Development of a UAV path planning approach for multi-building inspection with minimal cost

Shiwei Lin[1], Xiaoying Kong[1], Jack Wang[1], Ang Liu[1], Gengfa Fang[1] and Yunlong Han[1]

[1] Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, Australia

Abstract. This paper presents a UAV path planning approach for multi-building inspection, which is a new application for UAV path planning. It generates helix paths for single building inspection first and defines the possible points for collecting inspection data with reasonable time slots. After inspecting one building, the UAV flies to another building with a trajectory based on a cost matrix and a visited vector defined in this algorithm. The planning of the entire inspection path is evaluated considering several factors, such as distance, time, and altitude. The proposed algorithm is applied to historical giant communal homes, Fujian Tulou, consisting of five buildings.

Keywords: UAV, 3D path planning, multi-building.

1 Introduction

Historical buildings are symbols of specific eras. They have geographical and historical value and can vividly express past people live and the aesthetic and philosophy of the architect. Regular inspection of the historical buildings is essential to monitor their condition. Unmanned Aerial Vehicle (UAV) provides the ability of remote inspection by reaching the area that is hard for people to access. It provides the flexibility of building inspection. UAVs can gain different digital imagery with different cameras or sensors, such as ultrasonic sensors, high-resolution cameras, laser scanners, thermal cameras, near-infrared cameras [1, 2].

The collected data can be processed using computer vision and other technologies to detect surface decencies of infrastructure, including spalling, cracking, distortion, rusting, excessive movements, and misalignment [1]. Path planning for UAV-based inspection requires to find a path which is efficient and informative to obtain data from different views [1]. It also requires obstacle avoidance with an effective and optimal path if possible [1, 2]. In this research, path planning for UAV-based inspection of a group of historical buildings, Fujian Tulou. A Tulou is usually a large, enclosed, and fortified earth building, most commonly rectangular or circular in configuration, with very thick load-bearing rammed earth walls between three and five stories high and housing up to 800 people. Figure 1 shows the Tianluokeng Tulou cluster, consisting of five buildings with a square building in the center.

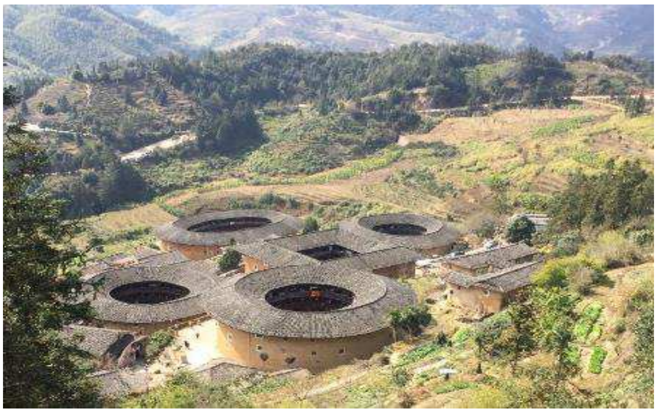

Fig. 1. Photo of the Tianluokeng Tulou cluster

2 Related Work

Regular inspection of buildings is required to ensure their condition is safe. It is more important to the historical buildings as heritage [3]. In general practice, the inspection is conducted by taking photos directly for every element to record the dilapidation or damage, which is expensive and time-consuming [2, 3]. With the development of UAV and relevant photogrammetry technologies, it provides great flexibility in many fields, such as urban planning, security, rural environment monitor [4], and recently building inspection [1, 3, 5-8].

The research of building inspection is to check the environmental condition and the appearance of a building and evaluate the condition of building [3]. The UAV survey is aiming to model the structure and recognize the condition [7]. UAV-based photogrammetry allows full documentation of buildings and evaluates the condition of buildings with less human resources in a short time, particularly for the places that are hard to reach [2-5].

UAVs follow a programmed flight path, capture digital images, and obtain survey data for unreachable areas from a terrestrial platform [2, 4]. UAVs are equipped with sensors for 3D data acquisition, obstacle avoidance, and navigation [1, 2, 4]. It achieves low cost and higher flexibility [8]. The survey data can be different from some special sensors, such as images, Forward-Looking Infrared technology, laser scanner, 3D point cloud, multi-attributed point cloud, and separate strips [2, 4, 8, 9].

Close range photogrammetric images or laser scanning data, or the combination of both are generally used for documentation of building inspection [2, 5]. With proper path planning, UAV can create a precise model of buildings with 2D photographs with dedicated software for identifying the defects [5]. The geometry of buildings is measured in the inspection, and required documentation is usually in the form of photogrammetric images [4, 9]. Deep learning algorithms and advanced photogrammetric techniques are employed to record building damages by a true-orthophoto autonomously [10]. A Convolutional Neural Network is fine-tuned for detecting surface crack [11].

There are many algorithms and methods that can be used for UAV path planning, such as rapidly exploring tree, genetic algorithm, particle swarm optimization, and A* search. They can also be used for building inspections. Particle swarm optimization is improved as enhance discrete particle swarm optimization for path planning of UAV inspection for bridges or buildings which have planar surfaces [1]. Besides, the inspection of the Perak Museum plans the path as flying from the bottom to the top of the building and moving to the right side, then moving to the ground [3]. The most common method for path planning in building inspection is the back and forth path.

However, a specific method should be applied for optimizing UAV path planning in different cases. This paper presents a UAV path planning approach for multi-building inspection, which is applied to a group of historical buildings, Fujian Tulou. They are unique enclosed and fortified earth buildings, and the general back and forth path is not an efficient path for inspection of the entire building. In order to inspect the condition of the entire buildings, including both interior and exterior side of the buildings and their roofs, to check if they are under good condition, optimal path planning is required. The building inspection methods proposed in the literature are only for a single building. But in our case, a group of buildings is inspected together, which requires path planning for different buildings and between the buildings.

3 Path Planning Algorithm

The proposed path planning algorithm generates helix paths for a single building inspection and defines the points for collecting inspection data with reasonable time slots. After inspecting one building, the UAV flies to another building with a trajectory based on a cost matrix and a visited vector. The planning of the entire inspection path is evaluated considering several factors, such as distance, time, and altitude.

3.1 Description of the algorithm

Using $x(t)$,$y(t)$,$z(t)$ to indicate the position of the UAV at the specific time t. The helix path for single building inspection can be defined as:

$$x(t) = r * \sin(vt/2\pi r) + x_g \tag{1}$$

$$y(t) = r * \cos(vt/2\pi r) + y_g \tag{2}$$

$$z(t) = \frac{vh}{2\pi r} * t + z_g + 1 \tag{3}$$

where x_g, y_g, z_g is the center coordinate of the ground floor of a building; v is the UAV flight speed, h is the height of each floor of the building, and r is the radius of the helix path. For interior side inspection, r is the radius of the building minus a few meters, for exterior inspection, r equals the radius of the building plus 2 meters.

With an optimal UAV flight speed v, decided by the onboard sensors and inspection condition, the time required for inspection one building with n floors is:

$$Ts = \frac{2\pi rn}{v} \tag{4}$$

After the inspection of one building's interior and exterior side, the UAV flies to another building for inspection. The principal consideration for path planning between buildings is the costs of distance, then attitude and time. The point of tangency of the helix path is selected to reduce the costs by a smooth path and avoid collisions with buildings, which is represented as (x_0, y_0, z_0), and z_0 is considered as the height as the destination building. The last point of the previous path is (x_1, y_1, z_1), The last point of the previous path on the same height as the point of tangency is (x_1, y_1, z_0). The Pythagorean Theorem is used to calculate the point of tangency, where the distance between the center point and the point of tangency is the radius of the destination building plus 5 meters to avoid collisions.

$$\mathrm{c} = \sqrt{(x_1 - x_c)^2 + (y_1 - y_c)^2}$$

$$\mathrm{a} = r + 5$$

$$b = \sqrt{c^2 - a^2} \tag{5}$$

where (x_c, y_c) is the coordinate of the center of the destination building, and r is the radius of the destination building.

The equations of x_0, y_0 are calculated with (5) as:

$$a^2 = (x_0 - x_c)^2 + (y_0 - y_c)^2 \tag{6}$$

$$b^2 = (x_0 - x_1)^2 + (y_0 - y_1)^2 \tag{7}$$

where (x_c, y_c) is the coordinate of the center of the destination building.

Then make the solution to equations that is close to the destination as (x_0, y_0, z_0). The path is generated by three points, the start coordinate (x_1, y_1, z_1), the destination coordinate located in the next building, and (x_0, y_0, z_0) which calculated by (5-7). The points of the path between the buildings is represented as $(x(t)$,$y(t)$,$z(t))$ at the specific time t. The equations of $x(t)$, $y(t)$ are:

$$x(t) = a_1 t^2 + b_1 t + c_1 \tag{8}$$

$$y(t) = a_2 t^2 + b_2 t + c_2 \tag{9}$$

where a_1, b_1, c_1 are the coefficients of $x(t)$, and a_2, b_2, c_2 are the coefficients of $y(t)$.

Point 1 is the start coordinate (x_1, y_1, z_1). Point 2 is (x_0, y_0, z_0), and Point 3 is the destination coordinate. The maximum time slots for two path segments are t_{max1} and t_{max2} are calculated by the Euclidean distance:

$$Distance1 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

$$t_{max1} = \frac{Distance1}{v} \tag{10}$$

$$Distance2 = \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2 + (z_3 - z_2)^2}$$

$$t_{max2} = \frac{Distance2}{v} \tag{11}$$

where Point 1 is (x_1, y_1, z_1), Point 2 is (x_2, y_2, z_2), Point 3 is (x_3, y_3, z_3), and v is the constant speed.

The coefficients of $x(t)$, $y(t)$ are calculated by the following system (12), (13):

$$\begin{bmatrix} 0 & 0 & 1 \\ {t_{max1}}^2 & t_{max1} & 1 \\ (t_{max1} + t_{max2})^2 & t_{max1} + t_{max2} & 1 \end{bmatrix} * \begin{bmatrix} a_1 \\ b_1 \\ c_1 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{12}$$

$$\begin{bmatrix} 0 & 0 & 1 \\ {t_{max1}}^2 & t_{max1} & 1 \\ (t_{max1} + t_{max2})^2 & t_{max1} + t_{max2} & 1 \end{bmatrix} * \begin{bmatrix} a_2 \\ b_2 \\ c_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \tag{13}$$

where a_1, b_1, c_1 are the coefficients of $x(t)$, and a_2, b_2, c_2 are the coefficients of $y(t)$; t_{max1} is the maximum time slot for traveling from Point 1 to Point 2, t_{max2} is the maximum time slot for traveling from Point 2 to Point 3.

The equations of $z(t)$ is:

$$z(t) = a_3[(x(t) - x_1)^2 + (y(t) - y_1)^2] + b_3\sqrt{(x(t) - x_1)^2 + (y(t) - y_1)^2} + c_3 + z_1 \tag{14}$$

where a_3, b_3, c_3 are the coefficients of $z(t)$, and (x_1, y_1, z_1) is the start coordinate of the path segment.

The coefficients of $z(t)$ are calculated by the following system (15):

$$\begin{bmatrix} 0 & 0 & 1 \\ (x_2 - x_1)^2 + (y_2 - y_1)^2 & \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} & 1 \\ (x_3 - x_1)^2 + (y_3 - y_1)^2 & \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2} & 1 \end{bmatrix} * \begin{bmatrix} a_3 \\ b_3 \\ c_3 \end{bmatrix} = \begin{bmatrix} z_1 - z_1 \\ z_2 - z_1 \\ z_3 - z_1 \end{bmatrix} \tag{15}$$

where a_3, b_3, c_3 are the coefficients of $z(t)$, Point 1 is (x_1, y_1, z_1), Point 2 is (x_2, y_2, z_2), and Point 3 is (x_3, y_3, z_3).

The path of *n* points is stored as:

$$\text{Points} = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{bmatrix} \tag{16}$$

3.2 Cost functions

The cost function is to evaluate the paths, and the path with minimal cost is considered as the best path. The weights of cost functions of the inspection and flying between the buildings are different.

The cost function is defined as:

$$f_{Cost} = w_1 * f_{distance} + w_2 * f_{time} + w_3 * f_{altitude} \tag{17}$$

where the sum of w_1, w_2 and w_3 is 1, $f_{distance}$ is the cost function of the distance, $f_{time}$is the cost function of flight time, $f_{altitude}$ is the change of the altitude. For inspection, w_1 is smaller than w_2 and w_3, because the distance is not the primary consideration. But for traveling between buildings, w_1 is larger than w_2 and w_3.

Distance

The cost function of distance is:

$$f_{distance} = \sum_{t=0}^{n} \sqrt{(x_{tNext} - x_t)^2 + (y_{tNext} - y_t)^2 + (z_{tNext} - z_t)^2} \tag{18}$$

where n is the size of t in the path, and $\sqrt{(x_{tNext} - x_t)^2 + (y_{tNext} - y_t)^2 + (z_{tNext} - z_t)^2}$ is the distance between the positions of each two t.

Altitude

The cost function of altitude is:

$$f_{altitude} = \sum_{t=0}^{n} \sqrt{(z_{tNext} - z_t)^2} \tag{19}$$

where n is the size of t in the path, and $\sqrt{(z_{tNext} - z_t)^2}$ is the change of altitude between the positions of each two t.

Time

The cost function of time is:

$$f_{time} = \sum_{t=0}^{n} \frac{\sqrt{(x_{tNext} - x_t)^2 + (y_{tNext} - y_t)^2 + (z_{tNext} - z_t)^2}}{v_t} \tag{20}$$

where n is the size of t in the path, and $\sqrt{(x_{tNext} - x_t)^2 + (y_{tNext} - y_t)^2 + (z_{tNext} - z_t)^2}$ is the distance between the positions of each two t, and v_t is the current v. v_t is determined by the change of altitude. v_t is different when the UAV is descent, ascent, or flying at the same altitude.

3.3 Implementation of the Algorithm

We implement the algorithm as follows: building the model for the Fujian Tulou, generate the path, then generate the points of collecting inspection data for exterior and interior side inspection. After inspecting one building, the UAV is planned to fly to another building.

The path of inspection is generated by (1), (2), and (3). The line space of t is set to 0.01. (x_g, y_g, z_g), the number of floors, the radius of each building, the height of each floor, the height of the roof are inputted. Once $z(t)$ reach to the specific height based on the inspection type, the iteration terminates and record current t as t_{max}. $z(t)$ is designed to almost reach the roof for exterior inspection and over the height of the building for interior side inspection to cover more areas. Using an empty matrix to store the points of the path as (16).

The points of generating inspection data are calculated from the generated paths, and they are related to the total number of rooms. The total room number is calculated based on the number of floors in the building and the number of rooms on each floor, and it is inputted with the path for generated points of generating inspection data as (4). The points of generating inspection data are concentrated in the middle of the path to generating valid inspection data for inspection. Then connecting the paths of the interior and exterior side inspection through the door and inspecting the entrance hall. The UAV is assumed to be charged after the interior side inspection or the exterior side inspection of one building.

There are five buildings: 1 2 3 4 5, and the order of buildings is determined by the cost matrix (17). Update the visited vector if the building has been visited and update the related costs of the visited building as 999. After defining the order, using (5) – (7) to get (x_0, y_0, z_0), then generate the path between every two buildings by (8) – (15).

$$Cost = \begin{bmatrix} 999 & cost(1,2) & cost(1,3) & cost(1,4) & cost(1,5) \\ cost(2,1) & 999 & cost(2,3) & cost(2,4) & cost(2,5) \\ cost(3,1) & cost(3,2) & 999 & cost(3,4) & cost(3,5) \\ cost(4,1) & cost(4,2) & cost(4,3) & 999 & cost(4,5) \\ cost(5,1) & cost(5,2) & cost(5,3) & cost(5,4) & 999 \end{bmatrix} \tag{21}$$

When processing to another building, the center and radius of the destination building are inputted to calculate *a, b, c* for the Pythagorean Theorem. Basing on the start coordinate and the destination coordinate and combining with Pythagorean Theorem, x_0, y_0 can be calculated as (5) – (7). The height of this point is supposed to be the same height as the destination point. Also, there will be two solutions for x_0, y_0, using the Euclidean distance to measure which one is closer to the destination point, and get the one that is closer to the destination point as the only solution.

Setting Point 1 as the start point. Point 2 and Point 3 are determined by the distance. If the (x_0, y_0, z_0) is closer to the start point than the distance between the start point and the destination point, then Point 2 is (x_0, y_0, z_0). Otherwise, Point 2 is the destination point. The coordinates of Point 1, Point 2, Point 3, and the destination are inputted for generating a path between two buildings. The constant speed is set to calculate t_{max1} and t_{max2}. The coefficients $\begin{bmatrix}a_1\\b_1\\c_1\end{bmatrix}$, $\begin{bmatrix}a_2\\b_2\\c_2\end{bmatrix}$ and $\begin{bmatrix}a_3\\b_3\\c_3\end{bmatrix}$ are calculated for getting the equations of $x(t)$,$y(t)$,$z(t)$ as (8) –(15). The line space of t is set to 0.01 for $x(t)$,$y(t)$,$z(t)$. If the path almost reaches the destination, the iteration terminates.

4 Simulation results

The model of the Fujian Tulou is presented in Figure 2, and the number is marked with each building:

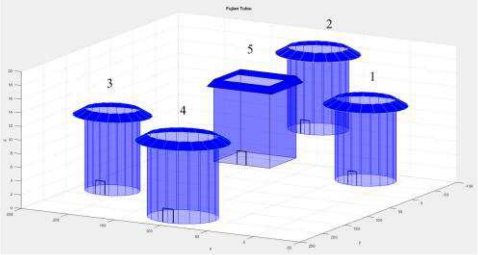

Fig. 2. The Fujian Tulou

To validate the algorithm, we use the simulation for path planning with MATLAB. The center coordinate and the radius of each building are inputted for generating the path of inspection, but the radiuses of building 5 are different between interior and exterior side inspection due to the particularity of the building. The speed of ascent is set to 6 m/s, the speed of descent is 4 m/s, and the speed is 18 m/s. When the UAV in the points of generating inspection data, UAV hovers. Because Building 5 is rectangular in configuration, we have considered making a similar path as the shape of the building, while the cost is higher than the helix path.

Figure 3 shows the flight path for each building, and the points of generating inspection data are marked by “*”.

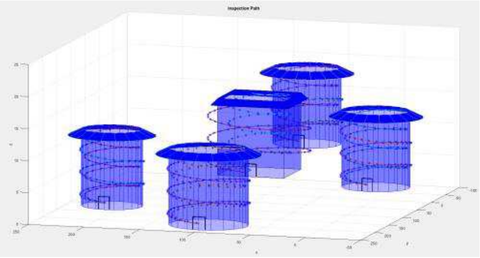

Fig. 3. The flight path for each building

The start building is Building 1. From the cost matrix (17) and the visited vector, the order of traveling the buildings is calculated as 1 2 5 4 3 by Dijkstra's algorithm.

The start point for Building 1 is the start of the exterior inspection path, which is close to the top of Building 1.

Figure 4 integrates the flight path as Path 1 for the group of buildings, and (x_0, y_0, z_0) is marked by "o."

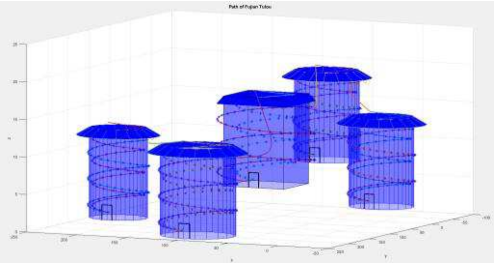

Fig. 4. The 3D flight path (Path 1) from the side viewpoint

Figure 5 presents the 3D flight path for the group of buildings from the top viewpoint.

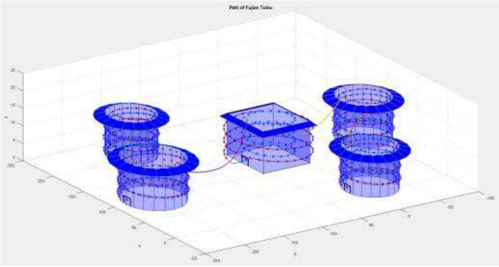

Fig. 5 The 3D flight path (Path 1) from the top viewpoint

Figure 6 shows another 3D flight path as Path 2 for the group of buildings. The path segments between the buildings are different. The destination points of Building 5 and 4 are different, so the direction of the inspection paths has been revised.

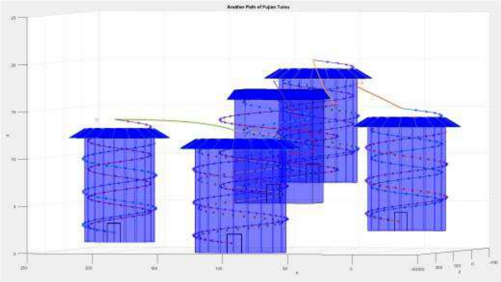

Fig. 6. Another 3D flight path (Path 2) from the side viewpoint

Figure 7 displays another 3D flight path for the group of buildings from the top viewpoint.

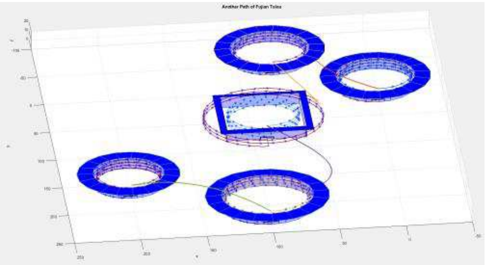

Fig. 7. Another 3D flight path (Path 2) from the top viewpoint

The traditional back and forth paths are usually applied to the exterior area of each building to get the inspection data. The back and forth paths are programmed to compare with the proposed methodology. The distance between the inspected building and UAV is equal to or less than 2m [3]. The path length of the horizontal flight is 2m. Figure 8 demonstrates the back and forth paths for the exterior area without defining the points of generating the inspection data.

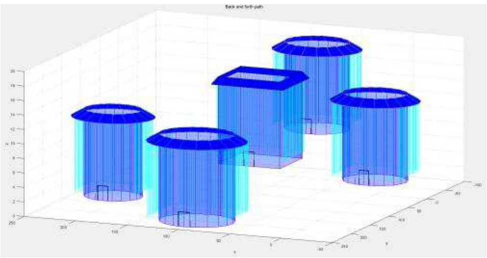

Fig. 8. The back and forth paths (Path 3) for each building

Setting the back and forth path for the vertical flight for Path 4 in Figure 9.

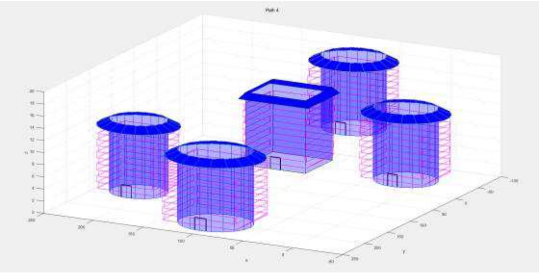

Fig. 9. Another back and forth paths (Path 4) for each building

The costs of the entire path are related to the path of each building, and the paths between every two buildings and the direction of the flight path are important. Table 1 shows the costs for the paths generated by the proposed methodology. Path 1 and Path 2 are generated by the proposed approach. Total cost calculates the costs of inspecting buildings and traveling between buildings. Path 3 and Path 4 are the back and forth paths without defining the inspection points. Table 2 shows the costs of the path of each building.

Table 1. Costs

Path	Each building					The path between				Total cost
	Building 1	Building 2	Building 3	Building 4	Building 5	Building 1 and 2	Building 2 and 5	Building 5 and 4	Building 4 and 3	
Path 1	532.5874	532.5874	469.2459	633.2072	606.2504	55.3632	46.3745	51.9892	43.9798	2971.5849
Path 2	532.5874	532.5874	473.5241	633.2072	606.2504	55.3632	46.3745	72.9840	68.6634	3021.5415

Table 2. Costs of each building

Path	Building 1	Building 2	Building 3	Building 4	Building 5
Path 1	532.5874	532.5874	469.2459	633.2072	606.2504
Path 2	532.5874	532.5874	473.5241	633.2072	606.2504
Path 3	859.4475	859.4475	746.5488	975.0643	859.4475
Path 4	600.1694	600.1694	542.8375	695.7231	600.1694

From comparing the costs among paths, Path 1 has the lowest costs, and the total cost of Path 1 is 2971.58. For a single building inspection, the proposed methodology has much lower costs than the back and forth path. The points of generating the inspection data are not defined for the back and forth paths, and the back and forth paths are only for the exterior area. The costs of the proposed methodology for the single building include the exterior and interior side inspection path, and the consumed time for hovering and generating the inspection data. The costs of Path 3 and Path 4 only include the exterior path, but they are usually higher than the proposed methodology. If the back and forth path wants more coverage, the costs rise rapidly. The total costs of the back and forth paths must be much higher than the proposed methodology if considering the consumed time of generating the data, the interior path, and the path of traveling between buildings. The proposed methodology ensures higher performance with a smoother path, fewer costs, and more coverage.

5 Conclusion

The path planning approach is developed for multi-building inspection with UAV, with the paths for traveling between buildings. The algorithm uses the helix path for collecting single building inspection data because the helix paths have a smooth turn angle and broader coverage. For traveling between buildings, both cost matrix and a visited vector are employed to calculate the order of processing inspections. The cost function assigns different weights to the cost for a single building and for traveling between buildings, basing on the intention of inspection. The factors considered in the cost function are distance, time, and altitude, as altitude change affects speed and energy consumption. The proposed path planning has been compared with the traditional back and forth path, and it has much better performance with less cost.

References

1. Phung, M.D., Quach, C.H., Dinh, T.H., and Ha, Q.: 'Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection'. Automation in Construction (81), 25-33 (2017).
2. Mader, D., Blaskow, R., Westfeld, P., and Weller, C.: '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, pp. 1135-1142. Copernicus GmbH, Czech Republic (2016).
3. Zainorizuan, M.J., Kaamin, M., Idris, N.A., Mohd Bukari, S., Ali, Z., Samion, N., Anjang Ahmad, M., Yee Yong, L., Alvin John Meng Siang, L., Mohamad Hanifi, O., Siti Nazahiyah, R., and Mohd Shalahuddin, A.: 'Visual Inspection of Historical Buildings Using Micro UAV'. In: MATEC Web of Conferences, vol. 103, EDP Sciences, (2017).
4. Roca, D., Lagüela, S., Díaz-Vilariño, L., Armesto, J., and Arias, P.: 'Low-cost aerial unit for outdoor inspection of building façades'. Automation in Construction (36), 128-135 (2013).
5. Markova, M., and Kravchenko, D.: '3D photogrammetry application for building inspection of cultural heritage objects'. Bulletin of Prydniprovs'ka State Academy of Civil Engineering and Architecture (1), 91-96 (2018).
6. Grosso, R., Mecca, U., Moglia, G., Prizzon, F., and Rebaudengo, M.: 'Collecting Built Environment Information Using UAVs: Time and Applicability in Building Inspection Activities'. Sustainability (12), 4731 (2020).
7. Buffi, G., Manciola, P., Gambi, A., and Montanari, G.: 'Unmanned Aerial Vehicle (UAV) and Building Information Modelling (BIM) Technologies in Concrete Dam Management: The Case of Ridracoli'. In bo (9), 36-43 (2018).
8. Vacca, G., Furfaro, G., and Dessì, A.: 'THE USE OF THE UAV IMAGES FOR THE BUILDING 3D MODEL GENERATION'. Remote Sensing and Spatial Information Sciences (XLII-4/W8), 217-223 (2018).
9. Pan, N.-H., Tsai, C.-H., Chen, K.-Y., and Sung, J.: 'Enhancement of external wall decoration material for the building in safety inspection method'. Journal of Civil Engineering and Management (26), 216-226 (2020).
10. Nex, F., Duarte, D., Steenbeek, A., and Kerle, N.: 'Towards Real-Time Building Damage Mapping with Low-Cost UAV Solutions'. Remote Sensing (11), (2019).
11. F. Kucuksubasi and A. Sorguc, "Transfer Learning-Based Crack Detection by Autonomous UAVs". In: 35th International Symposium on Automation and Robotics in Construction (ISARC 2018), arXiv.org, (2018).