



Helix-HPSO approach for UAV path planning in a multi-building environment

Shiwei Lin¹ · Xiaoying Kong² · Jianguo Wang¹ · Ang Liu¹

Received: 10 February 2022 / Accepted: 16 November 2022 / Published online: 24 November 2022
© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2022

Abstract

Regular inspection of historic buildings is essential, while path planning of the building inspection is challenging because it requires comprehensive coverage at a low cost. Most of the previous research does not consider the multiple buildings' environment. In this paper, a three-dimensional path planning approach is proposed to provide the inspection for multiple buildings. The proposed Helix-HPSO approach generates the helix-shaped path for each building and uses HPSO for path planning between buildings. The computational experiment validates the proposed approach. The helix-shaped path costs less than the traditional back-and-forth path for building inspection. HPSO is compared with other bio-inspired algorithms for optimization problems and PSO for path planning.

Keywords Multi-building · 3D path planning · UAV

1 Introduction

Unmanned Aerial Vehicles (UAVs) can provide remote inspections, and obtain different digital imagery by sensors or cameras, such as thermal, ultrasonic, laser scanners, high-resolution, and near-infrared [1,2]. Computer vision and other technologies can process the collected data to detect surface defects of infrastructure, including distortion, spalling, cracking, excessive movements, rusting, and misalignment [2]. Finding an informative and efficient path is required to perform UAV-based path planning inspection with data from different views [2].

This research was conducted because of an accident in Notre Dame de Paris, which attracted attention to protect the historical buildings and collect the building data in case of an accident. Historic buildings have historical value as symbols of specific eras and can exhibit the architect's aesthetic and past people's lives. Monitoring historic buildings' condition and regular inspections of historic buildings are essential.

The Helix-HPSO approach is proposed as UAV-based path planning for building inspection of multiple historical buildings. It uses the helix-shaped path for inspecting each building, which is a smooth path at a low cost and suitable for UAV constraints. Also, the Helix path is compared with the traditional inspection path. The proposed HPSO algorithm generates the path for flying to another building, considering the distance and collision avoidance. The HPSO has been compared with other bio-inspired algorithms through benchmark functions and with PSO for path planning. The contributions of this paper include the following:

- Improve PSO with the inspiration from the Harmony search algorithm as a new algorithm, HPSO
- A novel 3D path planning approach based on the helix-shaped path and proposed HPSO for a smooth and safe path
- Provide the generic building inspection approach in the multi-building environment

✉ Shiwei Lin
Shiwei.Lin-1@student.uts.edu.au

Xiaoying Kong
Xkong@mit.edu.au

Jianguo Wang
Jianguo.Wang@uts.edu.au

Ang Liu
Ang.Liu@student.uts.edu.au

¹ Faculty of Engineering and Information Technology, University of Technology Sydney, Broadway, Sydney, NSW 2007, Australia

² School of IT and Engineering, Melbourne Institute of Technology, Sydney Campus, Sydney, NSW 2000, Australia

This paper proposes a UAV-based path planning for a building inspection as the Helix-HPSO approach, and it is organized as follows. Section 2 reviews the UAV path planning algorithms and building inspection approaches. Section 3 describes the problem formulation, and Sect. 4 proposes the Helix-HPSO approach. The result of the computational experiment is in Sect. 5. This paper is concluded in Sect. 6.

2 Related work

The path planning algorithms can be classified as node-based [3], sampling-based [4], mathematical mode [5], multi-fusion-based [1] and bio-inspired algorithms [2]. A* and DWA is integrated for global path optimization for UAV path planning to improve safety, and efficiency [3]. A genetic algorithm and A* algorithm are integrated to solve the travelling salesman problem for LiDAR-equipped UAV path planning [6]. An improved A* is combined with a gravitational search algorithm for the optimal path with several optimization objectives, such as the return point and the heading angle [7].

Moreover, for UAV path planning, a biased sampling potentially guided intelligent bidirectional RRT* algorithm is proposed to overcome the slow convergence rate of exploration [4]. UAV path planning can be transformed into a nonlinear optimal control problem, and sequential convex programming is proposed to solve it by approximating the non-convex parts [5]. An improved Artificial Potential field (APF) is proposed for UAV path planning with additional control force, and it avoids local minima [8]. The chaotic bat algorithm is combined with an improved APF for faster position update and adaptive inertia weight [9].

An odd-even layered genetic algorithm (GA) is proposed for cooperative inspection path planning in [10]. Particle swarm optimization (PSO) and the mutation operator of GA are combined to improve the central force optimization algorithm and aim for complicated path optimization [11]. In a proposed random chemical reaction optimization algorithm [12], PSO and elitist selections are combined to act as the subsets of the set of molecules. It is used for UAV inspection path planning.

Additionally, a heuristic evolutionary algorithm in [13] implements mutation, substitution, crossover, smoothness and length operations to build the path for UAV. The reinforcement learning is improved based on the grey wolf optimizer algorithm to control the switch operations for individuals during UAV path planning [14]. The differential evolution algorithm is modified with symbiotic organism search for impressive local and global search ability [15]. A modified Mayfly algorithm is proposed in [16] with an enhanced crossover operator, adaptive Cauchy mutation and

an exponent decreasing inertia weight to search the configuration space and get the path with the lowest cost for UAV path planning.

UAVs offer excellent flexibility in many fields with the development of technologies and remote photogrammetry, including security, rural environment monitoring, urban planning, and, recently, infrastructure inspection [17–19]. UAV applications in construction include site surveying, safety inspection, building inspection, damage assessment, urban monitoring, road assessment, geo-hazard investigations, and progress monitoring [20,21].

UAV infrastructure inspections can be classified into five categories: bridges, power lines, buildings and facades, railways and sewers, and geographical inspections [17]. A regular building inspection is necessary to ensure their condition is safe. Maintenance and detection of building facade anomalies require periodic safety inspections and examinations [22]. They are more critical to the historical buildings as heritage [19].

Detail three-dimensional heritage reconstruction is crucial for interpretation, further analysis and physical reconstruction eventually [23]. Heritage documentation includes post-catastrophe damage assessment and the modelling of facades, monuments or entire buildings [23]. Inspectors usually take photos directly of every element in general practice to record damage or dilapidation, which is time-consuming and expensive [1,19]. UAV technology overcomes risky, time-consuming, and costly inspection practices for monitoring and inspecting infrastructure [24].

Additionally, the research of building inspection is for evaluating the condition of a building and checking the appearance of a building. The UAV survey aims to model the structure and recognize the state [25]. UAV-based photogrammetry evaluates the condition of facilities and allows complete documentation of buildings with fewer human resources and time, particularly for the areas difficult to access [2,19,26]. A UAV-based project's geometric result relies on the UAV manufacturing quality and the onboard sensor system, materials, shutter modes and digital sensor types [23].

The autonomous navigation of UAVs relies on local information in the GNSS-denied zones, and an RGB-D Kinect camera or visual information can obtain such local information to compute an analytical path [27]. Distance measuring units can also calculate the angle and the range between the structure and the UAV during the inspection [28]. The pre-processing algorithm processes the local information obtained from the point cloud and then uses the path planning algorithm to generate the path for the indoor environment [27].

Range-based and image-based techniques are employed widely in the three-dimensional documentation of buildings, and UAVs have advantages for image-based techniques

with aerial views [23]. UAVs follow a programmed flight path to record and survey historical buildings, obtaining survey data for unreachable areas with a terrestrial platform [2, 18, 29]. UAVs are equipped with sensors for navigation, 3D data acquisition, and obstacle avoidance [1, 2, 18]. It achieves higher flexibility with low-cost [30]. Survey data can vary from sensors, such as Forward-Looking Infrared technology, images, separate strips, multi-attributed point cloud, 3D point cloud, and laser scanners [1, 19, 30, 31]. UAV-derived point clouds can extract the surface after image processing [32]. The building information model and the UAV inspection workflow are integrated for an augmented reality solution with aerial video [33, 34].

Laser scanning data, close-range photogrammetric images, or a combination of both are generally applied for building inspection documentation [2, 26]. Close-range facade images are employed to inspect, and document facade anomalies, such as corrosion and cracks [22]. 2D photographs generated by UAVs enable the creation of a precise model of buildings with proper path planning and using dedicated software to identify the defects [26]. Close-range high-resolution inspections can achieve 3D modelling of a historical building [29]. Documentation of heritage buildings with 3D reconstruction and photogrammetry is based on motion with a dense matching algorithm and optical sensors [23]. LiDAR, stereo camera, IMU, and wide-angle camera are implemented for the UAV-UGV system for indoor and cluttered scenes [35].

Moreover, a GIS-based two-step procedure supports building facade inspection by processing the management of UAV-collected images [22] in the inspection, measuring the geometry of buildings and requiring documentation in the form of photogrammetric images [18, 36]. Advanced photogrammetric techniques and deep learning algorithms are applied to record building damages autonomously by a true-orthophoto [36]. A Convolutional Neural Network is fine-tuned to detect damage and surface cracks [37, 38]. Ultrasonic Beacon system replaces GPS for autonomous flight, and a geotagging method can locate damages [37].

For buildings with planar surfaces, particle swarm optimization is adapted as enhancing discrete particle swarm optimization to plan the path for UAV [1]. The most common path planning method for a building inspection is back-and-forth paths. The inspection of the Perak Museum used back-and-forth paths; the UAV flew from the bottom of the building, then moved to the top, and flew to the right side, then reached the ground [19].

From the literature, the popular algorithms used in UAV path planning are based on bio-inspired algorithms. The bio-inspired algorithms provide fast path generation with optimal solutions. For the building inspection, the related work lacks the consideration of multiple buildings. This paper proposes the Helix-HPSO approach to address the problem of the

multi-building environment, generating the path for each building and between the buildings.

3 Problem formulation

The basic components of UAV path planning include the start, the goal and the environment. The shape of most buildings can be modelled as rectangular or cylinder. The building inspection requires broad coverage with a short path. For proceeding between buildings, collision avoidance and flight distance should be considered. It is aimed at generating a safe and optimal path. The developed cost functions to evaluate the paths are as below.

3.1 Cost functions for building inspection

The cost function is defined as Eq. (1), and it is to evaluate the inspection paths.

$$f_{cost} = w_1 * f_{distance} + w_2 * f_{time} + w_3 * f_{altitude} \quad (1)$$

where w_1 , w_2 and w_3 are the weights of the cost functions of distance, flight time, and the change of altitude. The cost functions are listed as Eqs. (2)–(4). The sum of the weights is 1.

$$f_{distance} = \sum_{k=1}^n \sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2 + (z_{k+1} - z_k)^2} \quad (2)$$

where n is the size of the path points, and x_k is the current path point.

$$f_{altitude} = \sum_{k=1}^n \sqrt{(z_{k+1} - z_k)^2} \quad (3)$$

where $\sqrt{(z_{k+1} - z_k)^2}$ is the change of altitude.

$$f_{time} = \sum_{k=1}^n \frac{\sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2 + (z_{k+1} - z_k)^2}}{v} \quad (4)$$

where v is the current velocity.

3.2 Cost functions for path planning

The cost functions are developed to evaluate the optimal path during path planning, considering the distance and collision avoidance. The best solution would be the generated path for

flying to another building.

$$f_{path\ cost} = w_4 * f_{length} + w_5 * f_{collision} \tag{5}$$

where w_4 and w_5 are the weights of the length and collision cost functions. The cost functions are listed as Equations (6) – (9). The sum of the weights is 1.

$$f_{length} = \sum_{k=1}^n \sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2 + (z_{k+1} - z_k)^2} \tag{6}$$

where n is the number of path points, and x_k is the current path point.

$$d_k = \sqrt{(x_k - x_c)^2 + (y_k - y_c)^2 + (z_k - z_c)^2} \tag{7}$$

where d_k is the distance from the current path point to the centre of an obstacle.

$$c_k = \sum_{c=1}^{cn} r_c - d_k \tag{8}$$

where cn is the number of obstacles. The collision occurs if $r_c - d_k < 0$, which means that the current path point overlaps

with an obstacle c .

$$f_{collision} = \sum_{k=1}^n c_k \tag{9}$$

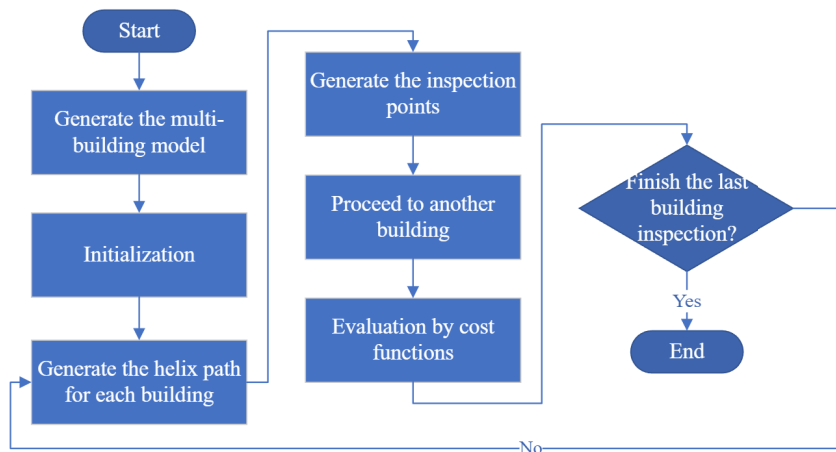
where $f_{collision}$ sums up the violation value for collisions that happen between each path point and obstacles.

4 Helix-HPSO approach

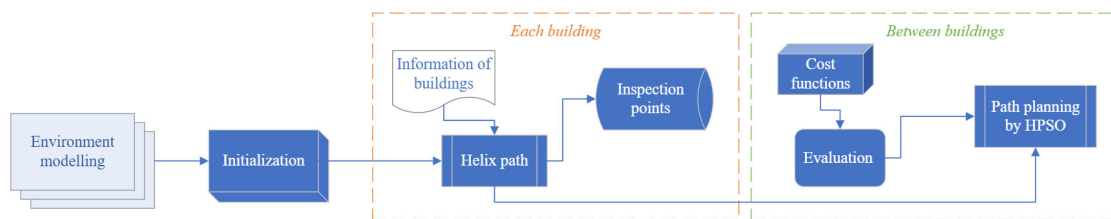
4.1 Overview

The helix-HPSO approach generates the helix path for each building, defining the points where info is collected with reasonable time slots. After inspecting one building, the UAV proceeds to another building with an optimal path based on a visited vector and a cost matrix, as shown in Fig. 1a. Cost functions evaluate several factors, including distance, time, and altitude. The helix-HPSO approach is demonstrated in Fig. 1b. The steps of the proposed algorithm are as follows:

1. Build the model for the multi-building environment.
2. Generate the inspection path and the points where info is collected for each building.



(a) The flow chart of the Helix-cost approach



(b) The principle of the Helix cost approach

Fig. 1 The Helix-cost

3. Proceed to another building after inspecting one building.

4.2 Path planning for each building: Helix path

4.2.1 Helix path

The helix path [39] for inspecting each building is generated through Eqs. (10)–(12). The position of the UAV is denoted as $(x(t), y(t), z(t))$ at the specific time t . The time slot is set to 0.01 for t . Recording current t as t_{max} and terminating the iteration once $z(t)$ reaches the determined height based on the inspection type.

$$x(t) = r * \sin\left(\frac{vt}{2\pi r}\right) + x_g \tag{10}$$

$$y(t) = r * \cos\left(\frac{vt}{2\pi r}\right) + y_g \tag{11}$$

$$z(t) = \left(\frac{vh}{2\pi r}\right) * t + z_g + 1 \tag{12}$$

where v is the UAV flight speed, (x_g, y_g, z_g) represents the centre of the ground floor of a building, r is the radius of the helix path, and h is the height of each floor.

The time required for an n -floor building inspection is defined by Eq. (13), with a flight speed v .

$$T_s = \frac{2\pi r n}{v} \tag{13}$$

The points of the path are stored as Eq. (14).

$$Points = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{bmatrix} \tag{14}$$

Algorithm 1 Generation Of Inspection Path

```

1: procedure HELIX PATH( $x_g, y_g, z_g, floorNum, radiusOfBuilding,$ 
    $heightOfEachFloor, heightOfRoof, inspectionType$ )
2:   if  $inspectionType == "interior"$  then
3:      $h \leftarrow heightOfEachFloorForInterior$ 
4:      $r \leftarrow radiusOfBuildingForInterior$ 
5:      $tMax \leftarrow getTmaxForInterior()$ 
6:   else
7:      $h \leftarrow heightOfEachFloorForExterior$ 
8:      $r \leftarrow radiusOfBuildingForExterior$ 
9:      $tMax \leftarrow getTmaxForExterior()$ 
10:  end if
11:  for  $time \leftarrow 0 : 0.01 : tMax$  do
12:     $xt \leftarrow r * \sin(v * time / 2 * pi * r) + x_g$ 
13:     $yt \leftarrow r * \cos(v * time / 2 * pi * r) + y_g$ 
14:     $zt \leftarrow (v * h) / (2 * pi * r) * time + z_g + 1$ 
15:     $points \leftarrow [points; xt, yt, zt]$ 
16:  end for
17: end procedure

```

Algorithm 1 presents the algorithm for generating the paths of interior and exterior inspection. The exterior and interior inspection paths are connected through the door, inspecting the entrance hall.

4.3 Path planning between buildings: HPSO

4.3.1 Cost matrix

After completing the inspection of one building, the UAV proceeds to another building for inspection. Each building is marked with numbers, and the cost matrix evaluates the order of inspecting buildings as Eq. (15). If a building has been visited, update the visited vector and the related costs as *infinity*.

$$Cost = \begin{bmatrix} infinity & cost(1, 2) & \dots & cost(1, m) \\ cost(2, 1) & infinity & \dots & cost(2, m) \\ \vdots & \vdots & \ddots & \vdots \\ cost(m, 1) & cost(m, 2) & \dots & infinity \end{bmatrix} \tag{15}$$

where m stands for the number of buildings in the environment.

4.3.2 Preliminary knowledge

Particle swarm optimization (PSO) Particle swarm optimization (PSO) is proposed by [40] and is inspired by the social behaviour of fish schooling or bird flocking. It uses the globally and local-oriented values to track each particle's coordinates in hyperspace. PSO uses initial random solutions and updates the velocity and position for each particle as Equations (16)–(17) to seek the optimal solutions in the configuration space. Algorithm 2 shows the principle of PSO.

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (pbest_i^t - x_i^t) + c_2 r_2 (gbest^t - x_i^t) \tag{16}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{17}$$

where v stands for particle velocity and ω is the inertial weight. $t + 1$ is the current time slot, and i is the current particle. $pbest$ is the best local-oriented value, and $gbest$ is the best global-oriented value. x is the position of the particle.

Harmony search (HS) Harmony search (HS) is proposed by mimicking the improvisation of music players [41]. It aims to get better solutions but with fewer iterations than other heuristic algorithms. Music harmony consists of several sound waves with different frequencies, and performances seek the best state. HS initializes a harmony memory (HM). Set the harmony memory considering rate (HMCR), harmony memory size (HMS), and pitching adjusting rate (PAR). Then, improvise a new harmony from HM as Eqs.

Algorithm 2 PSO algorithm

```

1: procedure PSO(x,v)
2:   Initialization
3:   for t ← 1 : tMax do
4:     for i ← 1 : number Of Particle do
5:       updateVelocity
6:       updatePosition
7:       Evaluation
8:       if cost(particleit) ≤ particlei.pbest then
9:         particlei.pbest ← particleit
10:      end if
11:      if cost(particleit) ≤ gbest then
12:        gbest ← particlei
13:      end if
14:    end for
15:  end for
16: end procedure
    
```

(18)–(20). Add the better harmony in HM and exclude the old harmony. Keep seeking harmony with the minimum harmony until satisfying the stopping criterion. HS is described by Algorithm 3.

$$x'_i = \begin{cases} x_i \in \{x_i^1, x_i^2, \dots, x_i^{HMS}\}, & \text{with probability HMCR} \\ x_i \in X_i, & \text{with probability (1-HMCR)} \end{cases} \tag{18}$$

where HMCR is the probability of getting one historic value in the HM

$$\begin{aligned} & \text{Pitch adjusting decision of } x'_i \\ & = \begin{cases} YES, & \text{with probability PAR} \\ NO, & \text{with probability (1-PAR)} \end{cases} \end{aligned} \tag{19}$$

If a value is from the HM, the pitching process happens. If the decision is YES, then updating x'_i as Eq. (20).

$$x'_i = x_i + \alpha \tag{20}$$

where α can be the result of the arbitrary distance bandwidth multiplying a uniform distribution in $[-1, 1]$ or the arbitrary distance bandwidth multiplying the standard normal distribution.

4.4 Harmony particle swarm optimization (HPSO)

From Sect. 2, The bio-inspired algorithms are widely used in UAV path planning, especially PSO and GA. Due to the requirement of fast computation, PSO is chosen to be improved for UAV path planning between buildings. HS is proposed for better solutions with fewer iterations, inspiring the improvement of the PSO approach. Thus, a hybrid HPSO is proposed for fast convergence and fewer iterations. The

Algorithm 3 HS algorithm

```

1: procedure HS(VarMin, VarMax, VarSize)
2:   Initialization
3:   for it ← 1 : itMax do
4:     for k ← 1 : nNew do
5:       xinew ← unifrnd(VarMin, VarMax, VarSize)
6:       for j ← 1 : nVar do
7:         if rand < HMCR then
8:           xinew ← xifromHM
9:           if rand < PAR then
10:            xinew ← xifromHM + α
11:          end if
12:        end if
13:      end for
14:      Evaluation
15:      Merge HM
16:      Update Best Solution
17:    end for
18:  end for
19: end procedure
    
```

inertial weight in HPSO is adjusted based on Eqs. (21)–(23).

$$\alpha = BW + randn() \tag{21}$$

where BW is the arbitrary distance bandwidth and $randn()$ generates a Gaussian distribution.

$$\omega = \omega + \alpha \tag{22}$$

where ω is the inertial weight of HPSO, the adjust processing only occurs when the current cost is larger than the personal best value, and the probability is less than PAR.

$$\omega = \omega * damping\ ratio \tag{23}$$

where $damping\ ratio$ is the ratio of updating the inertial weight.

Algorithm 4 demonstrates the proposed HPSO algorithm. The algorithm initializes the velocities and positions of the particles. Then update the positions and velocities ad Eqs. (16)–(17) and evaluate the solution by the cost function. If the current cost is less than the personal best value, it becomes the local best-oriented value. Otherwise, adjusting inertial weight for the algorithm based on the probability of PAR and α is calculated by the bandwidth multiplying the standard normal distribution. The proposed algorithm allows better solutions with fewer iterations and adjusts the inertial weight randomly for better search abilities. The global best value keeps updating until the termination criteria reach.

Algorithm 4 HPSO algorithm

```

1: procedure HPSO(v,x,damping,PAR,BW)
2:   Initialization
3:   for t ← 1 : tMax do
4:     for i ← 1 : N do
5:       updateVelocity
6:       updatePosition
7:       Evaluation
8:       if cost(particleit) ≤ particlei.pbest then
9:         particlei.pbest ← particleit
10:      else
11:        Alpha ← BW * randn()
12:        if rand ≤ PAR then
13:          w ← w + Alpha
14:        end if
15:      end if
16:      if cost(particleit) ≤ gbest then
17:        gbest ← particleit
18:      end if
19:    end for
20:    w ← w * damping
21:    BW ← BW * damping
22:  end for
23: end procedure
    
```

5 Computational experiment

5.1 Performance measurements

The proposed HPSO was compared with other bio-inspired methods, such as PSO, HS, GA [42] and Firefly algorithm (FA) [43]. The test functions are listed in Table 1 to compare the performance of the proposed algorithm with other bio-inspired algorithms. Each test function is a typical benchmark function with different characteristics. Ackley function has many local minima. The shapes of Rosenbrock and Sphere functions are like valley and bowl, respectively. The shape of Michalewicz function is a steep ridge. The number of the maximum iteration was 200.

Table 2 lists the mean iteration times when getting the optimal solution, the runtime of each algorithm, and fitness values for each benchmark function with different algorithms. Each algorithm has been run 20 times. The best value is highlighted

in bold. The proposed HPSO had the best performance for Ackley, which has many local minima. HPSO had minimal iteration times for the remaining test functions, while PSO had minimal runtime. Nevertheless, the difference between the runtime of PSO and HPSO was insignificant. It can be concluded that HPSO has great performance compared with other algorithms, especially for problems with many local minima. Getting rid of local optima and fast convergence are important for path planning, so HPSO is suitable for UAV path planning.

Figure 2 presents the test functions’ mean fitness value’s convergence curve. Figure 2 and Table 2 show that the proposed C-PSO converges faster than other bio-inspired algorithms. It has much fewer iteration times to get the optimization solutions, which is useful for path planning.

5.2 Scenario: Fujian Tulou

5.2.1 Modelling

A Tulou is an enclosed, large and fortified earthen building, most commonly circular or rectangular in configuration. It has thick, load-bearing, and rammed earthen walls, housing up to 800 people. It represents the living style from Song and Yuan Dynasty, around C.E. 960 to C.E. 1368. Figure 3a displays the photo of the Tianluokeng Tulou cluster comprising five main buildings, and Fig. 3b presents the model with the marked number of each building.

5.2.2 Simulation results

MATLAB validates the simulation of the path planning. The radius and the centre coordinate are inputted to generate the inspection path. The assumptions are as below:

- The speed is 18 m/s, the speed of ascent is 6 m/s, and the descent speed is 4 m/s.
- The UAV hovers when the UAV reaches the point of generating inspection data.

Table 1 Test functions

Name	Type	Test function
Ackley	Many Local Minima	$f_1(x) = -a \exp(-b \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}) - \exp(\sqrt{\frac{1}{d} \sum_{i=1}^d \cos(cx_i)}) + a + \exp(1)$
Rosenbrock	Valley-Shaped	$f_2(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
Sphere	Bowl-Shaped	$f_3(x) = \sum_{i=1}^d x_i^2$
Michalewicz	Steep Ridges/Drops	$f_4(x) = - \sum_{i=1}^d \sin(x_i) \sin^{2m}(\frac{ix_i^2}{\pi})$

Table 2 Test functions

Function		HPSO	PSO	HS	GA	FA
Ackley	Iterations	127.6	177.25	179.35	199.6	191.25
	Runtime (s)	0.007485	0.008081	0.106657	0.275348	1.008479
	Fitness value	0.00	0.00	0.01	0.00	0.00
Rosenbrock	Iterations	1	1	190.15	199.1	197.5
	Runtime (s)	0.005447	0.003763	0.107351	0.329114	0.970046
	Fitness value	0.00	0.00	4.05	7.86	0.40
Sphere	Iterations	102.8	192.4	174.6	199.1	187.45
	Runtime (s)	0.0057645	0.0045109	0.103878	0.2901371	0.9958323
	Fitness value	0.00	0.00	0.00	0.00	0.00
Michalewicz	Iterations	1	1	174.3	192.75	188.55
	Runtime (s)	0.005332	0.003449	0.115909	0.335478	1.070354
	Fitness value	-0.86	-0.84	-4.87	-9.31	-4.41

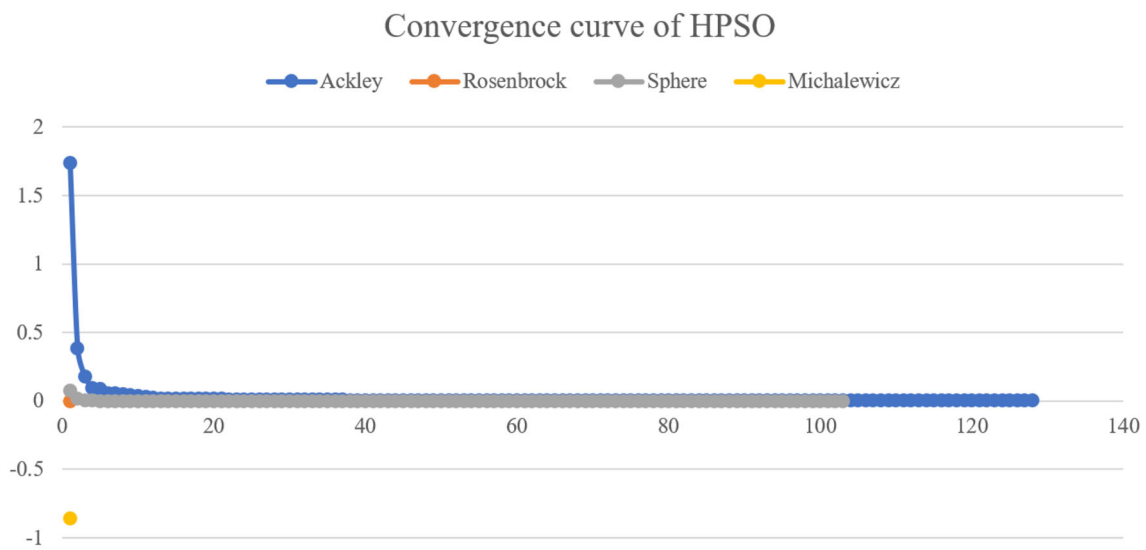
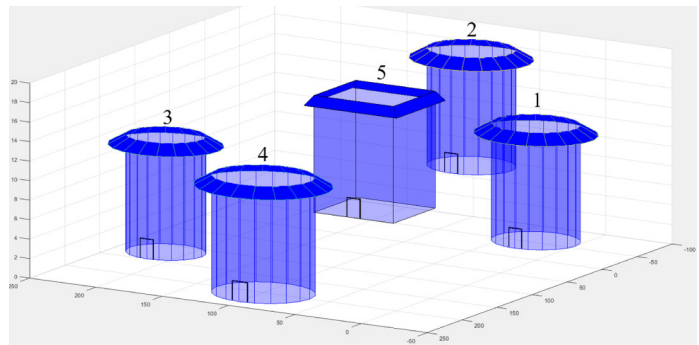


Fig. 2 Convergence curve of HPSO



(a) The Tianluokeng Tulou cluster

(b) Model of the cluster

Fig. 3 The Fujian Tulou

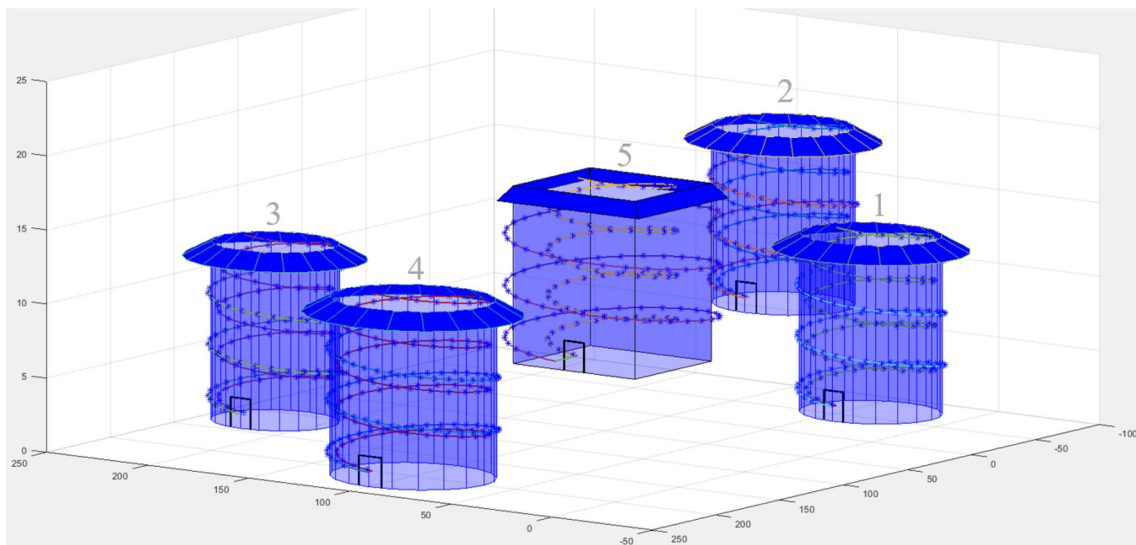


Fig. 4 The flight path for each building

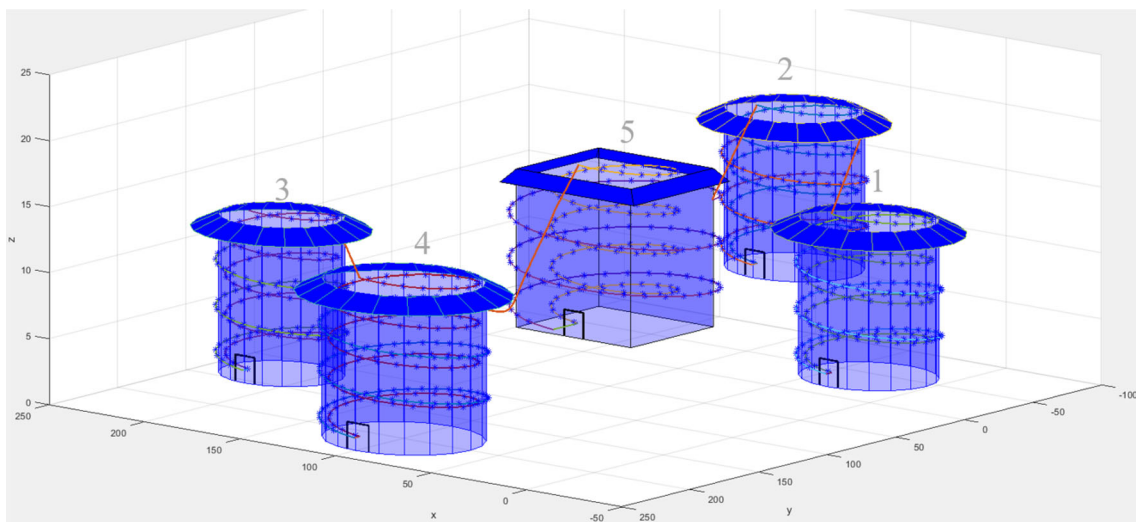


Fig. 5 The Helix-HPSO path

Table 3 Details of the Helix-HPSO path

Steps	Description	Path	Start	Destination
1	1 (From exterior to interior)	Building 1 inspection	(−29.97, −7.932, 12.48)	(0.196, 27, 15)
2	From 1 (interior) to 2 (exterior)	Proceed to Building 2	(0.196, 27, 15)	(44.74, −56.73, 17.5)
3	2 (From exterior to interior)	Building 2 inspection	(44.74, −56.73, 17.5)	(75.2, −23, 20)
4	From 2 (interior) to 5 (exterior)	Proceed to Building 5	(75.2, −23, 20)	(46.07, 50.23, 15.5)
6	5 (From exterior to interior)	Building 5 inspection	(46.07, 50.23, 15.5)	(89.66, 87, 17.99)
7	From 5 (interior) to 4 (exterior)	Proceed to Building 4	(89.66, 87, 17.99)	(65.29, 190.4, 10.48)
8	4 (From exterior to interior)	Building 4 inspection	(65.29, 190.4, 10.48)	(100.2, 232, 13)
9	From 4 (interior) to 3 (exterior)	Proceed to Building 3	(100.2, 232, 13)	(173.2, 141.8, 11.46)
10	3 (From exterior to interior)	Building 3 inspection	(173.2, 141.8, 11.46)	(200.2, 174, 14)

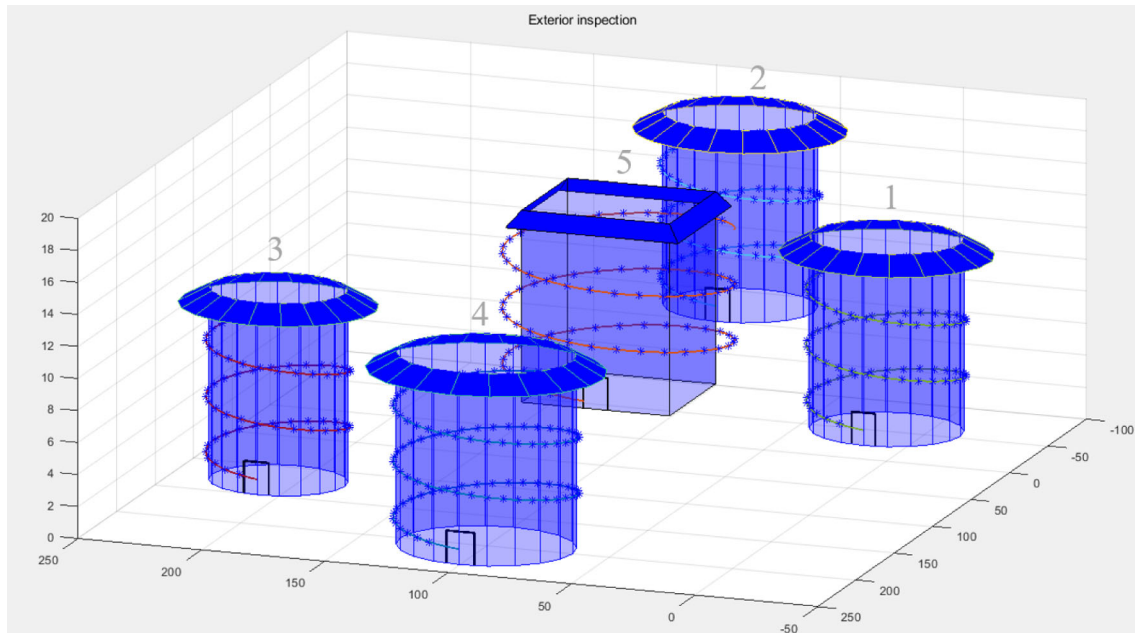
- UAV has been charged fully and operates in excellent condition.

Figure 4 demonstrates the flight path of each building, with the points where info is collected marked by "*". The interior and exterior inspection paths are connected through the hall, and it provides the possibility to check the condition of the entrance hall.

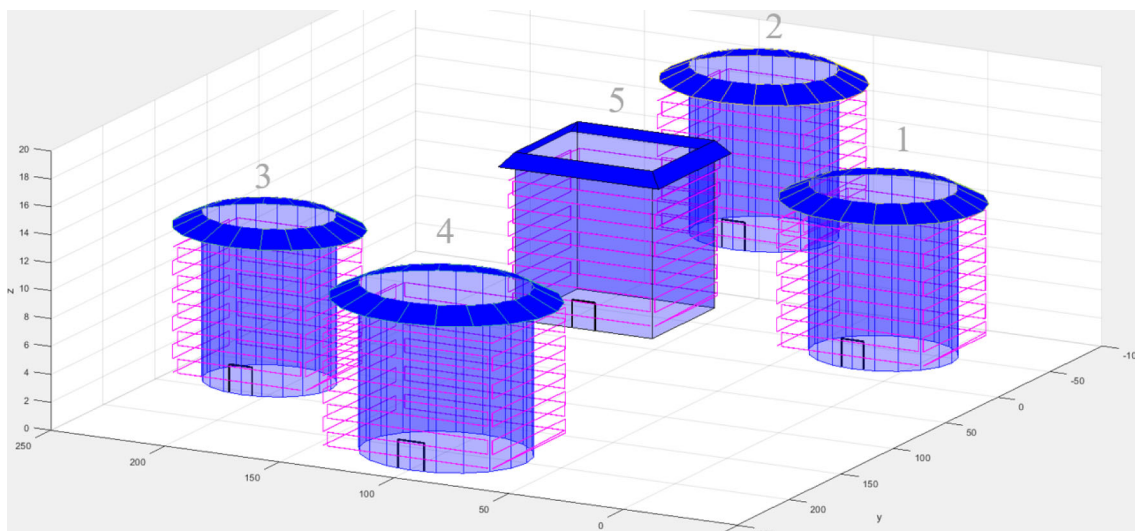
The start building is Building 1. From the visited vector and the cost matrix, the order of inspecting the buildings is

1 2 5 4 3, which is determined by Dijkstra's algorithm. The start is the exterior path of Building 1, which is close to the top.

The Helix-HPSO approach integrates the flight path; the entire flight path is shown in Figs. 5 and 7a. The Helix-HPSO path starts from the exterior inspection path of Building 1, following the interior path inspection path. After reaching the end of the interior path, the UAV flies to the next building defined in the order. After getting to Building 2, the UAV inspects the exterior and interior areas. The loops of inspec-



(a) The exterior inspection paths by Helix-HPSO

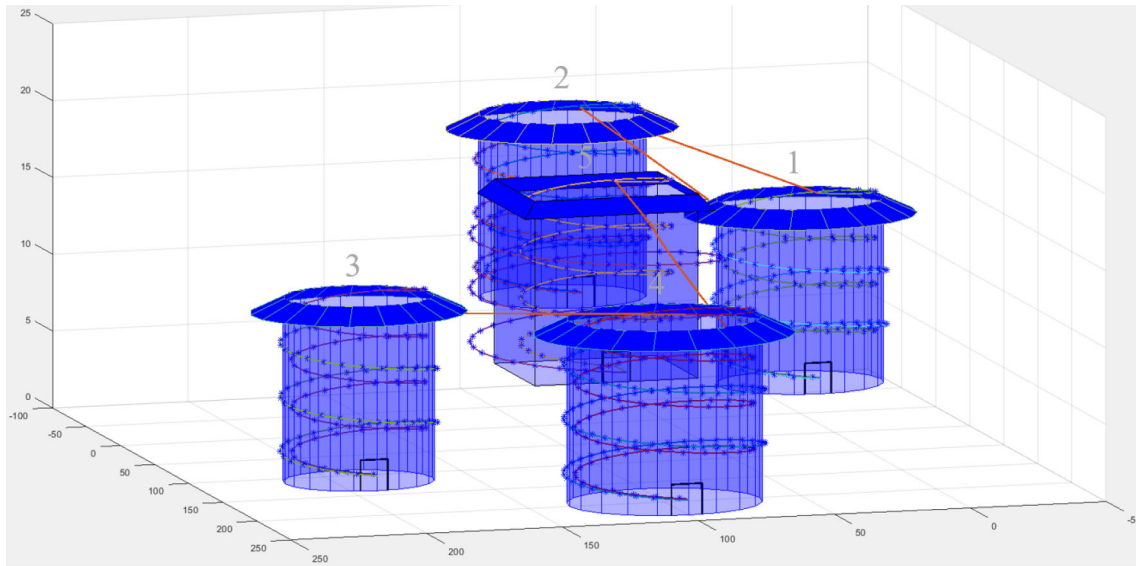


(b) The back-and-forth for each building

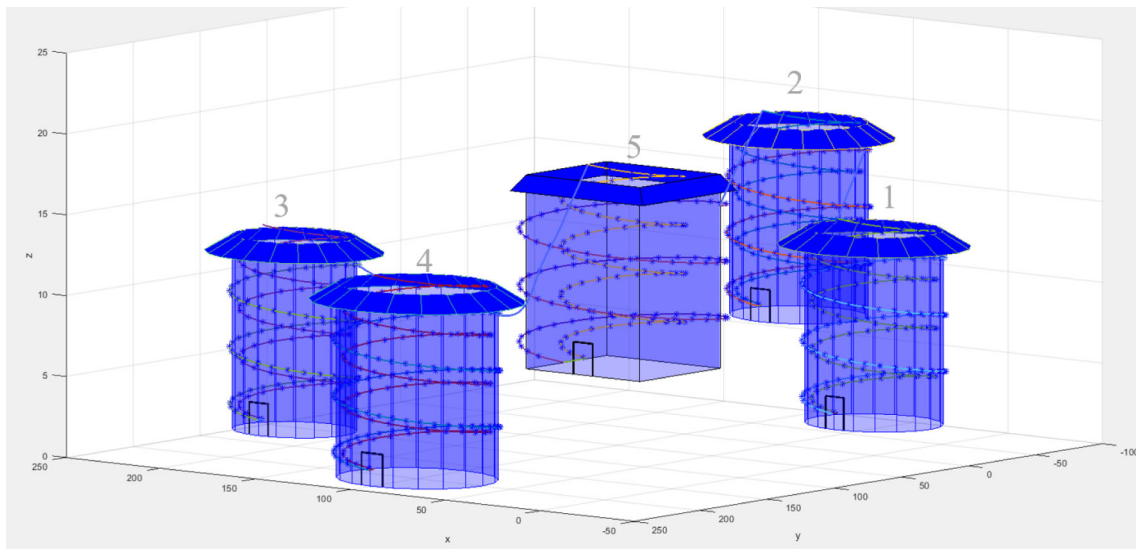
Fig. 6 The exterior inspection paths

Table 4 Costs of the exterior paths

Path	Building 1	Building 2	Building 3	Building 4	Building 5
The Helix-HPSO path	236.8405	236.8405	209.1750	280.4919	299.7959
The back-and-forth path	600.1694	600.1694	542.8375	695.7231	600.1694



(a) The path generated by HPSO



(b) The path generated by PSO

Fig. 7 The generated paths

Table 5 Comparison of PSO and HPSO paths

Path		From 1 to 2	From 2 to 5	From 5 to 4	From 4 to 3
The HPSO path	Iteration times	11	11	32	11
	Best cost	47.4371	39.4697	53.5707	58.0246
The PSO path	Iteration times	11	11	69	11
	Best cost	47.4371	39.4697	53.5756	58.0246

tion keep until the UAV reaches the last building. Table 3 lists the details for the generated path.

Comparison of path planning for each building The back-and-forth path is widely implemented for a building inspection to get the inspection data. It is programmed to compare with the Helix-HPSO approach. The distance between a UAV and a building is less than 2 m. The path length of the vertical flight is 1 m. The back-and-forth path is only for the exterior inspection without considering the costs of taking inspection photos, as shown in Fig. 6a. Figure 6b displays the exterior inspection path by defining the data generation points through the proposed Helix-HPSO approach.

The proposed approach implements the helix-shaped path for coverage inspection. Table 4 compares the costs of the exterior inspection of each building for different paths. The cost only considers the exterior inspection path, and they are calculated by Equation (1), considering the distance, altitude, and time. For getting complete documentation of the building, coverage path planning is required to get complete documentation of the building. If comprehensive coverage is required, the costs of traditional back-and-forth paths rise rapidly.

Comparison of path planning between the buildings PSO is compared with the proposed HPSO for path planning between the buildings. Figure 7 compares the path generated by PSO and HPSO. The HPSO path is marked orange, and the PSO path is marked blue.

Table 5 compares the performance of PSO and HPSO for path planning. If the algorithm gets the best solution ten times, the algorithm terminates. For the path between Buildings 1 and 2, Buildings 2 and 5, and Buildings 4 and 3, PSO and HPSO have the same performance. While for the path between 5 and 4, HPSO has fewer iteration times and costs than PSO. HPSO reduces by around 53.62% iteration times to get the optimal path.

6 Conclusion

The Helix-HPSO path planning approach is developed for UAV multi-building inspection, considering the paths travelling between buildings. Usually, the building inspection algorithm is designed for one building, while our approach is for a multi-building environment. The proposed method can be used for interior inspection, extending the traditional inspection area. The proposed helix path provides broader coverage and smooth turn angles at lower costs than traditional inspection paths. Cost function factors include time, distance, and altitude, as altitude change affects energy consumption and speed.

A visited vector and a cost matrix are implemented to determine the order of proceeding inspections for travelling between buildings. The PSO algorithm is improved by the inspiration of HS for getting rid of trapping on the local optima. The proposed HPSO exhibits outstanding performance when dealing with many local optima in the benchmark functions compared to other bio-inspired algorithms. For UAV path planning, the cost functions consider the distance and achieve collision avoidance. The HPSO provides faster convergence than PSO during UAV path planning, so it is used in the path planning between buildings.

Author Contributions All authors contributed to this research. Shiwei Lin performed model design, Data collection, experiment design and analysis. The draft of the manuscript was written by Shiwei Lin and Ang Liu and commented on by Xiaoying Kong and Jianguo Wang. All authors approved the final manuscript.

Funding No funds or grants were received for this manuscript.

Data availability Statement Not applicable.

Declarations

Conflict of interest No relevant interests to disclose.

Ethics approval This research involves simulation, and there is not required ethics approval.

Consent to participate Not applicable.

Consent for publication All authors agreed to publish the manuscript.

References

- Mader D, Blaskow R, Westfeld P, Weller C (2016) Potential of uav-based laser scanner and multispectral camera data in building inspection. In: ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLI-B1, pp. 1135–1142. <https://doi.org/10.5194/isprsarchives-XLI-B1-1135-2016>
- Phung MD, Quach CH, Dinh TH, Ha Q (2017) Enhanced discrete particle swarm optimization path planning for uav vision-based surface inspection. *Autom Constr* 81:25–33. <https://doi.org/10.1016/j.autcon.2017.04.013>
- Bai X, Jiang H, Cui J, Lu K, Chen P, Zhang M (2021) Uav path planning based on improved a and dwa algorithms. *International journal of aerospace engineering* 2021:1–12. <https://doi.org/10.1155/2021/4511252>
- Wu X, Xu L, Zhen R, Wu X (2019) Biased sampling potentially guided intelligent bidirectional rrt algorithm for uav path planning in 3d environment. *Math Probl Eng* 2019:1–12. <https://doi.org/10.1155/2019/5157403>
- Zhang Z, Li J, Wang J (2018) Sequential convex programming for nonlinear optimal control problems in uav path planning. *Aerosp Sci Technol* 76:280–290. <https://doi.org/10.1016/j.ast.2018.01.040>
- Bolourian N, Hammad A (2020) Lidar-equipped uav path planning considering potential locations of defects for bridge inspection.

- Automation in Construction **117**. <https://doi.org/10.1016/j.autcon.2020.103250>
7. Tian R, Cao M, Ma F, Ji P (2020) Agricultural uav path planning based on improved a and gravity search mixed algorithm. *J Phys: Conf Ser* 1631(1):12082. <https://doi.org/10.1088/1742-6596/1631/1/012082>
 8. Luo G-c, Yu J-q, Mei Y-s, Zhang S-y (2015) Uav path planning in mixed-obstacle environment via artificial potential field method improved by additional control force. *Asian journal of control* 17(5), 1600–1610. <https://doi.org/10.1002/asjc.960>. *istex: D5FC09FA138A45E026C35637F5C022F6E58D4272*
 9. Lin N, Tang J, Li X, Zhao L (2019) A novel improved bat algorithm in uav path planning. *Computers, materials & continua* 61(1), 323–344. <https://doi.org/10.32604/cmc.2019.05674>
 10. Wang J, Wang G, Hu X, Luo H, Xu H (2020) Cooperative transmission tower inspection with a vehicle and a uav in urban areas. *Energies* **13**(2). <https://doi.org/10.3390/en13020326>
 11. Chen Y, Yu J, Mei Y, Wang Y, Su X (2016) Modified central force optimization (mco) algorithm for 3d uav path planning. *Neuro-computing (Amsterdam)* 171:878–888. <https://doi.org/10.1016/j.neucom.2015.07.044>
 12. Yang Q, Yang Z, Zhang T, Hu G (2019) A random chemical reaction optimization algorithm based on dual containers strategy for multi-rotor uav path planning in transmission line inspection. *Concurrency and computation* 31(12). <https://doi.org/10.1002/cpe.4658>
 13. Fu Z, Yu J, Xie G, Chen Y, Mao Y (2018) A heuristic evolutionary algorithm of uav path planning. *Wirel Commun Mob Comput* 2018:1–11. <https://doi.org/10.1155/2018/2851964>
 14. Qu C, Gai W, Zhong M, Zhang J (2020) A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (uavs) path planning. *Appl Soft Comput* 89:106099. <https://doi.org/10.1016/j.asoc.2020.106099>
 15. Huo L, Zhu J, Li Z, Ma M (2021) A hybrid differential symbiotic organisms search algorithm for uav path planning. *Sensors (Basel, Switzerland)* 21(9):3037. <https://doi.org/10.3390/s21093037>
 16. Wang X, Pan J-S, Yang Q, Kong L, Snašel V, Chu S-C (2022) Modified mayfly algorithm for uav path planning. *Drones (Basel)* 6(5):134. <https://doi.org/10.3390/drones6050134>
 17. Jordan S, Moore J, Hovet S, Box J, Perry J, Kirsche K, Lewis D, Tse ZTH (2018) State-of-the-art technologies for uav inspections. *IET Radar, Sonar & Navigation* 12(2):151–164. <https://doi.org/10.1049/iet-rsn.2017.0251>
 18. Roca D, Lagüela S, Díaz-Vilariño L, Armesto J, Arias P (2013) Low-cost aerial unit for outdoor inspection of building façades. *Autom Constr* 36:128–135. <https://doi.org/10.1016/j.autcon.2013.08.020>
 19. Zainorizuan MJ, Kaamin M, Idris NA, Mohd Bukari S, Ali Z, Samion N, Anjang Ahmad M, Yee Yong L, Alvin John Meng Siang L, Mohamad Hanifi O, Siti Nazahiyah R, Mohd Shalahuddin A (2017) Visual inspection of historical buildings using micro uav. In: *MATEC Web of Conferences*, vol. 103. <https://doi.org/10.1051/mateconf/201710307003>
 20. Ham Y, Han KK, Lin JJ, Golparvar-Fard M (2016) Visual monitoring of civil infrastructure systems via camera-equipped unmanned aerial vehicles (uavs): a review of related works. *Visualization in Engineering* 4(1). <https://doi.org/10.1186/s40327-015-0029-z>
 21. Rachele G, Umberto M, Giuseppe M, Francesco P, Manuela R (2020) Collecting built environment information using uavs: Time and applicability in building inspection activities. *Sustainability (Basel, Switzerland)* 12(4731):4731. <https://doi.org/10.3390/su12114731>
 22. Chen K, Reichard G, Akanmu A, Xu X (2021) Geo-registering uav-captured close-range images to gis-based spatial model for building façade inspections. *Automation in Construction* **122**. <https://doi.org/10.1016/j.autcon.2020.103503>
 23. Murtiyoso A, Grussenmeyer P (2017) Documentation of heritage buildings using close-range uav images: dense matching issues, comparison and case studies. *Photogram Rec* 32(159):206–229. <https://doi.org/10.1111/phor.12197>
 24. Seo J, Duque L, Wacker J (2018) Drone-enabled bridge inspection methodology and application. *Autom Constr* 94:112–126. <https://doi.org/10.1016/j.autcon.2018.06.006>
 25. Markova M, Kravchenko D (2018) 3d photogrammetry application for building inspection of cultural heritage objects. *Bulletin of Prydniprovs'ka State Academy of Civil Engineering and Architecture* 1, 91–96. <https://doi.org/10.30838/j.Bspacea.2312.170118.82.44>
 26. Buffi G, Manioli P, Gambi A, Montanari G (2018) Unmanned aerial vehicle (uav) and building information modelling (bim) technologies in concrete dam management: The case of ridracoli. In *bo* 9(13):36–43
 27. Gonzalez de Santos LM, Frias Nores E, Martinez Sanchez J, Gonzalez Jorge H (2021) Indoor path-planning algorithm for uav-based contact inspection. *Sensors (Basel)* **21**(2). <https://doi.org/10.3390/s21020642>
 28. González-deSantos LM, Martínez-Sánchez J, González-Jorge H, Navarro-Medina F, Arias P (2020) Uav payload with collision mitigation for contact inspection. *Automation in Construction* **115**. <https://doi.org/10.1016/j.autcon.2020.103200>
 29. Murtiyoso A, Koehl M, Grussenmeyer P, Freville T (2017) Acquisition and processing protocols for uav images: 3d modeling of historical buildings using photogrammetry. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences IV-2/W2*, 163–170. <https://doi.org/10.5194/isprs-annals-IV-2-W2-163-2017>
 30. Pan N-H, Tsai C-H, Chen K-Y, Sung J (2020) Enhancement of external wall decoration material for the building in safety inspection method. *J Civ Eng Manag* 26(3):216–226
 31. Vacca G, Furfaro G, Dessì A (2018) The use of the uav images for the building 3d model generation. *Remote Sensing and Spatial Information Sciences XLII-4/W8*, 217–223
 32. Biçici S, Zeybek M (2021) An approach for the automated extraction of road surface distress from a uav-derived point cloud. *Automation in Construction* **122**. <https://doi.org/10.1016/j.autcon.2020.103475>
 33. Freimuth H, König M (2018) Planning and executing construction inspections with unmanned aerial vehicles. *Autom Constr* 96:540–553. <https://doi.org/10.1016/j.autcon.2018.10.016>
 34. Liu D, Xia X, Chen J, Li S (2021) Integrating building information model and augmented reality for drone-based building inspection. *Journal of Computing in Civil Engineering* **35**(2). [https://doi.org/10.1061/\(asce\)cp.1943-5487.0000958](https://doi.org/10.1061/(asce)cp.1943-5487.0000958)
 35. Asadi K, Kalkunte Suresh A, Ender A, Gotad S, Maniyar S, Anand S, Noghbaei M, Han K, Lobaton E, Wu T (2020) An integrated ugv-uav system for construction site data collection. *Automation in Construction* **112**. <https://doi.org/10.1016/j.autcon.2019.103068>
 36. Nex F, Duarte D, Steenbeek A, Kerle N (2019) Towards real-time building damage mapping with low-cost uav solutions. *Remote Sensing* **11**(3)
 37. Kang D, Cha Y-J (2018) Autonomous uavs for structural health monitoring using deep learning and an ultrasonic beacon system with geo-tagging. *Computer-Aided Civil and Infrastructure Engineering* 33(10):885–902. <https://doi.org/10.1111/mice.12375>
 38. Kucuksubasi F, Sorguc A (2018) Transfer learning-based crack detection by autonomous uavs. In: *35th International Symposium on Automation and Robotics in Construction (ISARC 2018)*
 39. Lin S, Kong X, Wang J, Liu A, Fang G, Han Y (2021) Development of a uav path planning approach for multi-building inspection with minimal cost. In: *Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, pp. 82–93

40. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory, pp. 39–43. <https://doi.org/10.1109/MHS.1995.494215>
41. Zong Woo G, Joong Hoon K, Loganathan GV (2001) A new heuristic optimization algorithm: Harmony search. *Simulation* (San Diego, Calif.) 76(2), 60–68. <https://doi.org/10.1177/003754970107600201>
42. Goldberg DE, Holland JH (1988) Genetic algorithms and machine learning. *Mach Learn* 3(2):95–99. <https://doi.org/10.1023/A:1022602019183>
43. Yang X-S Firefly Algorithm, Lévy Flights and Global Optimization, pp. 209–218. Springer, London (2009). https://doi.org/10.1007/978-1-84882-983-1_15

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.