} \leftarrow v_{jk} + 1$  and  $\text{angle}(s_{ij}, s_{jk}) < \beta$ 
                //  $v^*$  is used to record if  $f_k$  is visible from  $p_i$  with
                // sight.
7          $F\{f_k\} \leftarrow F - n \leftarrow \text{length}(F^1), S \leftarrow \emptyset$ 
            // initialization for the invisible faces list
8     while  $n \neq 0$  do
9          $i, j = \arg \max \frac{\sum v_{jk} \text{ and } f_k}{\sum v_{jk}}$ 
            // pick the pose with maximum coverage on remaining faces
10         $F^2 \leftarrow F^1 - \{f_k | v_{jk} = 1\}$ 
            // delete the faces that are visible with the pose from the
            // invisible faces collection
11         $S \leftarrow S \cup s_{ij}$ 
            // record the poses with maximum coverage
12         $n = \text{length}(F^2)$ 
13    next  $S$  so that the viewpoints position have the same sequence as  $P$ 

```

4. Experimental setup

This section describes the experiments designed and performed using the proposed method to evaluate its performance and efficiency. The GA was implemented based on the standard formulation [75] using the Distributed Evolutionary Algorithms in Python (DEAP) framework [76].

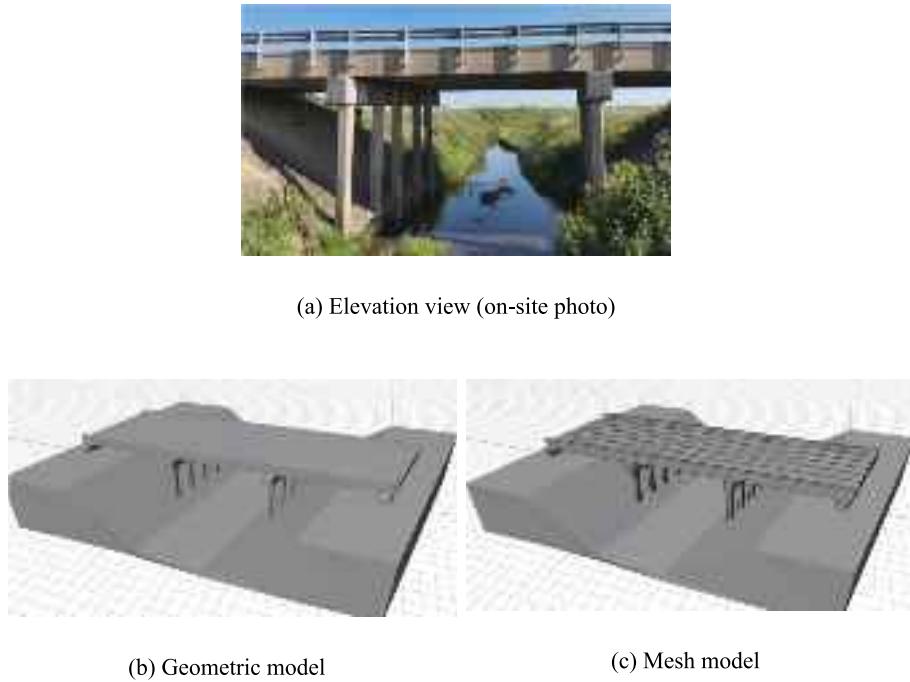


Fig. 8. Bridge model used in this study.

Table 1
Parameters for the baseline case.

IER	GER	Tournament size	Rule-based initialization proportion
0.5 0.75 1	0.05 0.1 0.2 0.5	12 25 50	0 0.5 1
Population size	Generations	Coverage requirement	FOV
125 250 500	500	90 % 95 % 99 %	60° 90° 120°

4.1. Model for experiment

To demonstrate the proposed mission planning algorithm, we use the model for an existing bridge in Champaign County, IL. Fig. 8 shows the bridge, along with the geometric and mesh models. The visible parts of the bridge, including the piers, superstructure, and bridge deck, are regarded as inspection targets. This bridge is a three-span structure with a total length of 25.3 m, a height of 5.2 m, a width of 7.3 m, a skew of 15 degrees, and a structural system composed of reinforced concrete girders. The bridge carries two traffic lanes over a river with a vertical clearance of about 4 m. Fig. 8 also depicts the geometric model constructed for the bridge and the mesh model generated with 3369 triangular faces, each with an average size of 0.13m². The size of each face is small enough so that the error caused by the discretization can be ignored.

4.2. Experimental parameters

This section provides a list of parameters and their definitions used in our experiments. Orthogonal experiments were conducted to investigate the influence of these parameters. The following four parameters determine the setup of the problem and are fixed throughout all our experiments. Table 1 summarizes the values chosen for these parameters.

- (1) **Visible distance:** the distance threshold for determining if a face is visible from a viewpoint. This parameter is influenced by multiple factors, including inspection requirements, camera resolution, FOV, and lighting conditions. For instance, with a smaller desired Ground Sampling Distance (GSD), lower camera

resolution, larger FOV, and a poor lighting condition, the visible distance should be set to a smaller value. Only when the distance between a face and a viewpoint is smaller than this threshold will the face be considered visible. A larger visible distance reduces the complexity of the path planning problem and facilitates convergence. Therefore, strategies including using a camera with higher resolution and improving the lighting condition by introducing artificial illumination can be applied in practice. Visible distance was defined as 10 m for the base case, and its influence is investigated. Note that this doesn't mean the UAS will navigate at this distance, but that any viewpoints beyond this threshold will be considered unsuitable for inspection purposes.

- (2) **Safety distance:** the threshold restricting the minimum distance between a viewpoint and a face to avoid accidents. All the viewpoints that have a distance to a face smaller than this value will be filtered out from the viewpoint space. A reasonable safety distance is necessary to prevent crash accidents, considering the localization accuracy of the UAS, the modeling precision, and the regulatory tolerances required by the infrastructure owner or inspection safety managers. In this paper a safety distance of 0.5 m was used.
- (3) **Visible inclination angle:** the angle threshold for determining if a face is visible from a viewpoint. Only when the angle between a face and the line of sight from a viewpoint to the face is smaller than this threshold will the face be regarded visible. A visible inclination angle of 45° was used for the base case, and the influence of visible inclination angle is investigated.
- (4) **Grid interval:** the interval of the grid on which viewpoint space is generated. In this paper a grid interval of 1 m was selected for a balance between resolution and computational cost.

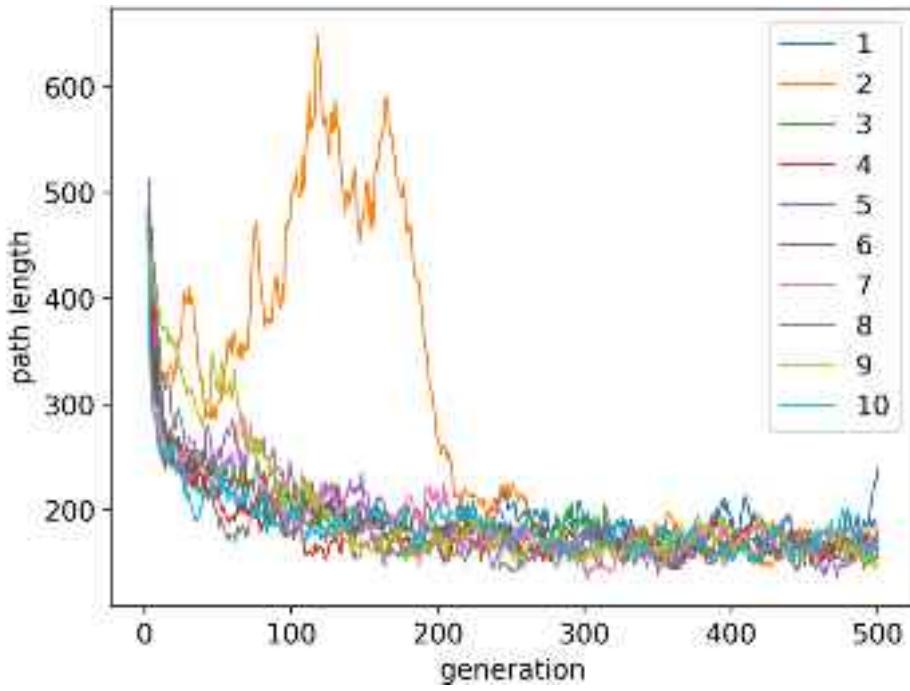


Fig. 9. Evolution for ten tests using base case parameters.

The following parameters affect the optimization algorithm and are varied across their respective practical ranges throughout our sensitivity analyses.

- (1) **Individual evolution rate (IER):** the probability of a child path going through mutation or crossover.
- (2) **Gene evolution rate (GER):** the probability for mutation to happen on each viewpoint, and for crossover between two pairs of neighbor points.
- (3) **Tournament size:** the number of individuals randomly selected for each tournament selection.
- (4) **Rule-based initialization proportion:** the proportion of initial paths generated by rule-based initialization. The remaining paths are generated following random initialization.
- (5) **Coverage goal:** the minimum coverage requirement for a path.
- (6) **FOV:** the field of view of the camera. This parameter only influences the results of the greedy algorithm.

4.3. Sensitivity analysis settings

We conducted a parametric analysis to research the influence of the parameters on the optimization results. Table 1 lists the values used for comparison in our sensitivity analysis, and the highlighted numbers are the parameters used for the baseline case. For each parameter setting, ten repetitions of the tests are conducted to quantify the impact of randomness in the optimization process.

5. Results and discussion

In this section, the results of feasibility analysis with the baseline case are firstly presented and compared with a direct 5DOF method. Sensitivity to the parameters mentioned in Table 1 is then analyzed, and examples of additional functionalities are provided for practical applications. The proposed method is also tested on different models to verify its applicability. Nearly all the experiments in this section achieved the required coverage within the first few generations, and the curves in this section showing the evolution of path length (e.g., Figs. 9, 11–13, 15–17, 20) start at the generation at which the coverage requirement is met.

5.1. Feasibility analysis

As stated before, using random initial paths and random operators such as mutation, crossover, and selection results in variations in the optimized response. We repeat each test ten times to demonstrate and quantify this randomness. The evolution of the ten experiments for the baseline case is shown in Fig. 9. The experiments successfully achieved the desired 95 % coverage goal after 3.3 generations on average, and path length consistently decreases with generation and eventually converges to a plateau. The average optimal path length is 162.2 m, and the standard deviation is 5.8 m, indicating that the optimization algorithm is robust despite its random nature. An interesting fact is that, for the second test, detrimental mutation or crossover happened before the 170th generation, leading to an increase in path length. However, the optimization gets back on track at around the 200th generation and reaches a level comparable to other tests, which indicates that the genetic algorithm successfully corrected the harmful mutations. These results reveal that the optimization algorithm demonstrates reliable convergence to a steady level.

Fig. 10 shows one of the optimized paths and the corresponding visible surfaces. The optimized paths tend to wind around the entire model, take pictures from different directions, and avoid repetitions to ensure efficiency, which intuitively demonstrates the overall success of the proposed SIPP method.

Fig. 11 provides a more exhaustive analysis of the efficiency of the optimized path after the first stage of determining viewpoint positions. Fig. 11(a) shows the increasing trend of the cumulative coverage with the addition of new viewpoints on the path during the inspection process, which has a steady upward trend. This shows that most of the viewpoints on the inspection path effectively contribute to generating visual coverage of the bridge. Fig. 11(b) shows the total coverage of each viewpoint (independent of other viewpoints) and the exclusive contribution of each viewpoint to coverage (the faces that are already visible from previous viewpoints are not counted). It can be seen that most viewpoints are uniquely contributing to coverage, while a few viewpoints provide duplicate coverage. The reason is that the path needs to be continuous, resulting in some inefficient viewpoints that primarily serve as the connection between other points. However, these

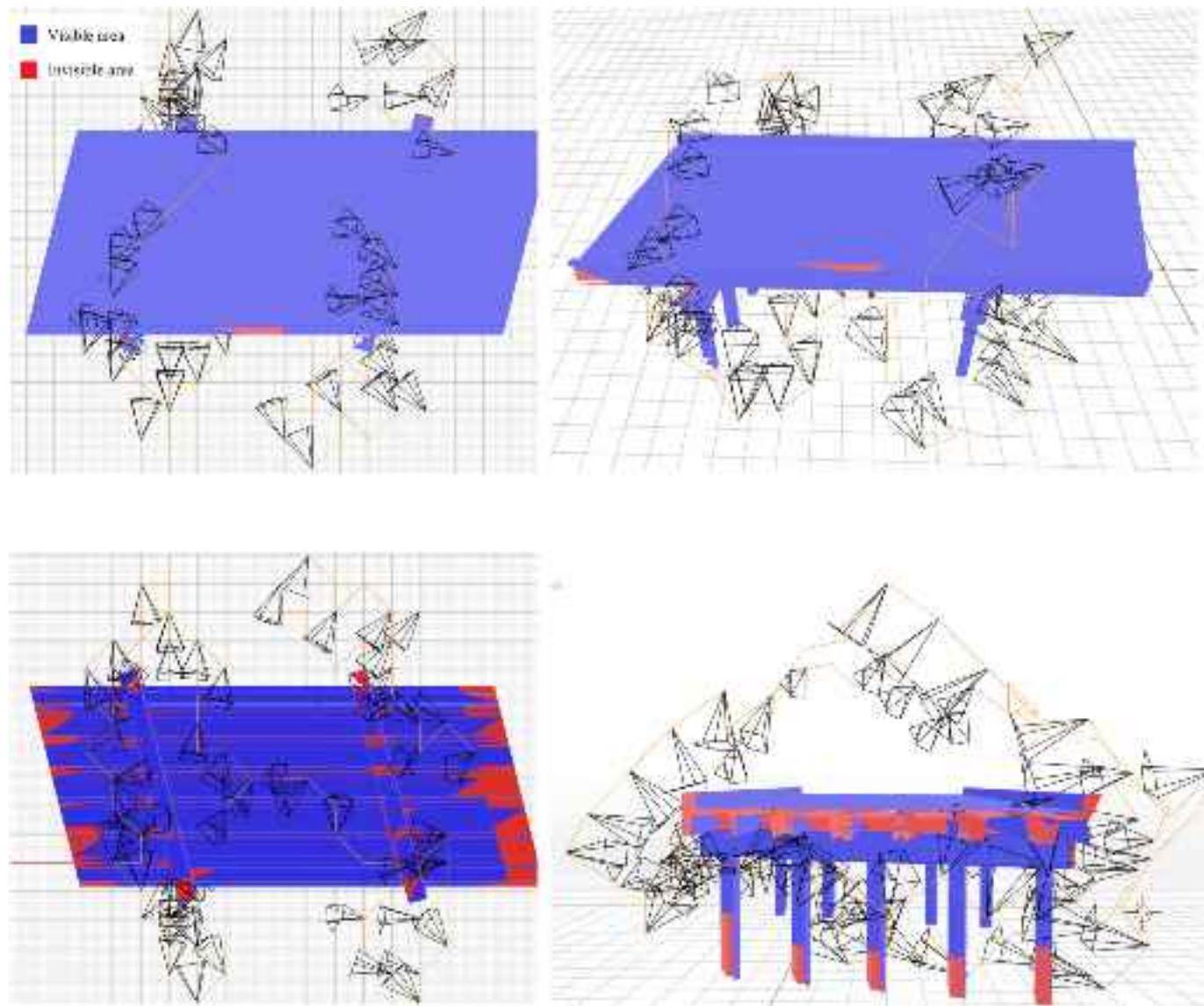


Fig. 10. Generated optimal path and visible surface (95 % coverage) viewed from different directions.

viewpoints can still be helpful when redundant viewpoints are needed for 3D reconstruction purposes.

5.2. Comparison with direct 5DOF optimization

As stated in the introduction, the inspection path could be determined via a direct 5-variable optimization [67]. This is achieved by simultaneously assuming five degrees of freedom (3D location and camera pose, including pitch and yaw) for the UAS at each point and solving the optimization problem for all five variables at the same time. To demonstrate the advantage of the proposed method, experiments are conducted using both the proposed two-stage method and a direct 5DOF optimization for comparison. An additional rotation mutation operator was incorporated into the 5DOF optimization, which randomly rotates the pose of each viewpoint. All parameter settings were kept identical and consistent across both methods. To assess the effectiveness of these two methods, given different complexities, they are tested within two scenarios: the baseline with a fine grid interval of 1 m and a coarse grid with an interval of 2 m. There are 3490 and 593 viewpoints in the viewpoint space of fine and coarse grids, respectively. The evolutions of the average path lengths are shown in Fig. 12. Ten repeated experiments

were conducted for robustness and reliability.

Compared to the direct 5DOF optimization, the proposed method shows faster convergence and yields a lower optimal path length (30.6 % and 44.5 % smaller path length for the coarse and fine grids, respectively). This indicates the reduced computational expense and superior optimization results of the proposed method. The standard deviation of the proposed method is also lower, indicating its more stable performance. Notably, for both grids, the optimal path length of the proposed method stabilizes at a similar level after convergence, in contrast to the direct 5DOF optimization which performs worse on the fine grid. The reason is that a finer grid significantly increases the scale of the solution space, making the optimization problem excessively challenging for the direct 5DOF optimization. However, by reducing the degrees of freedom from 5 to 3, the proposed method maintains an optimal outcome despite the added complexity imposed by the finer grid. To conclude, the proposed method exhibits apparent advantages in computational efficiency and optimization performance, especially when the problem is more complex.

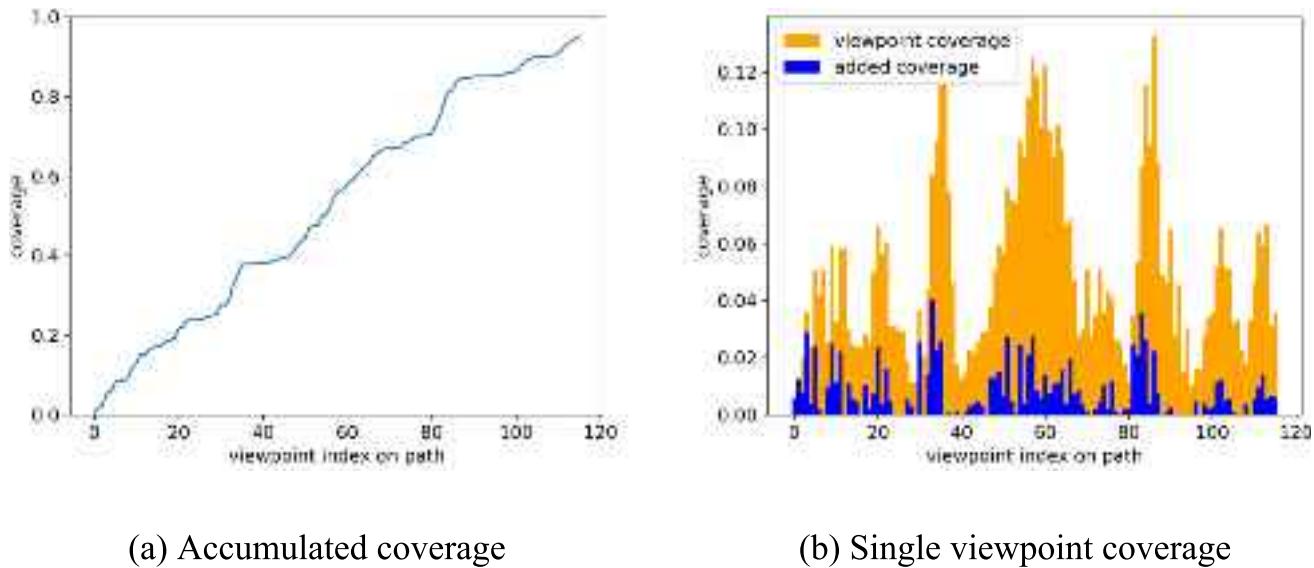


Fig. 11. Contribution of viewpoints on the optimal path to coverage.

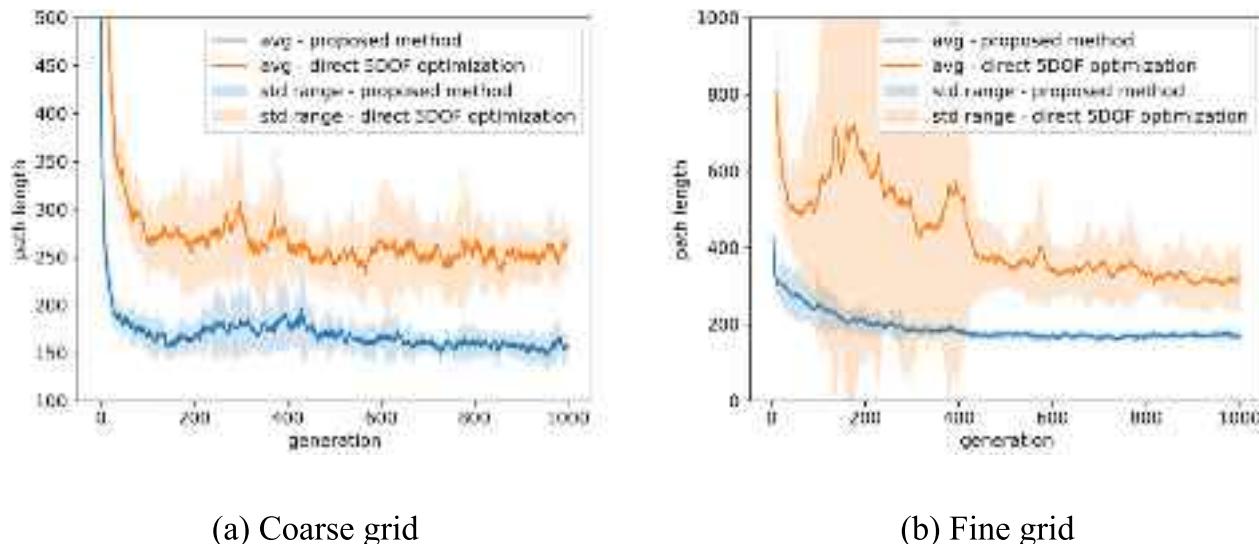


Fig. 12. Evolution of the proposed method and direct 5DOF optimization on different grids.

5.3. Influence of population and generation

Fig. 13 illustrates the evolutionary process for various population sizes. Across all three population settings, path length stabilizes after approximately 300 generations, indicating that a maximum generation limit of 500 is sufficient for optimization convergence. A comparison between population settings of 125 and 250 reveals that an inadequate population size results in suboptimal optimization outputs. Moreover, while a larger population of 500 initially leads to faster convergence than 250, both eventually reach a similar level after 500 generations. Considering the additional computational expense of higher population size and maximum generation settings, careful selection of parameters is crucial to achieve a balance between performance and computational cost.

5.4. Influence of evolution rates

We conducted experiments for different combinations of IER and GER, and the average and standard deviation of optimal path lengths are

summarized in Fig. 14. It is shown in Fig. 14 that both IER and GER have a significant influence on optimization results. It can be observed that an intermediate value for the two evolution rates leads to better paths, and the extreme values yield worse results. The reason is that the evolution rate controls the balance between exploration and exploitation. A higher evolution rate leads to more exploration of the solution space, meaning there is a greater chance of introducing random changes to the individuals, potentially enabling the algorithm to discover new, unexplored regions. However, excessive exploration may hinder the algorithm's ability to exploit the promising solutions it has already found. Additionally, with a low GER, a higher IER will lead to better results, indicating that the GER and IER tend to compensate for each other. Although extreme values are undesirable, the optimization results demonstrate robustness within a specific range of evolution rates. This suggests that, despite the influence of evolution rates on the results, the optimization process is not overly sensitive to these parameters.

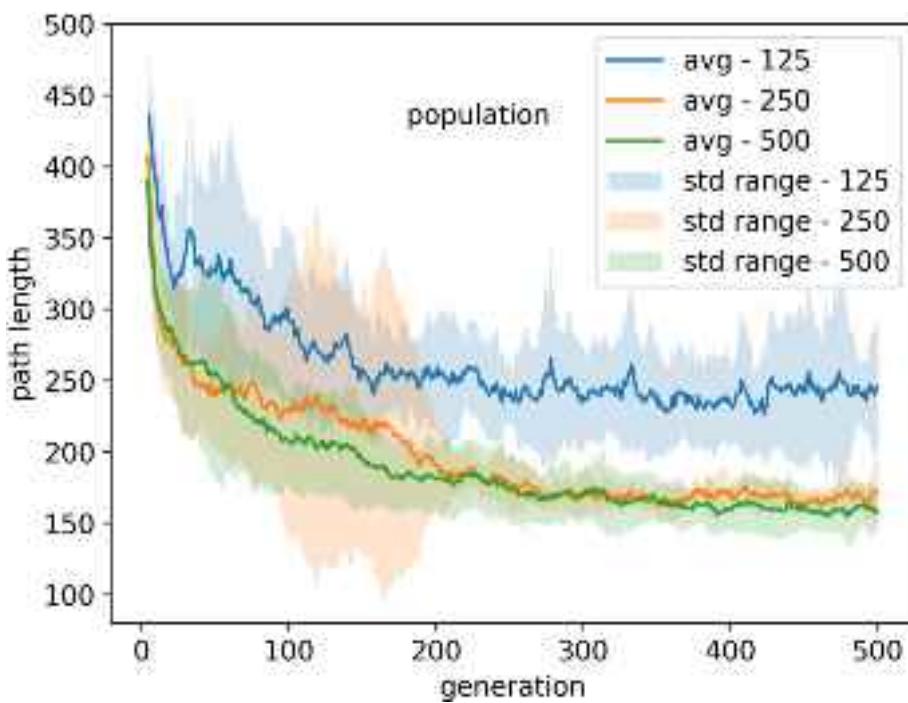


Fig. 13. Influence of population size and generation on the optimization process.

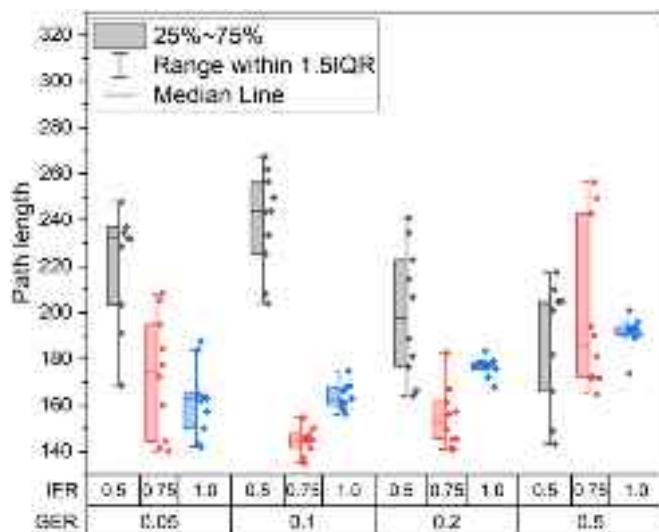


Fig. 14. Effect of evolution rates on path length.

5.5. Influence of rule-based initialization proportion

We conducted experiments for different rule-based initialization proportions, and the results are shown in Fig. 15. Fig. 15(a) shows that, compared with a completely random initialization strategy, using rule-based initialization significantly decreases the path length in the first 100 generations, which indicates that rule-based initialization helps to accelerate the convergence of the optimization. The coverage constraint is met after 11.9, 3.3, 1.5 generations on average for rule-based initialization proportion of 0, 0.5, 1, respectively. As shown in Fig. 15(b), rule-based initialization facilitates the convergence of the coverage constraint. The reason is that the random paths without rules move completely randomly at the beginning, requiring a longer path length and more generations to meet the coverage constraint. Although the difference decreases with evolution, combining rule-based initialization

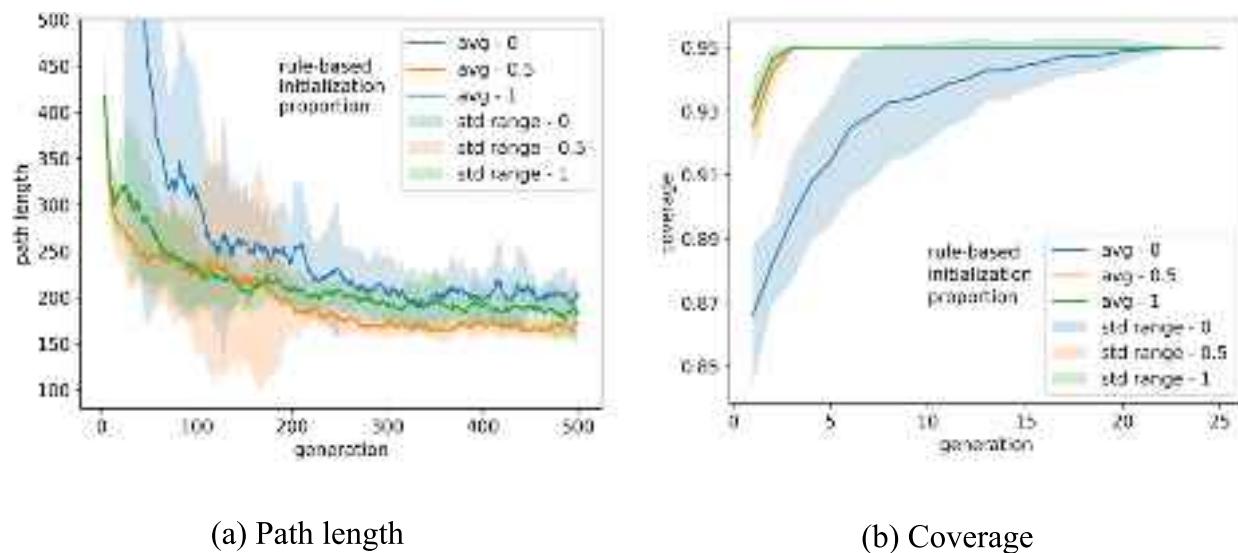
with random initialization still shows apparent advantages in the final optimal path length. When all the paths are initialized based on rules, there is no significant difference for the first 150 generations, while it eventually performs worse than the baseline case. The reason is that random initialization provides paths with more flexibility, which balances the drawback of rule-based initialization. A more advanced initialization could probably improve the result, while more manual design will be required by human inspectors.

5.6. Influence of tournament size

We also examined the impact of tournament size on optimization convergence, as presented in Fig. 16. Similar to the results observed for evolution rates, the shortest path length is achieved with a moderate tournament size of 25. Using a larger tournament size of 50 increases the path length by approximately 19 %, while a smaller tournament size of 12 results in about a 12 % increase in path length. Therefore, selecting an appropriate tournament size can enhance optimization results, although excessively large or small values can still produce acceptable outcomes.

5.7. Influence of inspection quality requirements

Visible distance and visible inclination angle play important roles in structural inspections by determining the visual inspection quality. To investigate the influence of inspection quality requirements on the optimization algorithm, we conducted a parametric study on these two parameters. As shown in Fig. 17(a), a smaller visible distance of 5 m leads to a slower convergence and a higher path length due to the reduced number of faces covered by each viewpoint. However, there is no significant difference between 10 m and 15 m, indicating that further loosening the distance requirement will not simplify the optimization problem. This is likely because the faces with a larger distance from the viewpoint are more likely to be obstructed by other faces, weakening the effect of increasing the visible distance. Fig. 17(b) demonstrates a similar trend for the visible inclination angle, while the optimization shows higher sensitivity compared to the visible distance. A looser (i.e., smaller) visible inclination angle simplifies the optimization problem,



(a) Path length

(b) Coverage

Fig. 15. Influence of rule-based initialization proportion on the optimization process.

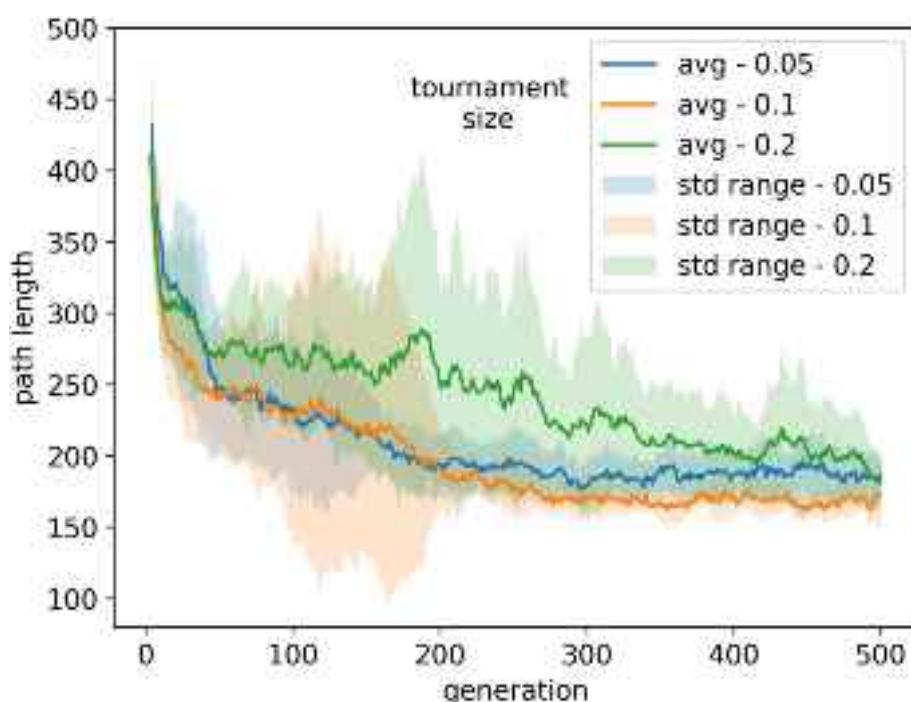


Fig. 16. Influence of tournament size on the optimization process.

whereas a stricter (i.e., larger) angle significantly increases the optimized path length and causes difficulty in convergence. Specifically, in Fig. 17(b), the inclination angle threshold of 55 results in convergence difficulties signified by the large fluctuations and the wide standard deviation band. This can be attributed to the corner faces that are difficult to access and need large inclination angles to cover. A comparison of corner face visibility between a looser and a stricter visible inclination angle is shown in Fig. 18. This figure shows that more corner faces are difficult to cover at the inclination angle threshold of 55° than 45°, mainly because of the geometric restrictions caused by the edges of the components that limit the views to sharp angles. This indicates that a stricter visible inclination angle makes it even harder to cover these faces, largely increasing the difficulty of solving the optimization

problem. Another reason is that the optimization parameters have been specifically selected for the base case. For different visible distances and inclination angles, further fine-tuning of optimization parameters is required for optimal performance.

5.8. Influence of FOV

In the proposed two-step formulation, FOV does not influence the GA optimization process (due to the omnidirectional camera assumption) and only affects the greedy pose determination step. We investigated the relationship between FOV and the number of poses required to fulfill the coverage constraint in the baseline case. As shown in Fig. 19, a larger FOV reduces the number of poses needed, suggesting that using a

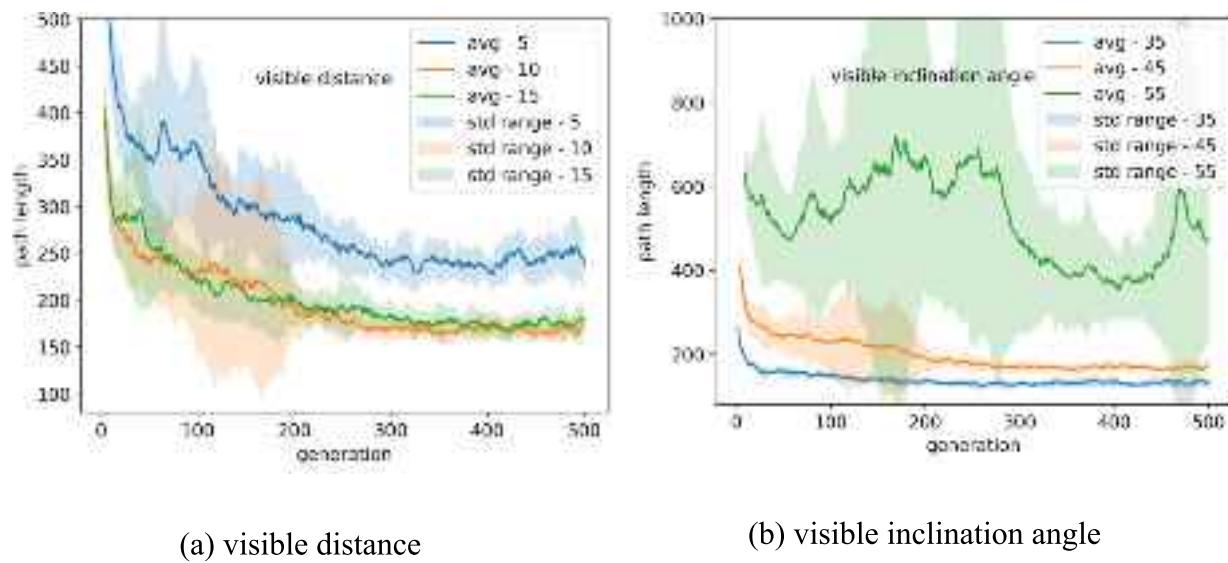


Fig. 17. Influence of inspection quality requirements on the optimization process.

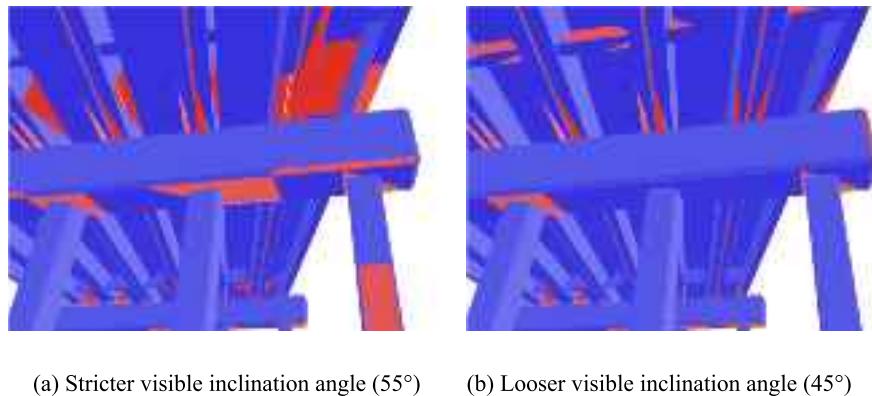


Fig. 18. Comparison in visible faces of paths with different visible inclination angles.

camera with a larger FOV can enhance inspection efficiency. This is because a larger FOV provides higher visibility from each viewpoint, similar to using a wide-angle camera. However, it is worth noting that, in practice, an excessively large FOV can lead to distortion and adversely impact inspection quality.

5.9. Influence of coverage constraint

The coverage constraint is a crucial factor in optimization. As illustrated in Fig. 20(a), when the coverage constraint is raised from 0.9 to 0.95, the path length increases gradually with minimal variation, signifying the method's reliability. With the stringent 0.99 coverage constraint, there is a substantial increase in both path length and standard deviation, indicating that higher coverage demands make convergence more challenging while all experiments still successfully converge. Fig. 20(b) demonstrates that a higher coverage requirement will take more optimization generations to meet, which is intuitive. It also shows that the constraint is met within 15 generations even for the high coverage requirement of 0.99, indicating the robustness of the proposed method.

5.10. Partial space inspection and no-fly zones

In real-world applications, various restrictions are often in place for safety and practical reasons. For instance, flying a UAS over a bridge

when traffic is not restricted poses a risk of driver distraction and accidents. In fact, flight above the deck may be unnecessary when the goal is the structural inspection of the substructure and superstructure, and the cracks on the roadway are not of immediate interest. Given the complexity of practical engineering scenarios, enabling partial space inspection is crucial for effective and practical structural inspection mission planning. Our proposed method conveniently accomplishes this by modifying the definition of the viewpoint configuration space and defining no-fly zones in areas of interest. The rule-based initialization is also altered accordingly to always select waypoints under the bridge. The results of restricting UAS flight above the bridge are illustrated in Fig. 21. In this figure, the UAS moves under the bridge and covers everywhere except the road above, thus avoiding driver distraction while inspecting the substructure and superstructure.

5.11. Focused inspection

Various parts of a structure may have different degrees of criticality to the overall performance and safety and may be subject to different levels of load-carrying demand and deterioration. It follows that effective inspections should take these disparities into account. As described earlier in this paper, the proposed framework allows for the assignment of varying degrees of importance to various parts of the structure by means of different weights for different structural components. To demonstrate, assume the case where the bottom of the girders at mid-

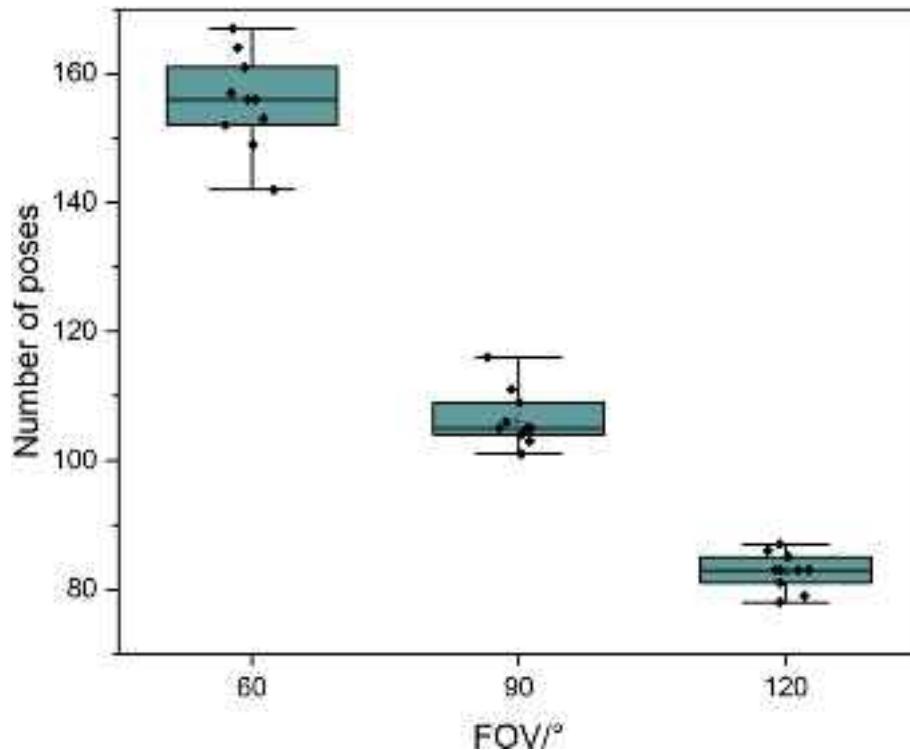


Fig. 19. Distribution of the number of poses needed for different FOV values.

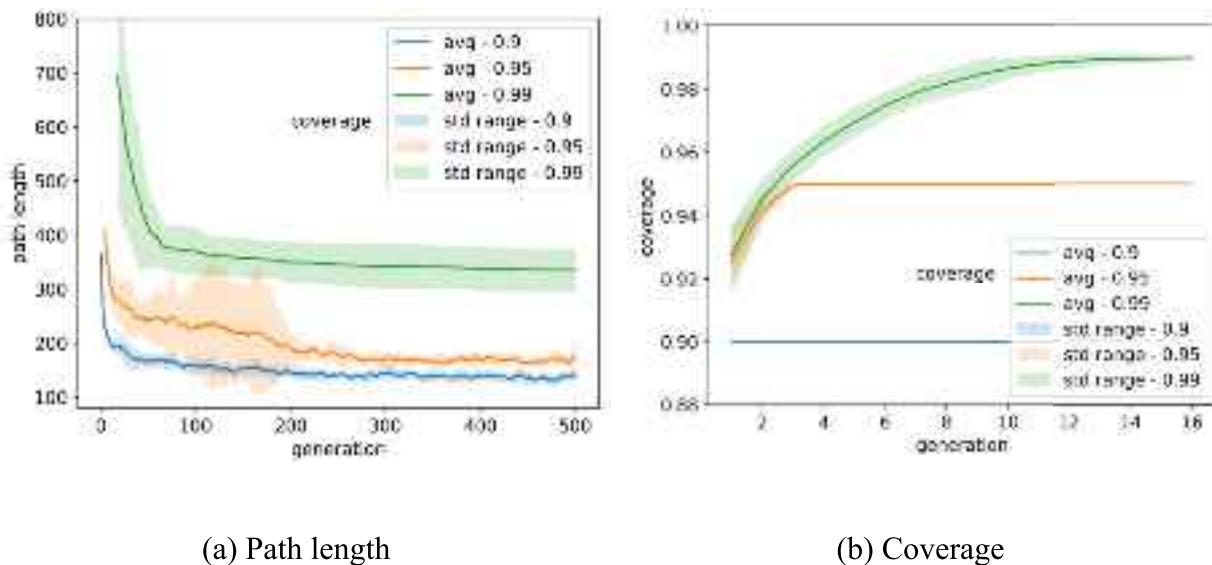


Fig. 20. Influence of coverage constraint on the optimization process.

span and the bridge deck joints above columns are identified as parts where damage tends to appear. To ensure greater attention to the more critical zones, a weight term (w_j) of 2 is assigned to them in Eq. (2) as opposed to a weight of 1 everywhere else, thus guaranteeing that these faces will be scanned at least twice as many times as the rest of the structure. The experimental settings and results are shown in Fig. 22. The critical zones are colored red in Fig. 22 (a) and (b). Fig. 22 (c) and (d) show the resulting heatmap of the frequency of faces being visited, and Fig. 22 (e) and (f) show the resulting heatmap with uniform weight as a comparison. It can be observed that when weights are assigned, the

critical parts are visited more frequently than adjacent areas. Applying weights increases the complexity of the optimization problem, leading to a longer path length and higher number of viewpoints required, while convergence is still stable. This example demonstrates the capability of the proposed methodology to prioritize critical details and components that require more stringent inspections based on prior knowledge from engineering expertise, structural analysis, or history of deteriorations. Additionally, covering the structure redundantly is proved feasible, leveraging the potential of tailoring the proposed framework to capture overlapping images for 3D reconstruction in the future work.

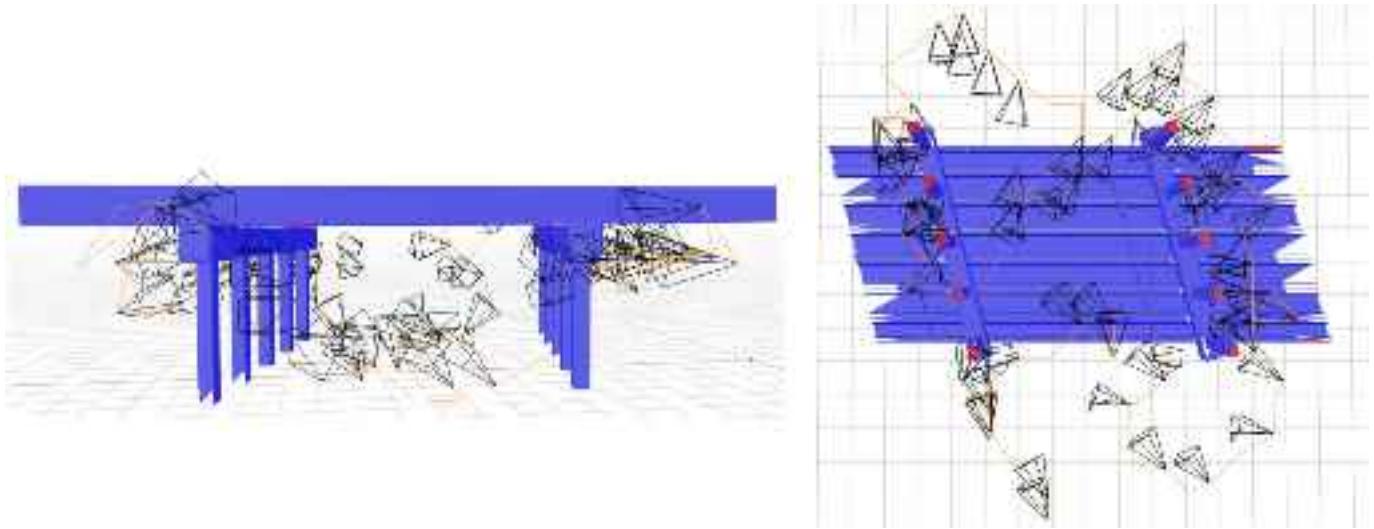


Fig. 21. Visualization for partial space inspection paths (no-fly zone above the bridge).

5.12. Test on different structures

The proposed method is tested on three different types of infrastructure models to verify its broad applicability, as shown in Fig. 23. These models were selected to provide a combination of diverse geometries and shapes. For all these cases, the proposed optimization algorithm successfully generated optimal paths with desired coverage, demonstrating its potential for application to multiple types of infrastructure inspection.

5.13. Discussions on implementation

Once the optimal inspection path is determined, further processing is required on the path to be executed by the UAS. Since the waypoints are initially planned in the local coordinate system, the first step involves transforming the local coordinates into global coordinates (e.g., WGS84 Geographic Coordinate System), which is a common process in Geographic Information Systems (GIS). One basic approach is to determine a transformation matrix by selecting reference points, a method explored in numerous studies on geo-referencing for BIM models [77–79]. Subsequently, the global coordinates of the waypoints can be imported using the Application Programming Interface (API) to upload the waypoint information to the UAS.

During UAS flight operations, Real-Time Kinematics (RTK) can be utilized to enhance positioning accuracy. As the proposed path planning method is performed offline, unmodeled obstacles can cause potential risks. Therefore, integrating online obstacle detection and avoidance functions is crucial for real-world applications [80]. With the rapid development of UAS, this functionality has been widely applied in commercial drones, enabling the implementation of the proposed path planning method.

Moreover, the proposed method also holds promise for multi-UAS inspections. Instead of initializing and performing the operators on a single path, multiple paths can be optimized simultaneously. Fitness can be defined as the total coverage of multiple paths (constraint) and the maximum length among them (objective), representing the optimality of completing the task with multi-UAS cooperation. Collision risks can be addressed by penalizing the fitness if pairs of UAS fail to maintain a desired safe distance from each other. While the proposed technique is generic and applicable to such problems, the specific formulations are outside the scope of this paper.

6. Conclusions

This paper presented a comprehensive method for UAS-based structural inspection path planning to enhance the efficiency and reliability of visual data collection. The proposed SIPP process encompasses several key stages, including modeling, space configuration, visibility quantification, and a two-stage path planning scheme. This scheme included a waypoint location optimization and a greedy pose determination, as opposed to directly optimizing all three locations and two pose variables simultaneously. To illustrate the proposed methodology, we employ a 3-span concrete girder bridge as a demonstrative example. The feasibility of employing a genetic optimization algorithm to solve SIPP was demonstrated by the consistent success of the technique in converging to paths with the desired coverage constraint. Visualization and analysis of the resulting paths reveal their tendency to meander around the bridge, with most viewpoints uniquely contributing to coverage accumulation, ensuring that the coverage constraint is met efficiently. Notably, the optimization algorithm demonstrates automatic error correction capabilities when detrimental mutations occur. It was also shown that the proposed two-stage optimization method provides a significant advantage over the direct 5DOF optimization.

The sensitivity analysis of the optimization parameters highlighted the significant influence of reasonable population and generation settings on convergence and computational cost-effectiveness. Moreover, it was demonstrated that evolution rates can be fine-tuned to improve the quality of optimal results. The application of partial rule-based initialization and a moderate tournament size also contribute to enhancing optimization outcomes. It was further demonstrated that the inspection parameters, including coverage constraint and FOV, significantly impact the optimization and greedy pose determination. Notably, higher coverage constraints and smaller FOV values result in more intricate inspection paths and poses, aligning with intuitive expectations.

Subsequently, we explored the potential of the proposed method to address specific SIPP characteristics and demonstrate its flexibility and broad applicability. In one case, we limit the inspection space to the underside of the bridge to prevent distractions to the drivers above. In another, we assign different importance levels to structural components based on engineering knowledge, introducing a weighted overlap requirement for optimization. This enables inspectors to assign greater importance to critical components, thereby enhancing the overall inspection efficiency.

Finally, we tested the proposed method on different types of infrastructure models. The method shows stable convergence and satisfactory performance in all cases, promising its broad applicability. Given

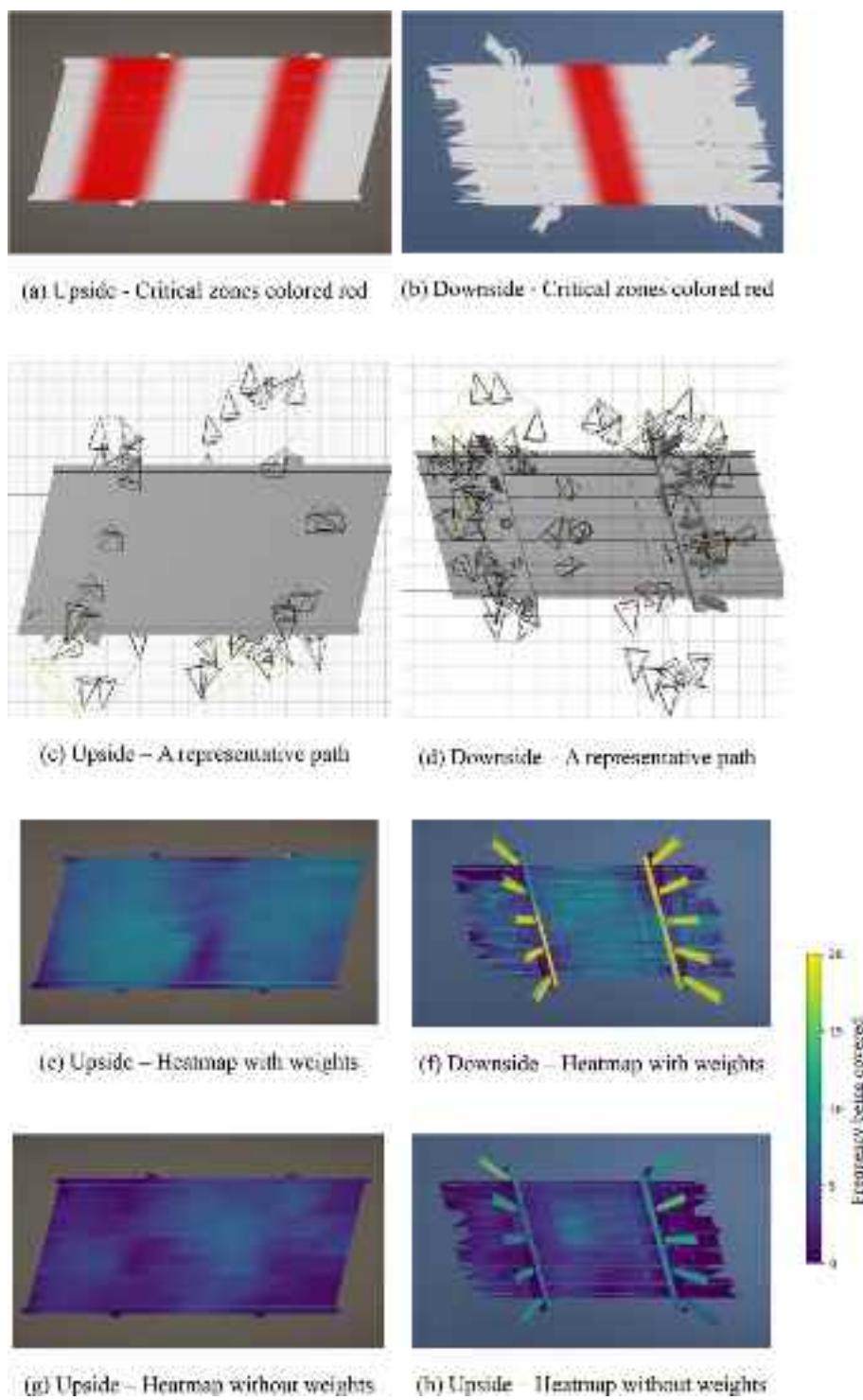


Fig. 22. Focused inspection experiment results.

the growing use of UAS, we believe that the proposed method lays a solid foundation for implementing automated UAS-based inspection approaches in infrastructure inspection.

While the proposed methodology successfully achieved the desired objectives, we recognize opportunities for further refinement. As the current method relies on a model of the structure as input, it is not readily applicable to cases without an existing model. The proposed method also determines the optimal path, assuming that the model accurately encompasses the *in situ* configurations of objects and geometries. As such, the path can not be adjusted in real-time during flights in

case of the presence of objects unaccounted for in the model, such as newly added maintenance details or overgrown vegetation. In such cases, where an *a priori* model is not present, or the system is expected to actively adjust to the environment, the application of algorithms based on reinforcement learning holds promise for real-time online path planning.

Additionally, as this study mainly focuses on visual inspections, the inspection coverage, quality, and efficiency are mainly considered in this study, while UAS can be used to conduct additional tasks in practice, including photogrammetric modeling, defect identification, and

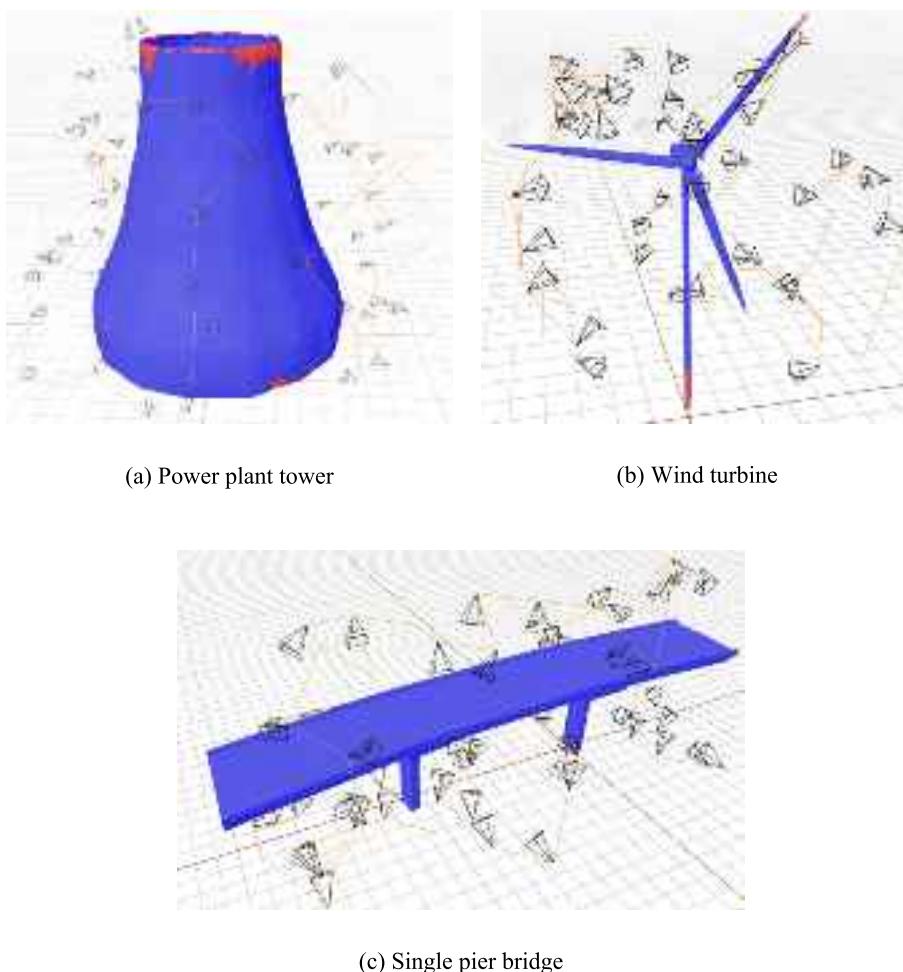


Fig. 23. Planned paths for different infrastructure models.

nondestructive scanning (e.g., infrared, radar). Therefore, considering additional performance metrics, such as 3D reconstruction and model generation, visual detection quality, and scanning resolution and accuracy, can further enhance the capacity of the proposed method to meet the diverse demands of SIPP.

While the two-stage optimization scheme devised in this paper was shown to simplify and accelerate convergence and lead to smaller path lengths (see Fig. 12 and the corresponding discussion), the two-stage formula results in the exclusion of the number of camera angles within the fitness. This is because the camera angles are decided using the greedy algorithm after the path optimization is concluded. This is based on the assumption that the cost associated with capturing an additional photo at a different angle at a single viewpoint is negligible compared to the cost of travel to new viewpoints. Such assumptions are borne out by empirical battery usage analyses for specific UAS hardware, indicating that battery consumption is dominated by certain flight maneuvers [81]. If this assumption is not warranted, the number of photos taken at different angles could be explicitly considered in the fitness.

CRediT authorship contribution statement

Yuxiang Zhao: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Benhao Lu:** Visualization, Data curation. **Mohamad Alipour:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Mohamad Alipour reports equipment, drugs, or supplies was provided by NVIDIA Corp. If there are other authors, they 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

The data and code relevant to this study are available at: github.com/zhaoyx-uiuc/GA-path-planning

Acknowledgements

The authors would like to express their sincere appreciation to the NVIDIA Corporation for their generous support with the donation of GPUs used in this study.

References

- [1] American Society of Civil Engineers (ASCE), Report card for American's Infrastructure, URL, <https://infrastructurereportcard.org/>, 2021.
- [2] B. Hubbard, S. Hubbard, Unmanned aircraft systems (UAS) for bridge inspection safety, *Drones* 4 (3) (2020) 40, <https://doi.org/10.3390/drones4030040>.
- [3] M. McGuire, M.J. Rys, A. Rys, A study of how unmanned aircraft systems can support the Kansas Department of Transportation's efforts to improve efficiency, safety, and cost reduction, 2016. URL, <https://rosap.ntl.bts.gov/view/dot/31030>.

- [4] M.N. Gillins, D.T. Gillins, C. Parrish, Cost-effective bridge safety inspections using unmanned aircraft systems (UAS), in: Geotechnical and Structural Engineering Congress, 2016, pp. 1931–1940, <https://doi.org/10.1061/9780784479742.16>.
- [5] B.F. Spencer, V. Hoskere, Y. Narazaki, Advances in computer vision-based civil infrastructure inspection and monitoring, *Engineering* 5 (2) (2019) 199–222, <https://doi.org/10.1016/j.eng.2018.11.030>.
- [6] J.K. Chow, K.F. Liu, P.S. Tan, Z. Su, J. Wu, Z. Li, Y.H. Wang, Automated defect inspection of concrete structures, *Autom. Constr.* 132 (2021) 103959, <https://doi.org/10.1016/j.autcon.2021.103959>.
- [7] E. Menendez, J.G. Victores, R. Montero, S. Martínez, C. Balaguer, Tunnel structural inspection and assessment using an autonomous robotic system, *Autom. Constr.* 87 (2018) 117–126, <https://doi.org/10.1016/j.autcon.2017.12.001>.
- [8] S. Agnisarman, S. Lopes, K.C. Madathil, K. Piratla, A. Gramopadhye, A survey of automation-enabled human-in-the-loop systems for infrastructure visual inspection, *Autom. Constr.* 97 (2019) 52–76, <https://doi.org/10.1016/j.autcon.2018.10.019>.
- [9] A.J. Lee, W. Song, B. Yu, D. Choi, C. Tirtawardhana, H. Myung, Survey of robotics technologies for civil infrastructure inspection, *J. Infrastruct. Intellig. Resilience* 2 (1) (2023) 100018, <https://doi.org/10.1016/j.intel.2022.100018>.
- [10] S. Lafamme, F. Ubertini, A.D. Matteo, A. Pirrotta, M. Perry, Y. Fu, J. Li, H. Wang, T. Hoang, B. Glisic, L.J. Bond, M. Pereira, Y. Shu, K.J. Loh, Y. Wang, S. Ding, X. Wang, X. Yu, B. Han, Y. Goldfeld, D. Ryu, R. Napolitano, F. Moreu, G. Giardina, P. Milillo, Roadmap on measurement technologies for next generation structural health monitoring systems, *Meas. Sci. Technol.* 34 (9) (2023) 093001, <https://doi.org/10.1088/1361-6501/acd135>.
- [11] Z. Wang, Q. Gao, J. Xu, D. Li, A review of UAV power line inspection, in: Proceedings of 2020 International Conference on Guidance, Navigation and Control, 2022, pp. 3147–3159, https://doi.org/10.1007/978-981-15-8155-7_263.
- [12] Y. Zhang, X. Yuan, W. Li, S. Chen, Automatic power line inspection using UAV images, *Remote Sens.* 9 (8) (2017) 824, <https://doi.org/10.3390/rs9080824>.
- [13] 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, <https://doi.org/10.1016/j.autcon.2021.103881>.
- [14] Y. Tan, G. Li, R. Cai, J. Ma, M. Wang, Mapping and modelling defect data from UAV captured images to BIM for building external wall inspection, *Autom. Constr.* 139 (2022) 104284, <https://doi.org/10.1016/j.autcon.2022.104284>.
- [15] K. Chen, G. Reichard, A. Akanmu, X. Xu, Geo-registering UAV-captured close-range images to GIS-based spatial model for building façade inspections, *Autom. Constr.* 122 (2021) 103503, <https://doi.org/10.1016/j.autcon.2020.103503>.
- [16] Y. Wu, Y. Qin, Z. Wang, L. Jia, A UAV-based visual inspection method for rail surface defects, *Appl. Sci.* 8 (7) (2018) 1028, <https://doi.org/10.3390/app8071028>.
- [17] M. Banić, A. Miltenović, M. Pavlović, I. Čirić, Intelligent machine vision based railway infrastructure inspection and monitoring using UAV, *Facta Universitatis, Ser. Mechan. Eng.* 17 (3) (2019) 357–364, <https://doi.org/10.22190/FUME190507041B>.
- [18] S. Chen, D.F. Laefer, E. Mangina, I. Zolanvari, J. Byrne, UAV bridge inspection through evaluated 3d reconstructions, *J. Bridg. Eng.* 24 (4) (2019) 05019001, [https://doi.org/10.1061/\(ASCE\)BE.1943-5592.0001343](https://doi.org/10.1061/(ASCE)BE.1943-5592.0001343).
- [19] N. Bolourian, A. Hammad, Lidar-equipped UAV path planning considering potential locations of defects for bridge inspection, *Autom. Constr.* 117 (2020) 103250, <https://doi.org/10.1016/j.autcon.2020.103250>.
- [20] S. Dorafshan, L.E. Campbell, M. Maguire, R.J. Connor, Benchmarking unmanned aerial systems-assisted inspection of steel bridges for fatigue cracks, *Transp. Res.* 2675 (9) (2021) 154–166, <https://doi.org/10.1177/03611981211001073>.
- [21] C.D. Lippitt, S. Zhang, The impact of small unmanned airborne platforms on passive optical remote sensing: a conceptual perspective, *Int. J. Remote Sens.* 39 (15–16) (2018) 4852–4868, <https://doi.org/10.1080/01431161.2018.1490504>.
- [22] H. Freimuth, M. König, Planning and executing construction inspections with unmanned aerial vehicles, *Autom. Constr.* 96 (2018) 540–553, <https://doi.org/10.1016/j.autcon.2018.10.016>.
- [23] S. Zhang, S.M. Bogus, S.V. Baros, P.R. Neville, H.A. Barrett, T. Eshelman, Bridge deck surface distress evaluation using S-UAS acquired high-spatial resolution aerial imagery, *Ann. GIS* 29 (2) (2023) 261–272, <https://doi.org/10.1080/19475683.2023.2166112>.
- [24] J.J. Lin, A. Ibrahim, S. Sarwade, M. Golparvar-Fard, Bridge inspection with aerial robots: automating the entire pipeline of visual data capture, 3D mapping, defect detection, analysis, and reporting, *J. Comput. Civ. Eng.* 35 (2) (2021) 04020064, [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000954](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000954).
- [25] P. Kim, J. Park, Y.K. Cho, J. Kang, UAV-assisted autonomous mobile robot navigation for as-is 3D data collection and registration in cluttered environments, *Autom. Constr.* 106 (2019) 102918, <https://doi.org/10.1016/j.autcon.2019.102918>.
- [26] G. Hardouin, F. Morbidi, J. Moras, J. Marzat, E.M. Mouaddib, Surface-driven next-best-view planning for exploration of large-scale 3D environments, *IFAC-PapersOnLine* 53 (2) (2020) 15501–15507, <https://doi.org/10.1016/j.ifacol.2020.12.2376>.
- [27] P.T. Lin, K.S. Tseng, Maximal coverage problems with routing constraints using cross-entropy Monte Carlo tree search, *Auton. Robot.* 48 (1) (2024), <https://doi.org/10.1007/s10514-024-10156-6>.
- [28] F.S. Hover, R.M. Eustice, A. Kim, B. Englot, H. Johannsson, M. Kaess, J.J. Leonard, Advanced perception, navigation and planning for autonomous in-water ship hull inspection, *Int. J. Robot. Res.* 31 (12) (2012) 1445–1464, <https://doi.org/10.1177/0278364912461059>.
- [29] Z. Yan, N. Jouandeau, A.A. Cherif, ACS-PRM: Adaptive cross sampling based probabilistic roadmap for multi-robot motion planning, in: Intelligent Autonomous Systems 12: Volume 1 Proceedings of the 12th International Conference IAS-12, 2013, pp. 843–851, https://doi.org/10.1007/978-3-642-33926-4_81.
- [30] M. Nazarahari, E. Khamirza, S. Doostie, Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm, *Expert Syst. Appl.* 115 (2019) 106–120, <https://doi.org/10.1016/j.eswa.2018.08.008>.
- [31] A.R. Soltani, H. Tawfik, J.Y. Goulermas, T. Fernando, Path planning in construction sites: performance evaluation of the Dijkstra, A*, and GA search algorithms, *Adv. Eng. Inform.* 16 (4) (2002) 291–303, [https://doi.org/10.1016/S1474-0346\(03\)00018-1](https://doi.org/10.1016/S1474-0346(03)00018-1).
- [32] C.S. Tan, R. Mohd-Mokhtar, M.R. Arshad, A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms, *IEEE Access* 9 (2021) 119310–119342, <https://doi.org/10.1109/ACCESS.2021.3108177>.
- [33] C. Luo, H. Mo, F. Shen, W. Zhao, Multi-goal motion planning of an autonomous robot in unknown environments by an ant colony optimization approach, in: *Advances in Swarm Intelligence: 7th International Conference, 2016, Proceedings, Part II* 7, pp. 519–527, doi: https://doi.org/10.1007/978-3-319-41009-8_56.
- [34] R. Rashid, N. Perumal, I. Elamvazuthi, M.K. Tagdeen, M.K.A. Ahmed Khan, S. Parasuraman, Mobile robot path planning using Ant Colony Optimization, in: *2016 2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA)*, 2016, pp. 1–6, <https://doi.org/10.1109/ROMA.2016.7847836>.
- [35] S. Sahu, B.B. Choudhury, PSO based path planning of a six-axis industrial robot, in: *Computational Intelligence in Data Mining: Proceedings of the International Conference on ICCIDM*, 2018, pp. 213–220, https://doi.org/10.1007/978-981-13-8676-3_19.
- [36] 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, <https://doi.org/10.1016/j.autcon.2017.04.013>.
- [37] Y.H. Lin, S.M. Wang, L.C. Huang, M.C. Fang, Applying the stereo-vision detection technique to the development of underwater inspection task with PSO-based dynamic routing algorithm for autonomous underwater vehicles, *Ocean Eng.* 139 (2017) 127–139, <https://doi.org/10.1016/j.oceaneng.2017.04.051>.
- [38] 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)*, 2020, pp. 1480–1486, <https://doi.org/10.1109/IROS45743.2020.9341089>.
- [39] K.O. Ellefsen, H.A. Lepikson, J.C. Albiez, Multiobjective coverage path planning: enabling automated inspection of complex, real-world structures, *Appl. Soft Comput.* 61 (2017) 264–282, <https://doi.org/10.1016/j.asoc.2017.07.051>.
- [40] J. Xie, L.R.G. Carrillo, L. Jin, Path planning for UAV to cover multiple separated convex polygonal regions, *IEEE Access* 8 (2020) 51770–51785, <https://doi.org/10.1109/ACCESS.2020.2980203>.
- [41] M. Puliti, G. Montaggioli, A. Sabato, Automated subsurface defects' detection using point cloud reconstruction from infrared images, *Autom. Constr.* 129 (2021) 103829, <https://doi.org/10.1016/j.autcon.2021.103829>.
- [42] E. Kaartinen, K. Dunphy, A. Sadhu, LiDAR-based structural health monitoring: applications in civil infrastructure systems, *Sensors* 22 (12) (2022) 4610, <https://doi.org/10.3390/s22124610>.
- [43] C. Maierhofer, Nondestructive evaluation of concrete infrastructure with ground penetrating radar, *J. Mater. Civ. Eng.* 15 (3) (2003) 287–297, [https://doi.org/10.1061/\(ASCE\)0899-1561\(2003\)15:3\(287\)](https://doi.org/10.1061/(ASCE)0899-1561(2003)15:3(287)).
- [44] A. Ibrahim, M. Golparvar-Fard, K. El-Rayes, Metrics and methods for evaluating model-driven reality capture plans, *Comput. Aided Civ. Inf. Eng.* 37 (1) (2022) 55–72, <https://doi.org/10.1111/mice.12693>.
- [45] D.Y. Yang, D.M. Frangopol, Risk-based inspection planning of deteriorating structures, *Struct. Infrastruct. Eng.* 18 (1) (2021) 109–128, <https://doi.org/10.1080/15732479.2021.1907600>.
- [46] N.J. Bertola, E. Brühwiler, Risk-based methodology to assess bridge condition based on visual inspection, *Struct. Infrastruct. Eng.* 19 (4) (2023) 575–588, <https://doi.org/10.1080/15732479.2021.1959621>.
- [47] D. Lattanzi, G. Miller, Review of robotic infrastructure inspection systems, *J. Infrastruct. Syst.* 23 (3) (2017) 04017004, [https://doi.org/10.1061/\(ASCE\)IS.1943-555X.0000353](https://doi.org/10.1061/(ASCE)IS.1943-555X.0000353).
- [48] I. Jeelani, M. Gheisari, Safety challenges of human-drone interactions on construction jobsites, in: *Automation and Robotics in the Architecture, Engineering, and Construction Industry*, 2022, pp. 143–164, https://doi.org/10.1007/978-3-030-77163-8_7.
- [49] Y. Narazaki, V. Hoskere, G. Chowdhary, B.F. Spencer, Vision-based navigation planning for autonomous post-earthquake inspection of reinforced concrete railway viaducts using unmanned aerial vehicles, *Autom. Constr.* 137 (2022) 104214, <https://doi.org/10.1016/j.autcon.2022.104214>.
- [50] M. Maboudi, M.R. Homaei, S. Song, S. Malihai, M. Saadatseresh, M. Gerke, A review on viewpoints and path planning for UAV-based 3D reconstruction, *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.* 16 (2023), <https://doi.org/10.1109/JSTARS.2023.3276427>.
- [51] C. Hoppe, A. Wendel, S. Zollmann, K. Pirker, A. Irschara, H. Bischof, S. Kluckner, Photogrammetric camera network design for micro aerial vehicles, in: *Computer Vision Winter Workshop (CVWW)*, 2012. <https://api.semanticscholar.org/CorpusID:195178458.z>.
- [52] W. Jing, J. Polden, P.Y. Tao, W. Lin, K. Shimada, View planning for 3d shape reconstruction of buildings with unmanned aerial vehicles, in: *14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2016, <https://doi.org/10.1109/ICARCV.2016.7838774>.
- [53] R. Huang, D. Zou, R. Vaughan, P. Tan, Active image-based modeling with a toy drone, in: *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, <https://doi.org/10.1109/ICRA.2018.8460673>.

- [54] X. Zhou, K. Xie, K. Huang, Y. Liu, Y. Zhou, M. Gong, H. Huang, Offsite aerial path planning for efficient urban scene reconstruction, *ACM Trans. Graph. (TOG)* 39 (6) (2020) 1–16, <https://doi.org/10.1145/3414685.3417791>.
- [55] N. Smith, N. Moehrle, M. Goesele, W. Heidrich, Aerial path planning for urban scene reconstruction: a continuous optimization method and benchmark, *ACM Trans. Graph. (TOG)* 37 (6) (2018) 1–15, <https://doi.org/10.1145/3272127.3275010>.
- [56] F. Yan, E. Xia, Z. Li, Z. Zhou, Sampling-based path planning for high-quality aerial 3D reconstruction of urban scenes, *Remote Sens.* 13 (5) (2021) 989, <https://doi.org/10.3390/rs13050989>.
- [57] C. Peng, V. Isler, Visual coverage path planning for urban environments, *IEEE Robot. Autom. Lett.* (2020) 5961–5968, <https://doi.org/10.1109/LRA.2020.3010745>.
- [58] O. Sharma, N. Arora, H. Sagar, Image acquisition for high quality architectural reconstruction, 45th Graphics Interface Conference, 2019. <https://api.semanticscholar.org/CorpusID:197467664>.
- [59] H.W. Tong, B. Li, H. Huang, C.Y. Wen, UAV path planning for complete structural inspection using mixed viewpoint generation, in: 2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV), 2022, pp. 727–732, <https://doi.org/10.1109/ICARCV57592.2022.10004359>.
- [60] L. Yu, M.M. Huang, S. Jiang, C. Wang, M. Wu, Unmanned aircraft path planning for construction safety inspections, *Autom. Constr.* 154 (2023) 105005, <https://doi.org/10.1016/j.autcon.2023.105005.8e>.
- [61] Y. Tan, W. Yi, P. Chen, Y. Zou, An adaptive crack inspection method for building surface based on BIM, UAV and edge computing, *Autom. Constr.* 157 (2024) 105161, <https://doi.org/10.1016/j.autcon.2023.105161>.
- [62] 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), 2015, pp. 6423–6430, <https://doi.org/10.1109/ICRA.2015.7140101>.
- [63] 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. Robot.* 40 (2016) 1059–1078, <https://doi.org/10.1007/s10514-015-9517-1>.
- [64] I.Z. Biundini, M.F. Pinto, A.G. Melo, A.L. Marcato, L.M. Honório, M.J. Aguiar, A framework for coverage path planning optimization based on point cloud for structural inspection, *Sensors* 21 (2) (2021) 570, <https://doi.org/10.3390/s21020570>.
- [65] P. Janousek, J. Faigl, Speeding up coverage queries in 3D multi-goal path planning, in: 2013 IEEE International Conference on Robotics and Automation (ICRA), 2013, pp. 5082–5087, <https://doi.org/10.1109/ICRA.2013.6631303>.
- [66] Y. Tao, Y. Wen, H. Gao, T. Wang, J. Wan, J. Lan, A path-planning method for wall surface inspection robot based on improved genetic algorithm, *Electronics* 11 (8) (2022) 1192, <https://doi.org/10.3390/electronics11081192>.
- [67] A. Ibrahim, M. Golparvar-Fard, K. El-Rayes, Multiobjective optimization of reality capture plans for computer vision-driven construction monitoring with camera-equipped UAVs, *J. Comput. Civ. Eng.* 36 (5) (2022) 04022018, [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0001032](https://doi.org/10.1061/(ASCE)CP.1943-5487.0001032).
- [68] S.S. Mansouri, C. Kanellakis, D. Wuthier, E. Fresk, G. Nikolakopoulos, Cooperative aerial coverage path planning for visual inspection of complex infrastructures, *Control. Eng. Pract.* 74 (2018) 118–131, <https://doi.org/10.1016/j.conengprac.2018.03.002>.
- [69] B. Hepp, M. Nießner, O. Hilliges, Plan3d: viewpoint and trajectory optimization for aerial multi-view stereo reconstruction, *ACM Trans. Graph. (TOG)* 38 (1) (2018) 1–17, <https://doi.org/10.1145/3233794>.
- [70] M. Roberts, D. Dey, A. Truong, S. Sinha, S. Shah, A. Kapoor, P. Hanrahan, N. Joshi, Submodular trajectory optimization for aerial 3d scanning, in: Proceedings of the IEEE International Conference on Computer Vision, 2017. <https://api.semanticscholar.org/CorpusID:969140>.
- [71] S. Ivić, B. Crnković, L. Grbčić, L. Matleković, Multi-UAV trajectory planning for 3D visual inspection of complex structures, *Autom. Constr.* 147 (2023) 104709, <https://doi.org/10.1016/j.autcon.2022.104709>.
- [72] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, R. Siegwart, Receding horizon path planning for 3D exploration and surface inspection, *Auton. Robot.* 42 (2018) 291–306, <https://doi.org/10.1007/s10514-016-9610-0>.
- [73] C. Papachristos, F. Mascarich, S. Khattak, T. Dang, K. Alexis, Localization uncertainty-aware autonomous exploration and mapping with aerial robots using receding horizon path-planning, *Auton. Robot.* 43 (2019) 2131–2161, <https://doi.org/10.1007/s10514-019-09864-1>.
- [74] S. Song, D. Kim, S. Jo, Online coverage and inspection planning for 3D modeling, *Auton. Robot.* 44 (2020) 1431–1450, <https://doi.org/10.1007/s10514-020-09936-7>.
- [75] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, 1989.
- [76] F.A. Fortin, F.M. De Rainville, M.A.G. Gardner, M. Parizeau, C. Gagné, DEAP: evolutionary algorithms made easy, *J. Mach. Learn. Res.* 13 (1) (2012) 2171–2175, <https://doi.org/10.5555/2503308.2503311>.
- [77] A.A. Diakite, S. Zlatanova, Automatic geo-referencing of BIM in GIS environments using building footprints, *Comput. Environ. Urban. Syst.* 80 (2020) 101453, <https://doi.org/10.1016/j.compenvurbsys.2019.101453>.
- [78] Y. Deng, J.C.P. Cheng, C. Anumba, Mapping between BIM and 3D GIS in different levels of detail using schema mediation and instance comparison, *Autom. Constr.* 67 (2016) 1–21, <https://doi.org/10.1016/j.autcon.2016.03.006>.
- [79] S. Jaud, A. Donaubauer, O. Heuncke, A. Borrman, Georeferencing in the context of building information modelling, *Autom. Constr.* 118 (2020) 103211, <https://doi.org/10.1016/j.autcon.2020.103211>.
- [80] P. Fraga-Lamas, L. Ramos, V. Mondéjar-Guerra, T.M. Fernández-Caramés, A review on IoT deep learning UAV systems for autonomous obstacle detection and collision avoidance, *Remote Sens.* 11 (18) (2019) 2144, <https://doi.org/10.3390/rs11182144>.
- [81] H.V. Abeywickrama, B.A. Jayawickrama, Y. He, E. Dutkiewicz, Comprehensive energy consumption model for unmanned aerial vehicles, based on empirical studies of battery performance, *IEEE Access* 6 (2018) 58383–58394, <https://doi.org/10.1109/ACCESS.2018.2875040>.