Three-Dimensional Modeling and Crack Detection for High Pier Bridges Based on UAV Route Optimization Algorithm

Xinfeng Yin, Ph.D.[1]; Xin Fu[2]; Yang Quan[3]; Tao Chen[4]; and Zhou Huang, Ph.D., F.ASCE[5]

Abstract: This study proposes a path optimization algorithm specifically designed for high-pier bridges in mountainous areas. By integrating drone oblique photogrammetry with an improved dung beetle optimization algorithm, the research achieves high-resolution image acquisition and rapid modeling of refined 3D models. The study employs an improved dung beetle optimization algorithm, incorporating chaotic mapping, adaptive weights, and spiral search strategies to optimize flight path planning and enhance the accuracy of the bridge's 3D model. Subsequently, the algorithm is utilized to automatically detect and quantitatively analyze surface cracks on bridge piers. Finally, a specific high-pier bridge is selected as the research object to validate the effectiveness of the path planning and the accuracy of the bridge surface crack identification method. The results indicate: (1) the algorithm can be effectively applied to practical engineering projects, with superior modeling outcomes; (2) significant improvements are observed in planar and elevation accuracy, demonstrating the effectiveness of optimized path planning in enhancing modeling quality under complex terrain conditions; and (3) the crack identification accuracy exceeds 90%, with a width calculation error of less than 10%, and the algorithm can accurately locate the cracks. **DOI: [10.1061/JPCFEV.CFENG-5191](https://doi.org/10.1061/JPCFEV.CFENG-5191).** © *2025 American Society of Civil Engineers.*

Author keywords: Drone technology; Three-dimensional (3D) modeling; Bridge pier crack identification; Dung beetle optimization algorithm; High-precision modeling.

Introduction

Crack detection in large-span bridges in complex mountainous environments is critical for bridge safety assessment (Li et al. 2020; Adhikari et al. 2014; Jeong et al. 2020; Yan et al. 2025). However, traditional manual inspection methods face significant challenges, including rugged terrain, low inspection accuracy, and poor operational efficiency, and are therefore unable to meet the requirements of modern bridge health monitoring (Chen et al. 2024a, b; Yuan et al. 2024; Congress et al. 2022; Caballero-Martin et al. 2024; Zhang et al. 2025). Recent advances in unmanned aerial vehicle (UAV) technology could provide a promising solution for bridge crack detection. By combining path-planning algorithms with image-processing techniques, it is now possible to generate detailed 3D models, thus facilitating the efficient and accurate inspection of

bridges in challenging environments (Thai et al. 2024; Zhang et al. 2022; Liu et al. 2024; Huang et al. 2025).

Various path-planning algorithms have been proposed in UAV-based bridge inspection research (Wu et al. 2020; Namian et al. 2021). These algorithms have improved the efficiency and accuracy of path planning by combining various optimizations strategies and constraints. For example, Wang et al. (2024) proposed a method for generating 3D models of bridges through automatic path planning by UAVs. Zhao et al. (2024) proposed an effective two-stage method that guarantees coverage as a constraint to ensure damage detectability. However, some bridges are constructed in complex terrain, such as mountainous hills and canyons. While existing path-planning algorithms have demonstrated effectiveness in various scenarios, they still face notable limitations in global search capability, convergence efficiency, and adaptability to complex terrains. Rapidly exploring random trees has made great progress in the field of path planning, but the algorithm still suffers from low node utilization and unstable paths (Xie et al. 2020; Huang et al. 2024; Pohan et al. 2024). Therefore, there is an urgent need to optimize path-planning algorithms to improve the efficiency of searching paths. In addition, researchers have been exploring automated crack detection systems based on UAV imaging technology. For example, Duque et al. (2018) provided the state-of-the-art and of-the-practice on visual inspection, monitoring, and analysis of infrastructure using unmanned aerial vehicles. Won et al. (2021) introduced a method for generating panoramic images of bridges based on depth matching. The method generates high-quality images by image matching and random sample consensus algorithm, which overcomes the common problems of ghosting and misalignment in traditional methods. In the context of image processing and 3D modeling, "ghosting" refers to the appearance of faint, duplicate images or artifacts in the final output, often caused by issues such as camera movement or improper image stitching. Misalignment occurs when images fail to accurately overlap or match up during the stitching process, resulting in visible seams or distortions in the reconstructed 3D model. Jiang et al. (2025) developed a method to detect defects

[1]Professor, School of Civil and Environmental Engineering, Changsha Univ. of Science and Technology, Changsha, Hunan Province 410114, China. Email: yinxinfeng@163.com

[2]School of Civil and Environmental Engineering, Changsha Univ. of Science and Technology, Changsha, Hunan Province 410114, China. Email: fuxinang@163.com

[3]Ph.D. Candidate, School of Civil and Environmental Engineering, Changsha Univ. of Science and Technology, Changsha, Hunan Province 410114, China. Email: yangquandoc@163.com

[4]School of Civil and Environmental Engineering, Changsha Univ. of Science and Technology, Changsha, Hunan Province 410114, China. Email: chentao_csust@163.com

[5]School of Civil Engineering, Changsha Univ., Changsha, Hunan Province 410022, China (corresponding author). ORCID: https://orcid.org/0000-0002-9694-2259. Email: huangzhoubridge@163.com

© ASCE   04025056-1   J. Perform. Constr. Facil.

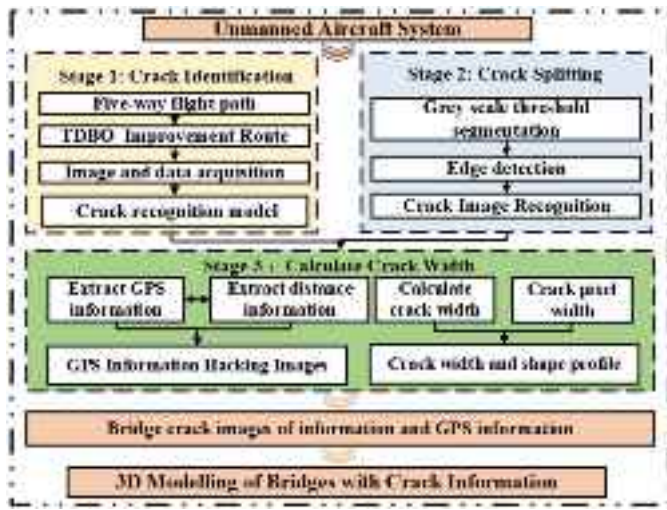J. Perform. Constr. Facil., 2025, 39(6): 04025056

Fig. 1. Unmanned aircraft system.

on the underside of bridges using close-range UAV photography and the YOLO v8 network. Although these methods have made significant advances in crack identification accuracy, there are still challenges in accurately locating the spatial position of cracks in real 3D structures (Zhou et al. 2024; Bu et al. 2022; Yin et al. 2023).

To address these challenges, this paper proposes a UAV-based path-planning method that utilizes the tuned dung beetle optimization algorithm (TDBO). The method aims to overcome the inefficiencies, inaccuracies, and lack of adaptability to complex environments that are inherent in traditional bridge pier crack detection techniques. Additionally, a systematic and reliable solution for detecting cracks in bridge abutments is introduced by integrating high-precision 3D modeling with conventional machine learning-based crack detection methods. In the path-planning phase, the improved dung beetle optimization algorithm incorporates chaotic mapping, adaptive weighting, and a spiral search strategy, thereby significantly enhancing global search capability and convergence efficiency. As a result, path coverage is improved, and computation time is reduced, even under challenging terrain conditions. During the modeling and identification phase, high-precision 3D models are generated using inclined photogrammetry, and texture mapping techniques are employed to accurately locate and quantify crack widths as well as to analyze their spatial distribution. Field tests are conducted to validate the automation level and accuracy of the proposed method. The main components of the study are organized as shown in Fig. 1.

Original Dung Beetle Optimization Algorithm

The dung beetle optimizer (DBO) algorithm is inspired by the various natural behaviors of dung beetles (Xue and Shen 2023), its inspiration comes from dung beetles' rolling, dancing, foraging, stealing, and breeding behaviors. The algorithm considers global exploration and local development, so it has the characteristics of fast convergence and high accuracy and can effectively solve complex optimization problems. The flight path optimization of the UAV is implemented by different search strategies through four agents, namely, rolling dung beetles, brood balls, baby dung beetles, and thieves. Specifically, rolling dung beetles are responsible for global search and path exploration, brood balls are solution evolution and quality enhancement, baby dung beetles are utilized to generate new

solutions and local optimization, and thieves are used to simulate competition and diversity preservation to prevent the algorithm from falling into a local optimum.

Ball-Rolling Behavior

In the simulation, the ball-rolling dung beetle moves in a straight line, following the orientation of the sun, with its position updated as follows:

$$x_i(t+1) = x_i(t) + \alpha \times k \times x_i(t-1) + b \times \Delta x \tag{1}$$

$$\Delta x = |x_i - X^W| \tag{2}$$

where t = the current cycle number; $x_i(t)$ = the position coordinates of the ith dung beetle at the tth iteration; α is a natural constant; k is a deflection factor set to 0.1; b is set to 0.05, X^W represents the global worst position; and Δx is used to simulate changes in light intensity.

When a dung beetle encounters an obstacle in its path, it alters its direction by performing a dancing maneuver to find a new route forward. To model this dancing behavior, the new forward direction is typically represented by a tangent function (Liu et al. 2022). Consequently, the position update formula for the rolling dung beetle is as follows:

$$x_i(t+1) = x_i(t) + \tan(\theta)|x_i(t) - x_i(t-1)| \tag{3}$$

where $\theta \in [0, \pi]$ = the angle of deflection.

Reproductive Behavior

The selection of suitable spawning sites is crucial for dung beetles to ensure a safe environment for their offspring. In the DBO, the spawning areas of female dung beetles are modeled as defined as follows:

$$Lb^* = \max(X^* \times (1-R), Lb) \tag{4}$$

$$Ub^* = \min(X^* \times (1+R), Ub) \tag{5}$$

where X^* = the current local optimal position; Lb^* and Ub^* = the worst and best values of the spawning area, respectively; $R = 1 - t/T_{max}$ and T_{max} represent the maximum number of iterations; and Lb and Ub represent the worst and best values of the explored range, respectively.

It is assumed that each female dung beetle produces only one larva per iteration. As the boundary of the generated region continuously changes, the location of the larvae shifts during each iteration, as represented in the following:

$$B_i(t+1) = X^* + b_1 \times (B_i(t) - Lb^*) + b_2 \times (B_i(t) - Ub^*) \tag{6}$$

where $B_i(t)$ = the position information of the i larva at the t cycle; and b_1 and b_2 = two different random number columns.

Foraging Behavior

When a small dung beetle forages, the algorithm sets an optimal foraging area defined as follows:

$$Lb^b = \max(X^b \times (1-R), Lb) \tag{7}$$

$$Ub^b = \min(X^b \times (1+R), Ub) \tag{8}$$

where X^b = the global optimum; and Lb^b and Ub^b = the worst and best values of the optimal foraging area, respectively.

© ASCE

04025056-2

J. Perform. Constr. Facil.

The rules for changing the position of a small dung beetle are as follows:

$$x_i(t+1) = x_i(t) + C_1 \times (x_i(t) - Lb^b) + C_2 \times (x_i(t) - Ub^b) \quad (9)$$

where C_1 = a normally distributed random variable; and C_2 = a random vector that takes values in the interval (0,1).

Theft Behavior

Moreover, the positioning strategy of the thief dung beetle is updated as follows:

$$x_i(t+1) = X^b + S \times h \times (|x_i(t) - X^*|) + (|x_i(t) - X^b|) \quad (10)$$

where h = a random vector of size $1 \times D$; and S is a constant.

Improving the Dung Beetle Optimization Algorithm

Introduction of Chaotic Mapping

The classical DBO algorithms typically rely on the random point method for global search. However, this approach often leads to insufficient population diversity, which negatively affects the convergence speed of the algorithm. To overcome this limitation, chaotic mapping is introduced (Li et al. 2021; Moysis et al. 2021). The nonlinear and periodic properties of chaotic mapping allow for the generation of more complex and random sequences, thereby enhancing population diversity and preventing premature convergence to local optima. Among the various chaotic mappings, the circle mapping is relatively more stable and provides better distribution uniformity. Consequently, the circle mapping is used to initialize the population of the DBO algorithm, as described by the following formula:

$$x_{n+1} = \text{mod}\left(3.85x_n + 0.4 - \frac{0.7}{3.85\pi}\sin(3 \cdot 85\pi x_n), 1\right) \quad (11)$$

where mod $(\cdot)$ = a modulus operation; and n = the latitude of the solution, generally taken as 2,000. Thus, the circle mapping chaotic initialization distribution is given in Fig. 2, with a relatively uniform sequence data distribution.

Adaptive Weights

The classical DBO employs a fixed population size and proportion during the iteration process, which limit its ability to dynamically adapt to the optimization requirements at different stages.
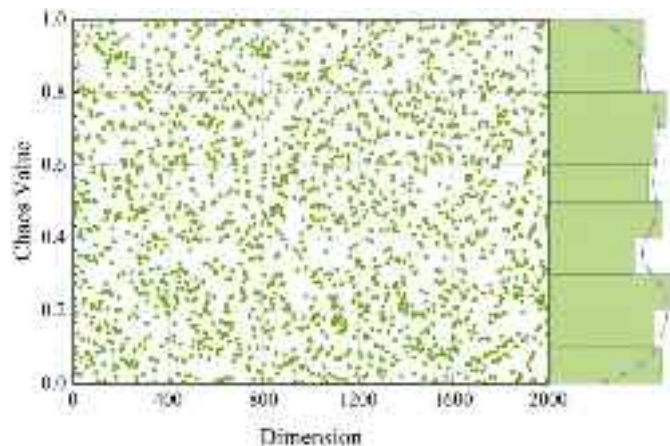

Fig. 2. Circle mapping chaotic initialization distribution.

This rigidity significantly hinders exploration efficiency and flexibility, often resulting in premature convergence and unnecessary consumption of computational resources. To address these issues, this paper proposes an adaptive weight adjustment strategy that automatically adjusts the weights based on the specific stages of the optimization process and the current environment. On one hand, the weights are adjusted as the number of iterations increases to prevent stagnation in the search process. On the other hand, the weights are modified when the population approaches a local optimum, encouraging the population to escape local optimal and continue searching for the global optimum. The specific formula is as follows:

$$P_p(t) = P_p(t_1) - t \times \left(\frac{P_p(t_1) - P_p(t_{\max})}{T_{\max}}\right) \quad (12)$$

$$p\text{Num}(t) = N \times P_p(t) \times t \quad (13)$$

where $P_p(t_1)$ and $P_p(t_{\max})$ = the producer weights at the beginning and end of the iteration; N = the dung beetle size; and $p\text{Num}(t)$ represents the number of dung beetles.

Spiral Search Strategy

The classical DBO utilizes a tangent function to simulate the adjustment direction of the dung beetle when encountering obstacles. However, due to its high randomness, parameter sensitivity, and limited convergence capability, this approach often leads to an unstable search process, negatively affecting global and local optimization performance. To overcome these limitations, a spiral search strategy is introduced, which constructs a search model based on rotational motion. This strategy enhances the global search capability of the algorithm and helps prevent premature convergence to local optima. The updated dung beetle position formula is given in Eq. (14)

$$\begin{aligned}X_i(t+1) = X^* + e^{rl} \cdot \cos(2\pi l) \times b_1 \times (X_i(t) - Lb^*) \\ + e^{rl} \cdot \cos(2\pi l) \times b_2 \times (B_i(t) - Ub^*)\end{aligned} \quad (14)$$

where b_1 and b_2 = two independent random vectors; and r is a parameter of the dynamic spiral search shape; and l is a random number between [0, 1].

Algorithm Accuracy Evaluation

The UAV 3D path-planning problem is an objective function optimization problem, where the optimization value of the objective function is typically used to assess the performance of the algorithm (Fang and Savkin 2024). For example, the six typical benchmark functions from the CEC2005 test suite are used to evaluate the optimization performance of the TDBO algorithm (Cheng et al. 2021). A comparison is made with the original dung beetle optimization algorithm (DBO), the grey wolf optimization algorithm (GWO), and the sparrow search algorithm (SSA) to assess the optimization effectiveness of the TDBO algorithm. Four single-modal benchmark functions (F1–F4) and two multimodal benchmark functions (F9–F10) are selected for testing the method. The details of the test function and the constructed profile are presented in Table 1 and Fig. 3, respectively.

To increase the reliability of the test results and minimize the impact of random variations, the group size for each test is set at 30, with 500 iterations per test. For each benchmark function, 30 independent test groups are performed, and the optimal results from each group are recorded, as shown in Fig. 4. The mean, best value, and standard deviation are then calculated. The detailed results are presented in Table 2.

Among the six typical benchmark functions (F1–F4, F9–F10) in the CEC2005 test suite, the TDBO algorithm significantly

Table 1. Test function information

Function formula	Dimensionality	Search area	Theoretical value
$F_1(x) = \sum_{i=1}^{D} x_i^2$	30	$[-100, 100]$	0
$F_2(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i|$	30	$[-100, 100]$	0
$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	30	$[-100, 100]$	0
$F_4(x) = \max(|x_1|, |x_2|, \ldots, |x_n|)$	30	$[-100, 100]$	0
$F_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	$[-5.12, 5.12]$	0
$F_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)$	30	$[-32, 32]$	0

outperforms DBO, GWO, and SSA. In the single-modal benchmark function tests (F1–F4), TDBO excels in the best, mean, and Std metrics. In the F1 function, TDBO demonstrates optimal performance, while in the F2, F3, and F4 functions, it also achieves higher accuracy, approaching the theoretical optimal solution.

In the tests of multimodal benchmark functions (F9, F10), TDBO maintains a significant advantage. In the F9 function, TDBO reaches the theoretical optimal solution, whereas the other algorithms perform relatively poorly. In the F10 function, although the best values of all four algorithms are close to the optimal solution, TDBO outperforms the others in terms of mean and Std. Overall, TDBO's mean and best values generally surpass those of the other three algorithms across the three types of benchmark functions, and TDBO also converges with fewer iterations to the same level of accuracy.

Inclined Photography and Route Planning

Photogrammetric technology enables the precise reproduction of an object's spatial position and 3D morphology by analyzing images captured from multiple viewpoints. This process involves several key steps, including image preprocessing, aerial triangulation with control points, the generation of a high-density point cloud, model construction, texture mapping, and the final reconstruction of a detailed 3D model. Sophisticated UAV flight path planning is an essential aspect of oblique photogrammetry. Commonly used flight path-planning methods include five-way flight and circular routes. Selecting an appropriate path while adjusting parameters such as image overlap, camera inclination, focal length, and exposure settings facilitates automated image acquisition by UAVs. However, traditional planning methods are not universally applicable. For instance, five-way flight is a common UAV flight path-planning method used in oblique aerial photography, where the drone captures images from five main directions (front, back, left, right, and top) to create a 3D model. While effective for large-scale modeling, it can result in data redundancy for single structures like bridge piers due to significant image overlap, unnecessary coverage of irrelevant areas, and potential visual blind spots that leave critical details missing. Conversely, circular flights are effective for modeling individual buildings and small structures but tend to create visual blind spots in areas such as the underside of bridge decks, leading to missing or incomplete reconstructions.

Although these methods enable rapid 3D reconstruction of bridges, the resulting models often lack detailed features, particularly in areas such as the pier bottoms, limiting their utility for comprehensive visualization. To achieve accurate 3D reconstruction during the construction phase, fine-tuned oblique photography route

planning is essential. The dung beetle optimization algorithm is employed to plan UAV flight paths around bridge piers in complex mountainous environments. By optimizing camera positions and determining the optimal flight trajectory, the algorithm identifies and marks critical feature points along the bridge pier contours. These points are then connected to create a 3D feature surface, ensuring that the UAV captures all key details and achieves complete coverage of the bridge piers. The UAV camera sensor size is $b \times h$, the focal length is d, the object distance is D, the heading distance S_1, and the side distance S_2. Then the expressions for the heading overlap rate ξ and the collateral overlap rate η are

$$\xi = 1 - \frac{S_1 \times d}{D \times b} \tag{15}$$

$$\eta = 1 - \frac{S_2 \times d}{D \times h} \tag{16}$$

Crack Identification

The crack detection method, leveraging machine learning and image-processing techniques, involves a series of preprocessing steps on the crack image (Mohan and Poobal 2018; Aliyari et al. 2022). These steps include image preprocessing, binarization, and support vector machine classification, among others, to generate a clear crack image and facilitate width calculation (Yang et al. 2018). The processing flow is illustrated in Fig. 5.

Image Preprocessing

Image preprocessing involves several steps, including gray-scale conversion, linear gray-scale transformation, filtering, and denoising. Specifically, gray-scale conversion refers to the process of transforming a color image into a gray-scale image. Given the high computational load associated with processing the 3D pixel matrix, which can impact processing speed, converting the color image to gray scale reduces the computational burden. The processing method is expressed by the following equation (Cao et al. 2017):

$$F(x, y) = 0.3R(x, y) + 0.59G(x, y) + 0.11B(x, y) \tag{17}$$

where $F(x, y)$ = the gray-scale value after gray-scale conversion of a pixel; and $R(x, y)$, $G(x, y)$, and $B(x, y)$ = the values of the R, G, and B components, respectively, for the pixel (x, y) in the color image.

Following the gray-scale process, issues such as uneven illumination and darker details within the image may persist. The application of linear grey-scale transformation addresses these issues, enhancing the visual appearance of the crack target and improving
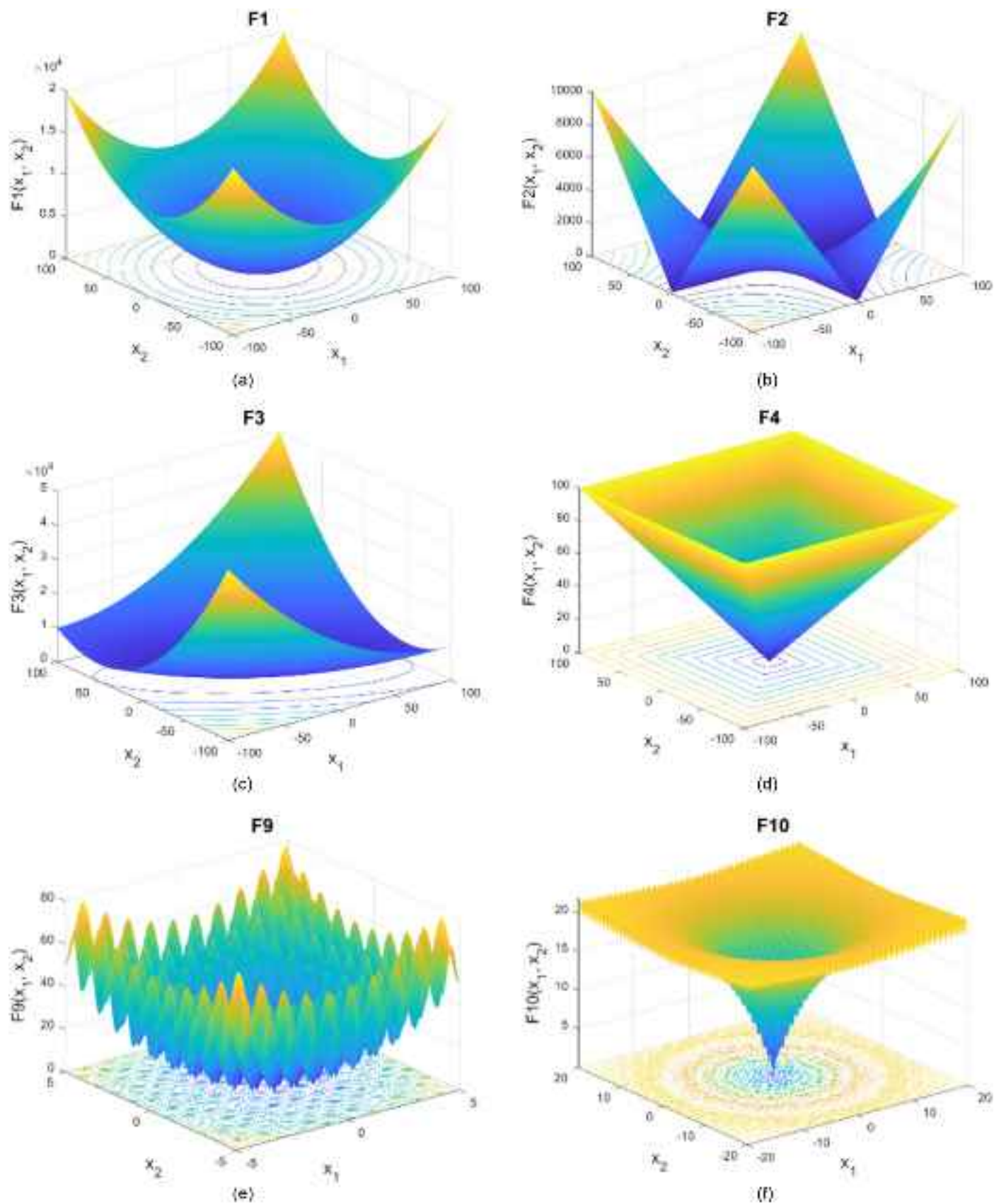
© ASCE   04025056-4   J. Perform. Constr. Facil.

J. Perform. Constr. Facil., 2025, 39(6): 04025056

Fig. 3. Benchmark function surface plot: (a) benchmark function surface plot of F1; (b) benchmark function surface plot of F2; (c) benchmark function surface plot of F3; (d) benchmark function surface plot of F4; (e) benchmark function surface plot of F9; and (f) benchmark function surface plot of F10.
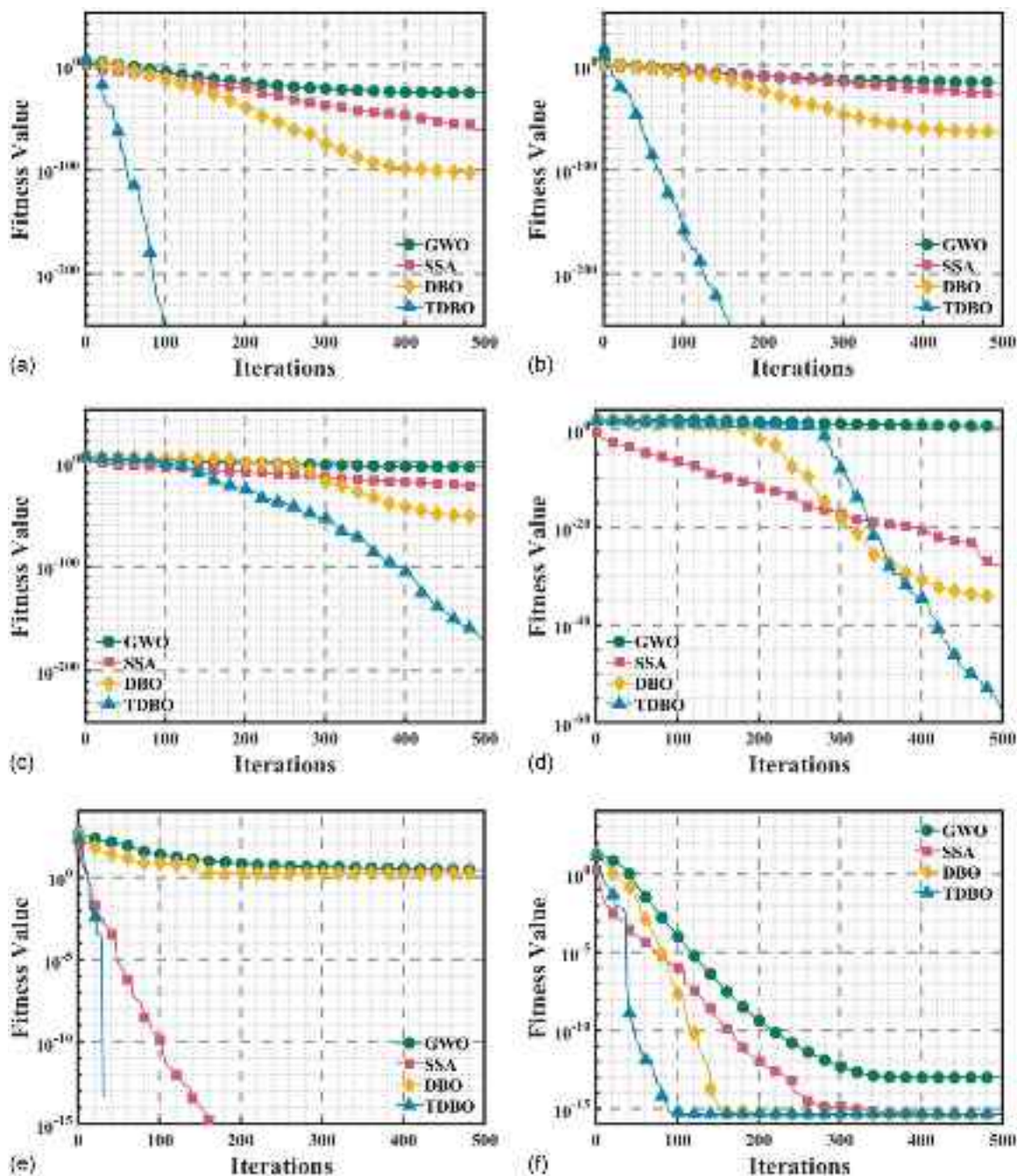
Fig. 4. Iterative convergence curves: (a) iterative convergence curves for F1; (b) iterative convergence curves for F2; (c) iterative convergence curves for F3; (d) iterative convergence curves for F4; (e) iterative convergence curves for F9; and (f) iterative convergence curves for F10.

the detail visibility. While image enhancement increases the overall contrast, it does not fully eliminate interference factors within the image (Li et al. 2019). Therefore, a filtering and denoising algorithm is required to remove these disturbances, thereby clarifying the crack outline.

Binarization is the conversion of an image into a binary format consisting of only black and white intensity values by applying a grey-scale threshold. A commonly used method is Otsu thresholding; however, this method is mainly effective for images that exhibit a distinct bimodal distribution (Talab et al. 2016). If the bimodal nature of the image is not so obvious, then it may leave obvious noise regions after segmentation. To address this situation, this paper proposes a dynamic denoising method based on connected domains, which determines whether the current connected domains

should be filtered or not by calculating the ratio of the noisy region to the cracked region.

Crack Identification and Segmentation

After the second denoising, most of the noise regions have been filtered out, but crack burrs remain, and some cracks are discontinuous. In this paper, a crack connection algorithm based on the KD-tree is used to identify the endpoints of adjacent cracks (Yusoff et al. 2023). By calculating the Euclidean distance between these endpoints and comparing it with a predefined threshold, the algorithm determines whether the endpoints should be connected. For three consecutive points, $P_{i-1}(x_{i-1}, y_{i-1})$, $P_i(x_i, y_i)$, $P_{i+1}(x_{i+1}, y_{i+1})$ on the crack path, the curvature $\Delta\kappa_i$ change is calculated as

© ASCE

04025056-6

J. Perform. Constr. Facil.

Table 2. Comparison of test function results

Basis function	Statistic	GWO	SSA	DBO	TDBO
F1	Best	3.36×10^{-29}	0.00×10^{0}	1.41×10^{-160}	0.00×10^{0}
	Mean	1.20×10^{-27}	4.9×10^{-54}	7.71×10^{-110}	0.00×10^{0}
	Std	2.03×10^{-27}	2.6×10^{-53}	4.22×10^{-109}	0.00×10^{0}
F2	Best	2.41×10^{-17}	0.00×10^{0}	4.54×10^{-81}	0.00×10^{0}
	Mean	9.37×10^{-17}	5.91×10^{-28}	3.33×10^{-54}	0.00×10^{0}
	Std	7.92×10^{-17}	2.38×10^{-27}	1.82×10^{-53}	0.00×10^{0}
F3	Best	2.73×10^{-8}	0.00×10^{0}	4.49×10^{-150}	0.00×10^{0}
	Mean	1.72×10^{-5}	6.63×10^{-30}	6.77×10^{-77}	2.73×10^{-160}
	Std	4.06×10^{-5}	3.63×10^{-29}	3.71×10^{-76}	1.49×10^{-159}
F4	Best	8.21×10^{-1}	2.65×10^{-256}	6.62×10^{-74}	0.00×10^{0}
	Mean	4.74×10	3.83×10^{-30}	3.23×10^{-39}	3.83×10^{-70}
	Std	2.67×10	1.27×10^{-29}	1.72×10^{-38}	2.05×10^{-69}
F9	Best	3.83×10^{-30}	2.72×10^{-256}	6.42×10^{-74}	0.00×10^{0}
	Mean	3.09×10^{0}	3.47×10^{-30}	3.37×10^{-39}	0.00×10^{0}
	Std	4.71×10^{0}	1.58×10^{-29}	1.54×10^{-38}	0.00×10^{0}
F10	Best	7.51×10^{-14}	4.37×10^{-16}	4.44×10^{-16}	4.52×10^{-17}
	Mean	9.91×10^{-14}	4.44×10^{-16}	5.63×10^{-16}	4.84×10^{-29}
	Std	1.22×10^{-14}	0.00×10^{0}	6.49×10^{-16}	0.00×10^{0}

$$\Delta \kappa_i = \left| \frac{(x_{i+1} - 2x_i + x_{i-1})(y_{i+1} - y_{i-1}) - (y_{i+1} - 2y_i + y_{i-1})(x_{i+1} - x_{i-1})}{[(x_{i+1} - x_{i-1})^2 + (y_{i+1} - y_{i-1})^2]^{3/2}} \right| \tag{18}$$

When the curvature change of adjacent points exceeds the design threshold (the default value is 0.15), it is judged as an abnormal inflection point, and the connection is disconnected. A support vector machine is employed for crack classification, which categorizes cracks into regular and irregular types (Ye et al. 2024; Cai et al. 2024). The classification of cracks into regular and irregular types is based on their geometric and spatial characteristics. Regular cracks, such as transverse and longitudinal cracks, have uniform shapes and predictable orientations, often resulting from consistent structural loading or material properties. In contrast, irregular cracks appear more random and may be influenced by localized factors like material defects or construction imperfections. For regular cracks, the length and maximum width are calculated. For irregular cracks,

the area fraction of the cracked region is computed. The identified crack images and their associated characteristic parameters are stored and exported to an Excel file.

The bridge surface crack detection system developed on the MATLAB platform integrates the operations mentioned here and consists of four core modules: (1) the file information module, which manages the basic attributes and metadata of the images; (2) the operation module, which implements functions for image preprocessing, edge detection, and feature extraction; (3) the result display module, which visualizes the processed crack images and recognition results; and (4) the data storage and export module, which stores the analysis results in Excel format and generates standardized analysis reports.
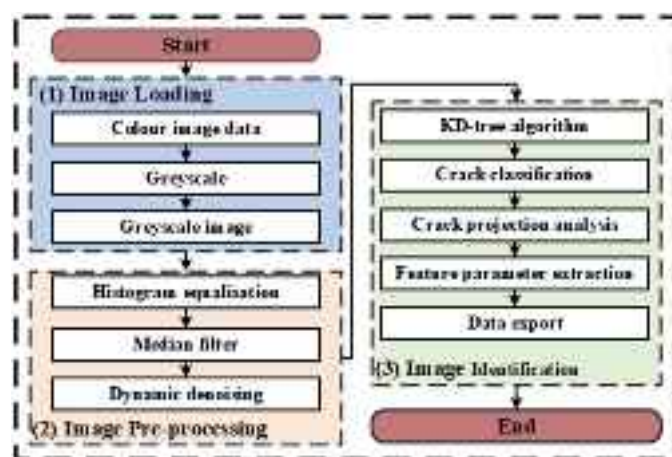
This paper presents a spatial alignment algorithm that combines mutual information with correspondence points to achieve accurate crack mapping (Jiang et al. 2023). The algorithm not only offers broad convergence and high alignment accuracy but also effectively filters out invalid alignment data. The objective function in the multidimensional minimization formula in Eq. (19) is optimized to enhance alignment precision

$$MC_{I_a I_B(C)} = r(-MI(I_A, I_B(C)) + (1 - r)E(\mathrm{Corr}, C)$$
$$C = (\theta_x, \theta_y, \theta_z, t_x, t_y, t_z, f)$$
$$0 \leq \mathrm{r} \leq 1 \tag{19}$$

where MC = the weighted sum of the minimization objective function of the algorithm; IA = the preprocessed image to be aligned; IB = the rendering of the 3D model; r = the contribution ratio for each item; C = the camera's intrinsic and extrinsic parameters; $(\theta_x, \theta_y, \theta_z$ and $t_x, t_y, t_z)$ = the rotation angles and translation vector of the camera in the world coordinate system, respectively; and f denotes the camera's focal length.

Fig. 5. Crack identification segmentation process.

© ASCE 04025056-7 J. Perform. Constr. Facil.

J. Perform. Constr. Facil., 2025, 39(6): 04025056

Under the condition where the camera is parameterized by C, the objective function based on mutual information and corresponding points is given by Eq. (20)

$$MI(I_A, I_B(C)) = -\sum_{a,b} p(a,b) \log\left(\frac{p(a,b)}{p(a)p(b)}\right) \qquad (20)$$

where MI = the shared information between image IA and image IB; $P(a,b)$ = the joint probability of event (a,b); $P(a)$ = the probability of a pixel value an in image IA; and $p(b)$ = the probability of a pixel value b in image IB

$$E(\text{Corr}, c) = \frac{1}{N} \sum_{\forall \text{cor}_i} \sqrt{(x_{pi(C)} - x_i)^2 + (y_{pi(C)} - y_i)^2} \qquad (21)$$

where E = the average pixel distance between the image points and the corresponding 3D model points projected onto the image plane; (x_i, y_i) = the corresponding point coordinates in the preprocessed aligned images; and x_{pi}, y_{pi} = the 2D coordinates of the projected corresponding points of the 3D model on a plane.

The texture coordinates of the vertices are linearly interpolated to compute the remaining texture coordinates (Huang et al. 2021). Using the texture mapping technique, the crack feature map is applied to the 3D model, resulting in the fusion of the crack feature image with the real 3D model. This process generates a realistic 3D model with highlighted crack features.

Practical Engineering Verification

To verify the effectiveness of the refined routes generated by the improved dung beetle optimization algorithms in the rapid modeling of bridges as well as the accuracy of the crack identification procedure, an applied study was conducted using the high pier of the Yuyang River Bridge as the research subject.

Engineering Background

The Yuyang River Bridge is located in Lizhuxicun, Wangjiafan Township, Yidu City, spanning the Yuyang River. It spans a typical V-shaped river valley with a cutting depth of approximately 183 m and a riverbed width of about 37 m. The bridge features steep terrain on both sides, with local cliffs and precipices. The pier construction includes multiple high piers, with the section between Pier 5 and Pier 6 being the focus of our detailed route planning and modeling. The terrain on both sides of the river is steep, with local cliffs and precipices. The reason we chose the under-construction Yueyang River Bridge over the completed bridge is that early crack detection

during construction allows for timely intervention to prevent escalation of the problem, save maintenance costs, and extend the service life and safety of the bridge. The construction phase is dynamic, and the structure is subject to a variety of loads and conditions, making it an ideal environment to study crack formation and validate UAV-based inspection methods. In addition, bridges under construction provide more detailed design and construction records, facilitating accurate assessments and providing better accessibility for UAV imaging. During the testing period, more than a dozen bridge piers were surveyed, and the section between Pier 5 and Pier 6 was selected for detailed route planning, as shown in Fig. 6.

Overall Route Planning

Planning an aerial image acquisition route for a UAV is the first step in the 3D reconstruction of a target building. Currently, commonly used methods for acquiring rough models include five-way flight routes and circular routes. Inclined photography five-way flight routes are particularly suited for large-area operations, making them ideal for establishing the rough model of a bridge abutment.

The image acquisition equipment used for the bridge is a DJI Mavic 3E drone coupled with a high-precision Leica camera. A five-way flight path is employed to establish a rough 3D model of the Yilai Expressway Yuyang River Bridge, which serves as the informational base for refined route planning between the UAV and the bridge. During the process of establishing the rough model, the UAV performs an oblique photography flight at a relative altitude of 50 m above the target bridge area to collect the necessary images.

The result of the rough 3D model of the Yuyang River Bridge is shown in Fig. 7. While the on-the-fly method allows for rapid modeling with a limited data set, the reconstruction of certain details remains suboptimal, resulting in a rough model. This limitation arises from the low pixel accuracy, which is influenced by the long shooting distance and the presence of visual blind spots. As illustrated in the zoomed-in portion of the image, textures on the bottom and sides of the bridge are missing, and the reconstruction of the abutments appears distorted and incomplete.

Fine Route Planning

This study focuses on the bridge between piers 5 and 6 of the Yuyanghe Bridge, using the DBO and TDBO algorithms to plan refined UAV flight paths. The accuracy and efficiency of these algorithms are compared and analyzed in relation to the five-direction flight paths.

The application of path optimization algorithms to the practical problem of 3D high-pier bridge modeling necessitates algorithm refinement. Marking feature points on the initial 3D model is crucial,
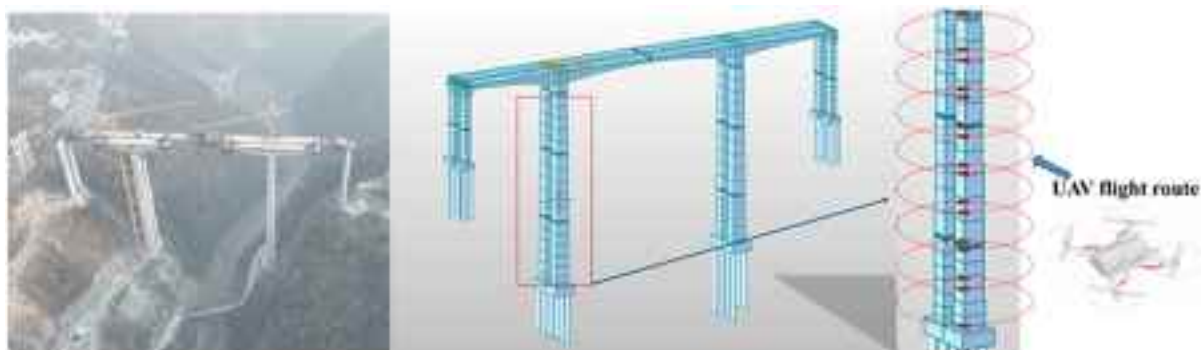

Fig. 6. Route planning area. (Image by authors.)

Fig. 7. Overall route planning. (Images by authors.)

as the number of effective feature points captured by the UAV directly impacts the heading overlap rate. By employing contour theory, the work area is divided into layers based on height values, and path planning is performed within each layer. Once the current layer is fully covered, the nodes directly above (or below) the layer are identified and used as the starting points for the next layer's path planning. This process is repeated until the entire 3D workspace is fully covered; the path-planning idea is shown in Fig. 8.

The KML file is a bridge for human–computer interaction, and the GIS software extracts the KML format terrain data of the target area and resolves the geographic coordinates and height information in it to construct a 3D terrain model. Under the MATLAB environment, two algorithms are used to perform segmented path planning. The geographic location information of each waypoint in the path planning, yaw angle, gimbal pitch angle, route speed, and waypoint tasks are compiled into a KML file and imported into the UAV, and the UAV can read the KML file, fly according to the preset parameters in the file, and perform the route tasks.

Model Comparison

The actual bridge test collected refined route images of the side and bottom of the bridge pier as well as the bridge abutment (Suleymanoglu et al. 2023). These images, along with tilt photography route images, were combined to generate a 3D model. The modeling results are compared, as shown in Fig. 9.

The rough model obtained from the reconstruction based on inclined routes is shown in Fig. 9(a), while the 3D model, reconstructed using the planning of inclined photography combined with the DBO algorithm, is shown in Fig. 9(b). The 3D model obtained from the planning of reconstruction based on inclined photography and the TDBO algorithm is shown in Fig. 9(c). The modeling quality of these three combinations has been compared. Regarding the

front texture of the bearing platform, Fig. 9(a) exhibits noticeable missing texture, whereas Figs. 9(b and c) show no missing texture, with both models demonstrating comparable completeness. However, the consistency of the model in Fig. 9(c) is superior. For the reconstruction of the lateral structure of the bearing platform, Fig. 9(a) shows significant distortions and missing areas in the model. In Fig. 9(b), the 3D reconstruction is nearly free of missing data, though some surface distortions remain. Fig. 9(c) provides the most accurate model, exhibiting greater overall completeness and better details. A comparison of the modeling efficiency of the inclined photographic routes and refined routes is provided in Table 3.

By comparing the modeling outcomes of different route planning methods, it can be observed that, although refined routes require the collection of additional image data, leading to an increase in both external and internal work time, their 3D reconstruction is more accurate, with fewer missing and distorted features. In contrast, the traditional oblique photography route, while more efficient in terms of time consumption for external and internal tasks and capable of quickly generating the bridge pier model, suffers from lower accuracy. This is due to the limited number of images, which can create visual blind spots, especially at the bottom of the bridge pier and other areas. Consequently, the resulting 3D model has a lower accuracy. Based on the experimental data, the tilt photography + TDBO route offers a better balance between modeling accuracy and time efficiency, making it a more optimal choice for efficiency and precision.

Since the quality of the model created using the five-way route flight is relatively poor, it will not be included in the accuracy comparison. Five elevation feature points were selected as check points at the bridge construction site, located at the top of the pier, the middle of the pier, the bottom of the pier, the bearing platform, and the foundation. The coordinates of these points were measured using GPS RTK technology, and the results were considered the reference values. The coordinates of the corresponding points on the 3D model were then extracted as the calculated values. The models generated by fusing the five-way aerial photography with the DBO or TDBO routes are designated as Model 1 and Model 2, respectively. The accuracy errors between these two models are then compared to assess the modeling performance. The error calculation formula is as follows:

$$m_x = \sqrt{\frac{\sum (\Delta x)^2}{n}} \qquad (22)$$

$$m_y = \sqrt{\frac{\sum (\Delta y)^2}{n}} \qquad (23)$$

$$m_z = \sqrt{\frac{\sum (\Delta z)^2}{n}} \qquad (24)$$
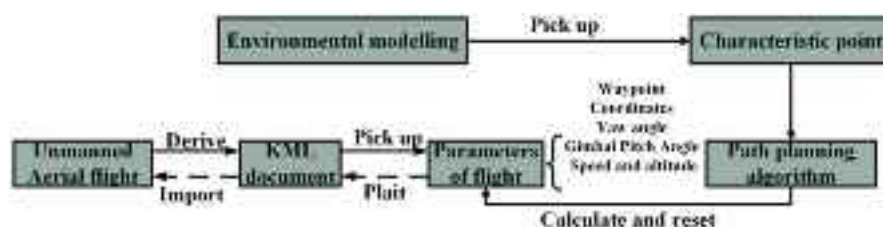
$$m = \sqrt{(m_x)^2 + (m_y)^2} \qquad (25)$$

Fig. 8. Ideas for route planning.

Fig. 9. Effect of modeling multiple combinations of data: (a) five-direction aerial photography; (b) five-direction aerial photography +DBO; and (c) five-direction aerial photography +TDBO. (Images by authors.)

Table 3. Comparison of the efficiency of the three route modeling approaches

Route planning	Number of images	Time-consuming external work (min)	Time-consuming industry work (h)
Five-direction aerial photography	299	16	1
Five-direction aerial photography +DBO	1,362	77	4
Five-direction aerial photography +TDBO	1,036	63	3

Table 4. Statistics of 3D model accuracy

Checkpoint number	Model 1 plane error (m)	Model 1 elevation error (m)	Model 2 plane error (m)	Model 2 elevation error (m)
Top of the piers	0.082	0.235	0.054	0.162
Middle of the pier	0.067	0.174	0.046	0.128
Bottom of the piers	0.095	0.213	0.059	0.151
Abutment	0.078	0.198	0.048	0.139
Infrastructural	0.091	0.256	0.052	0.178

where mx, my, mz = the error of the model in the x, y, and z plane, respectively.

The statistics yielded the following results, as shown in Table 4 and Fig. 10. From the data, it can be concluded that the accuracy of Model 2 is significantly higher than that of Model 1. In terms of planimetric error, the maximum error is reduced from 0.095 to 0.059 m, and the median error from 0.079 to 0.051 m. In terms of elevation error, the maximum error is reduced from 0.256 to 0.178 m, and the median error from 0.208 to 0.147 m. The accuracy of Model 2 is significantly higher than that of Model 1.

The UAV's refined route primarily focuses on capturing images of the outer and lower surfaces of the bridge abutment. Consequently,

the UAV's path should cover as many feature points as possible. The KML files generated by the two algorithms in the same environment are analyzed, with path length, path planning time, and the number of traversed feature points serving as the performance metrics. The results are presented in Table 5.

An analysis of UAV flight parameters in the KML file reveals that the object distance DDD of the TDBO algorithm is smaller compared to that of the DBO algorithm. This reduction in object distance effectively increases the heading overlap rate and lateral overlap rate. The enhanced overlap rate directly contributes to improved data redundancy, which subsequently enhances the completeness and accuracy of the collected data. These improvements ultimately lead to a more complete and faithful 3D reconstruction model. It can be concluded that the TDBO algorithm not only
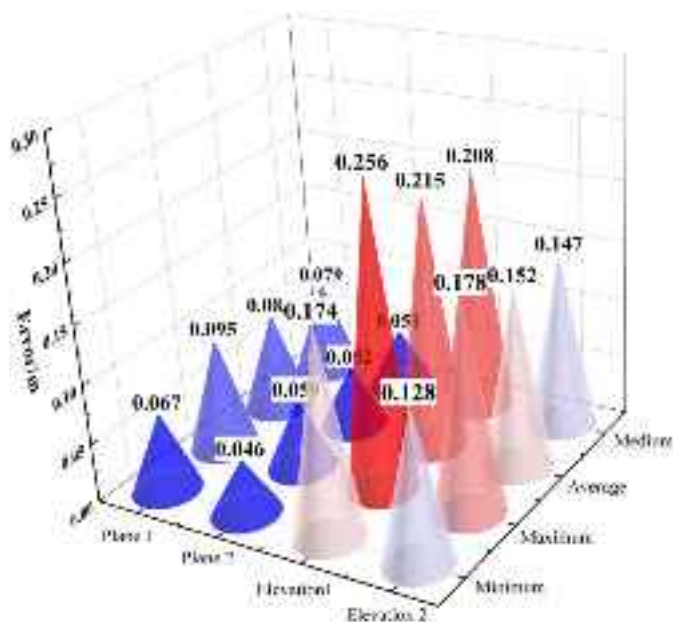
© ASCE 04025056-10 J. Perform. Constr. Facil.

J. Perform. Constr. Facil., 2025, 39(6): 04025056

Fig. 10. 3D model error statistics.

Table 5. Performance comparison of different path planning algorithms

Targets	DBO	TDBO
Path length (m)	3,043	2,596
Route planning time (s)	243	198
Number of feature points	89	114

Table 6. Crack identification results

Type of crack	Precision	Recall	F1-score
Transverse crack	91.58	91.23	91.74
Longitudinal cracks	90.34	92.54	91.43
Other cracks	90.33	87.32	88.80
Totally	90.12	91.08	90.60

shortens the path length and accelerates planning speed but also significantly improves the coverage of feature points. Moreover, even in the absence of ground control points, the algorithm ensures that the overlap rate satisfies the requirements of high-precision photogrammetry.

Crack Identification Results

To evaluate the crack recognition performance, common evaluation metrics in segmentation tasks are introduced, such as precision, recall, and F1-score (Fu et al. 2021). Further, these formulas are provided as follows:

$$\text{Precision} \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{26}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{27}$$

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{28}$$

The crack identification performance is summarized in Table 6. In this section, the proposed model is processed for training and prediction. The SDNET2018 data set is used for training to evaluate the performance of the crack recognition algorithm and is supplemented by the inclusion of images of cracks captured in real projects. Cross-validation is used to evaluate the performance of the model during the training process, and metrics such as accuracy, recall, and F1-score are calculated. The batch size was set to 32, the number of training calendar elements was set to 150, and the network model was optimized using an adaptive learning rate optimizer. In addition, an early stopping strategy was implemented to address the overfitting problem and to ensure the generalization ability of the model.

The algorithm presented in this paper selects typical cracks from the captured images for detection, with some results shown in Fig. 11. These results demonstrate that the proposed crack segmentation algorithm can effectively identify and segment the cracks in their entirety.

To verify the accuracy of the crack width calculation method, this paper detects the crack widths in the captured images and measures the actual crack widths using a crack width meter (with an accuracy of 0.01 mm). The results in Table 7 indicate that the calculated crack widths are closer to the measured values, with the relative error between the maximum and average widths of the detection results being less than 10%. This confirms the effectiveness of the crack width detection method proposed in this paper.

The improved dung beetle optimization algorithm is employed to reconstruct the model, while the correlation information and homonymous point matching algorithm are used to align corresponding feature points between the image and the model. Additionally, the texture mapping technique is applied to create 3D mapping of the
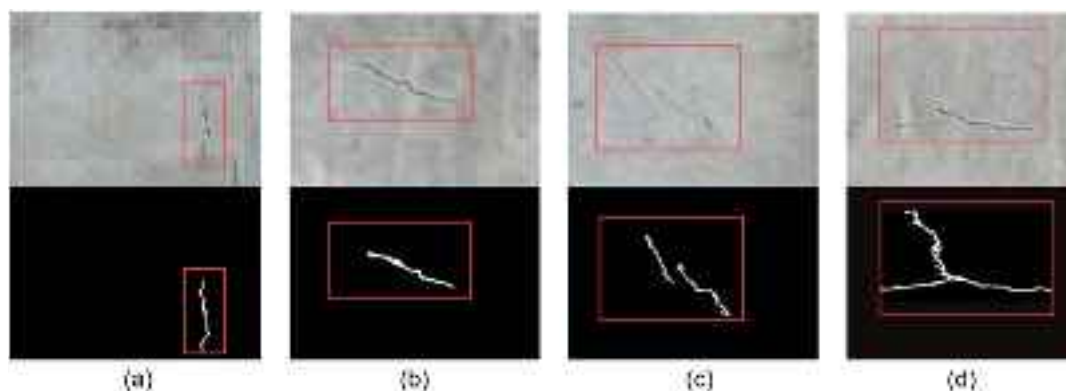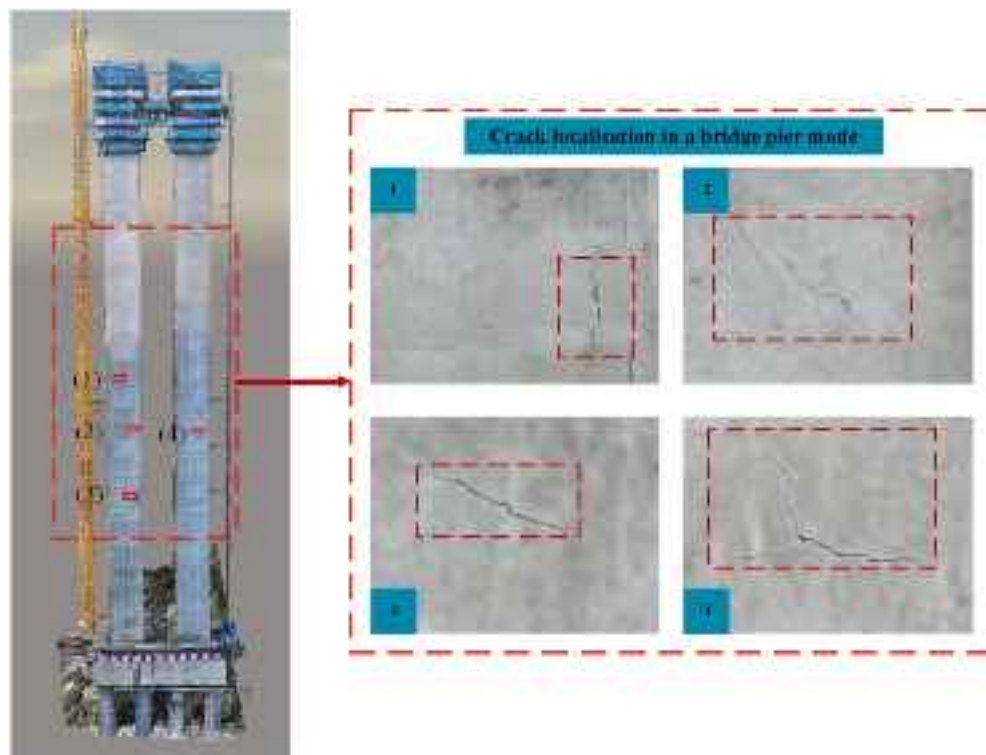

Fig. 11. Results of crack identification.

Table 7. Crack width detection results

Picture number	Measured width (mm)		Calculate width (mm)		Absolute error (mm)		Relative error (%)	
	Maximum values	Average value	Maximum values	Average value	Maximum values	Average value	Maximum values	Average value
1	1.16	0.39	1.187	0.418	0.037	0.038	2.36	7.38
2	1.28	0.37	1.326	0.399	0.056	0.039	3.63	8.07
3	0.97	0.47	1.008	0.492	0.048	0.032	3.97	4.79
4	0.79	0.41	0.777	0.389	0.023	0.031	1.68	5.26

Fig. 12. 3D model and crack evaluation of the pier. (Image by authors.)

cracks. The resulting 3D model of the high pier, which incorporates crack information, is shown in Fig. 12.

Conclusion

This paper addresses the technical challenges of crack detection and path planning for long-span bridges in complex mountainous environments by proposing an integrated solution that combines an improved dung beetle optimization algorithm with multidimensional image information. Experimental validation yields the following conclusions:

1. The optimization performance of the algorithm is significantly enhanced by employing chaotic mapping, adaptive weights, and spiral search strategies. Comparative analysis with other algorithms reveals that the improved algorithm achieves higher fitness values in multimodal and unimodal benchmark function tests, effectively mitigating the issue of local optima trapping.
2. Based on the acquired five-directional flight path image data, the application of the TDBO algorithm to supplement the flight path enables observation from multiple heights and angles, effectively reducing visual blind spots. This process not only

enhances the model's completeness and texture fineness but also significantly improves the model's refinement.
3. Through the comparative analysis of 3D model accuracy, the study found that the maximum error in the plane direction of the pier detail model decreased from 0.095 to 0.059 m, and the average error reduced from 0.079 to 0.051 m; in the elevation direction, the maximum error decreased from 0.256 to 0.178 m, and the average error reduced from 0.208 to 0.147 m. Using the improved dung beetle algorithm for modeling significantly enhanced the accuracy of the model in the plane and elevation directions, demonstrating the effectiveness of the optimized path-planning strategy in improving model quality in complex terrains.
4. The study of the UAV crack identification program and the quantitative analysis of cracks accurately located the cracks. Additionally, the results were compared with the actual measurements. The program validated that the accuracy of crack identification exceeded 90%, with the relative error of the maximum and average widths being less than 10%, meeting the requirements for bridge defect detection.

Future research could focus on optimizing dynamic environmental adaptability and improving real-time path-planning algorithms

© ASCE 04025056-12 J. Perform. Constr. Facil.

J. Perform. Constr. Facil., 2025, 39(6): 04025056

to enhance the performance of UAVs under different climatic conditions. Meanwhile, enhancing the generalization capability of the crack identification algorithm could expand its application to more complex bridge structures.

Data Availability Statement

Some or all data, models, or codes that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments

Author Contributions

Xinfeng Yin: Funding acquisition; Investigation. Xin Fu: Writing – original draft; Writing – review and editing. Yang Quan: Supervision; Validation. Tao Chen: Software. Zhou Huang: Conceptualization; Data curation.

References

Adhikari, R. S., O. Moselhi, and A. Bagchi. 2014. "Image-based retrieval of concrete crack properties for bridge inspection." *Autom. Constr.* 39 (Apr): 180–194. https://doi.org/10.1016/j.autcon.2013.06.011.

Aliyari, M., B. Ashrafi, and Y. Z. Ayele. 2022. "Hazards identification and risk assessment for UAV-assisted bridge inspections." *Struct. Infrastruct. Eng.* 18 (3): 412–428. https://doi.org/10.1080/15732479.2020.1858878.

Bu, T., J. Zhu, and T. Ma. 2022. "A UAV photography-based detection method for defective road marking." *J. Perform. Constr. Facil.* 36 (5): 04022035. https://doi.org/10.1061/(ASCE)CF.1943-5509.0001748.

Caballero-Martin, D., J. M. Lopez-Guede, J. Estevez, and M. Graña. 2024. "Artificial intelligence applied to drone control: A state of the art." *Drones* 8 (7): 296. https://doi.org/10.3390/drones8070296.

Cai, R., J. Li, Y. Tan, W. Shou, and A. Butera. 2024. "Automated geometric quantification of building exterior wall cracks based on computer vision." *J. Perform. Constr. Facil.* 38 (4): 04024015. https://doi.org/10.1061/JPCFEV.CFENG-4618.

Cao, L., L. Jiao, Z. Li, T. Liu, and Y. Zhong. 2017. "Grayscale image colorization using an adaptive weighted average method." *J. Imaging Sci. Technol.* 61 (6): 1–10. https://doi.org/10.2352/J.ImagingSci.Technol.2017.61.6.060502.

Chen, W., J. Hou, Y. Wang, and M. Yu. 2024a. "Visualization analysis of concrete crack detection in civil engineering infrastructure based on knowledge graph." *Case Stud. Constr. Mater.* 21 (Dec): e03711. https://doi.org/10.1016/j.cscm.2024.e03711.

Chen, X., C. Wang, C. Liu, X. Zhu, Y. Zhang, T. Luo, and J. Zhang. 2024b. "Autonomous crack detection for mountainous roads using UAV inspection system." *Sensors* 24 (14): 4751. https://doi.org/10.3390/s24144751.

Cheng, X., J. Li, C. Zheng, J. Zhang, and M. Zhao. 2021. "An improved PSO-GWO algorithm with chaos and adaptive inertial weight for robot path planning." *Front. Neurorobot.* 15 (Nov): 770361. https://doi.org/10.3389/fnbot.2021.770361.

Congress, S. S. C., A. J. Puppala, M. A. Khan, N. Biswas, and P. Kumar. 2022. "Application of unmanned aerial technologies for inspecting pavement and bridge infrastructure assets conditions." *Transp. Res. Rec.* 2679 (1): 529–543. https://doi.org/10.1177/03611981221105273.

Duque, L., J. Seo, and J. Wacker. 2018. "Synthesis of unmanned aerial vehicle applications for infrastructures." *J. Perform. Constr. Facil.* 32 (4): 04018046. https://doi.org/10.1061/(ASCE)CF.1943-5509.0001185.

Fang, Z. X., and A. V. Savkin. 2024. "Strategies for optimized UAV surveillance in various tasks and scenarios: A review." *Drones* 8 (5): 193. https://doi.org/10.3390/drones8050193.

Fu, H., D. Meng, W. Li, and Y. Wang. 2021. "Bridge crack semantic segmentation based on improved deeplabv3+." *J. Mar. Sci. Eng.* 9 (6): 671. https://doi.org/10.3390/jmse9060671.

Huang, S. F., C. Y. Lin, and D. C. Cai. 2021. "Enhancing the learning effect of virtual reality 3D modeling: A new model of learners design collaboration and a comparison of its field system usability." *Univers. Access Inf. Soc.* 20 (3): 429–440. https://doi.org/10.1007/s10209-020-00750-7.

Huang, T., K. Fan, and W. Sun. 2024. "Density gradient-RRT: An improved rapidly exploring random tree algorithm for UAV path planning." *Expert Syst. Appl.* 252 (Oct): 124121. https://doi.org/10.1016/j.eswa.2024.124121.

Huang, Z., X. F. Yin, Y. Quan, Y. Liu, and P. Xiang. 2025. "Stochastic vibration prediction of long-span bridges under traffic load based on deep neural network of multi-mode information fusion." *Adv. Eng. Software* 207 (Sep): 103953. https://doi.org/10.1016/j.advengsoft.2025.103953.

Jeong, E., J. Seo, and J. Wacker. 2020. "Literature review and technical survey on bridge inspection using unmanned aerial vehicles." *J. Perform. Constr. Facil.* 34 (6): 04020113. https://doi.org/10.1061/(asce)cf.1943-5509.0001519.

Jiang, S., S. Gu, and Z. Yan. 2023. "Pavement crack measurement based on aerial 3D reconstruction and learning-based segmentation method." *Meas. Sci. Technol.* 34 (1): 015801. https://doi.org/10.1088/1361-6501/ac8e22.

Jiang, S., Y. Zhang, F. Wang, and Y. Xu. 2025. "Three-dimensional reconstruction and damage localization of bridge undersides based on close-range photography using UAV." *Meas. Sci. Technol.* 36 (1): 015423. https://doi.org/10.1088/1361-6501/ad90fb.

Li, C., J. Liu, A. Liu, Q. Wu, and L. Bi. 2019. "Global and adaptive contrast enhancement for low illumination gray images." *IEEE Access* 7 (Nov): 163395–163411. https://doi.org/10.1109/ACCESS.2019.2952545.

Li, G., Q. Liu, S. Zhao, W. Qiao, and X. Ren. 2020. "Automatic crack recognition for concrete bridges using a fully convolutional neural network and naive Bayes data fusion based on a visual detection system." *Meas. Sci. Technol.* 31 (7): 075403. https://doi.org/10.1088/1361-6501/ab79c8.

Li, L., S. Xu, H. Nie, Y. Mao, and S. Yu. 2021. "Collaborative target search algorithm for UAV based on chaotic disturbance pigeon-inspired optimization." *Appl. Sci.* 11 (16): 7358. https://doi.org/10.3390/app11167358.

Liu, B., T. Yang, X. Wu, B. Wang, H. Zhang, and Y. Wu. 2024. "UAV imagery-based railroad station building inspection using hybrid learning architecture." *Meas. Sci. Technol.* 35 (8): 086206. https://doi.org/10.1088/1361-6501/ad4ab5.

Liu, H., X. Li, M. Fan, G. Wu, W. Pedrycz, and P. N. Suganthan. 2022. "An autonomous path planning method for unmanned aerial vehicle based on a tangent intersection and target guidance strategy." *IEEE Trans. Intell. Transp. Syst.* 23 (4): 3061–3073. https://doi.org/10.1109/TITS.2020.3030444.

Mohan, A., and S. Poobal. 2018. "Crack detection using image processing: A critical review and analysis." *Alexandria Eng. J.* 57 (2): 787–798. https://doi.org/10.1016/j.aej.2017.01.020.

Moysis, L., K. Rajagopal, A. V. Tutueva, C. Volos, B. Teka, and D. N. Butusov. 2021. "Chaotic path planning for 3D area coverage using a pseudo-random bit generator from a 1D chaotic map." *Mathematics* 9 (15): 1821. https://doi.org/10.3390/math9151821.

Namian, M., M. Khalid, G. Wang, and Y. Turkan. 2021. "Revealing safety risks of unmanned aerial vehicles in construction." *Transp. Res. Rec.* 2675 (11): 334–347. https://doi.org/10.1177/03611981211017134.

Pohan, M. A. R., B. R. Trilaksono, S. P. Santosa, and A. S. Rohman. 2024. "Path planning using combined informed rapidly-exploring random tree star and particle swarm optimization algorithms." *IEEE Access* 12 (Apr): 56582–56608. https://doi.org/10.1109/ACCESS.2024.3389152.

Suleymanoglu, B., M. Gurturk, Y. Yilmaz, A. Soycan, and M. Soycan. 2023. "Comparison of unmanned aerial vehicle-LiDAR and image-based mobile mapping system for assessing road geometry parameters via digital terrain models." *Transp. Res. Rec.* 2677 (8): 617–632. https://doi.org/10.1177/03611981231157730.

Talab, A. M. A., Z. Huang, F. Xi, and L. HaiMing. 2016. "Detection crack in image using Otsu method and multiple filtering in image processing

techniques." *Optik* 127 (3): 1030–1033. https://doi.org/10.1016/j.ijleo.2015.09.147.

Thai, H. D., C. W. Yoon, and J. H. Huh. 2024. "Recent development of drone technology software engineering: A systematic survey." *IEEE Access* 12: 128729–128751. https://doi.org/10.1109/ACCESS.2024.3454546.

Wang, F., Y. Zou, E. del Rey Castillo, Y. Ding, Z. Xu, H. Zhao, and J. B. Lim. 2024. "Automated UAV path-planning for high-quality photogrammetric 3D bridge reconstruction." *Struct. Infrastruct. Eng.* 20 (10): 1595–1614. https://doi.org/10.1080/15732479.2022.2152840.

Won, J., J. W. Park, C. Shim, and M. W. Park. 2021. "Bridge-surface panoramic-image generation for automated bridge-inspection using deep-matching." *Struct. Health Monit. Int.* 20 (4): 1689–1703. https://doi.org/10.1177/1475921720930380.

Wu, Y., M. Abdel-Aty, O. Zheng, Q. Cai, and S. Zhang. 2020. "Automated safety diagnosis based on unmanned aerial vehicle video and deep learning algorithm." *Transp. Res. Rec.* 2674 (8): 350–359. https://doi.org/10.1177/0361198120925808.

Xie, J., L. R. G. Carrillo, and L. Jin. 2020. "Path planning for UAV to cover multiple separated convex polygonal regions." *IEEE Access* 8 (Mar): 51770–51785. https://doi.org/10.1109/ACCESS.2020.2980203.

Xue, J., and B. Shen. 2023. "Dung beetle optimizer: A new meta-heuristic algorithm for global optimization." *J. Supercomput.* 79 (7): 7305–7336. https://doi.org/10.1007/s11227-022-04959-6.

Yan, W. L., X. F. Yin, Y. Liu, K. Tuohuti, L. S. Wu, and Y. Liu. 2025. "Bridge damage detection based on vehicle scanning method and parallel convolutional neural network." *Measurement* 245 (Mar): 116563. https://doi.org/10.1016/j.measurement.2024.116563.

Yang, S. T., K. F. Li, and C. Q. Li. 2018. "Numerical determination of concrete crack width for corrosion-affected concrete structures." *Comput. Struct.* 207 (Sep): 75–82. https://doi.org/10.1016/j.compstruc.2017.07.016.

Ye, Y. F., K. L. Ji, and P. Wang. 2024. "Identification of rail cracks based on path graph features and SVM." *Adv. Mech. Eng.* 16 (5): 16878132241252228. https://doi.org/10.1177/16878132241252228.

Yin, X. F., Z. Huang, and Y. Liu. 2023. "Bridge damage identification under the moving vehicle loads based on the method of physics-guided deep neural networks." *Mech. Syst. Signal Process.* 190 (May): 110123. https://doi.org/10.1016/j.ymssp.2023.110123.

Yuan, Q., Y. Shi, and M. Li. 2024. "A review of computer vision-based crack detection methods in civil infrastructure: Progress and challenges." *Remote Sens.* 16 (16): 2910. https://doi.org/10.3390/rs16162910.

Yusoff, I. N., M. A. M. Ismail, H. Tobe, T. Miyoshi, K. Date, and Y. Yokota. 2023. "Discontinuity pattern detection and orientation measurement for tunnel faces by using structure from motion photogrammetry." *Displays* 76 (Jan): 102356. https://doi.org/10.1016/j.displa.2022.102356.

Zhang, C., Y. Zou, F. Wang, E. del Rey Castillo, J. Dimyadi, and L. Chen. 2022. "Towards fully automated unmanned aerial vehicle-enabled bridge inspection: Where are we at?" *Constr. Build. Mater.* 347 (Sep): 128543. https://doi.org/10.1016/j.conbuildmat.2022.128543.

Zhang, X. B., J. Wang, Z. Z. Chen, Y. Quan, Z. Z. Zheng, T. Y. Zhang, J. Cao, X. N. Xie, X. C. Liu, and P. Xiang. 2025. "A distributed fiber optic sensor-based approach for crack asphalt structure under freeze-thaw cycling tests." *Constr. Build. Mater.* 476 (May): 141262. https://doi.org/10.1016/j.conbuildmat.2025.141262.

Zhao, Y. X., B. H. Lu, and M. Alipour. 2024. "Optimized structural inspection path planning for automated unmanned aerial systems." *Autom. Constr.* 168 (Dec): 105764. https://doi.org/10.1016/j.autcon.2024.105764.

Zhou, L., et al. 2024. "UAV vision-based crack quantification and visualization of bridges: System design and engineering application." *Struct. Health Monit.* 24 (2): 1083–1100. https://doi.org/10.1177/14759217241251778.

J. Perform. Constr. Facil.