

RESEARCH ARTICLE

Evaluation and Enhancement of Resolution-Aware Coverage Path Planning Method for Surface Inspection Using Unmanned Aerial Vehicles

WEITONG WU¹, YUKI FUNABORA¹, (Member, IEEE),
SHINJI DOKI¹, (Senior Member, IEEE), KAE DOKI², (Member, IEEE),
SATORU YOSHIKAWA³, TETSUJI MITSUDA⁴, AND JINGYU XIANG³

¹Department of Electrical Engineering and Computer Science, Graduate School of Engineering, Nagoya University, Nagoya, Aichi 464-8603, Japan

²Department of Electrical and Electronics Engineering, Aichi Institute of Technology, Toyota, Aichi 470-0392, Japan

³SOOKEN INC., Nisshin, Aichi 470-0111, Japan

⁴DENSO Corporation, Kariya, Aichi 448-8661, Japan

Corresponding authors: Weitong Wu (doki_wu@nagoya-u.jp) and Yuki Funabora (funabora@nagoya-u.jp)

This work was supported in part by the Regional Reconstruction Practical Development Promotion Project Cost Subsidy.

ABSTRACT We implemented and evaluated our previous path planning method for inspection using unmanned aerial vehicles (UAVs) in real-world, and identified its shortcomings in handling positioning errors. Then, we proposed an enhanced method to address this problem. The previous method theoretically guaranteed complete coverage of targets and data quality. However, we verified it in bridge inspection experiments and found that the former has not been ensured. The crucial factors of data omission are clarified as the errors in UAV positioning. Our previous method relies on appropriately setting ideal allowances to counteract positioning errors, which is challenging in practice. Therefore, we proposed an enhanced path planning method, which adaptively adjusts allowances according to positioning error to prevent omission while minimizing waypoints. In the simulation including positioning disturbances, the enhanced method consistently achieved full coverage in 1000 times simulation with over 28% waypoints less than the previous one.

INDEX TERMS Surface inspection, visual inspection, inspection path planning, coverage path planning (CPP), unmanned aerial vehicle (UAV), remote sensing, structural health monitoring.

I. INTRODUCTION

Recently, unmanned aerial vehicles (UAVs) equipped with cameras have gained popularity for capturing high-resolution images of construction surfaces to facilitate visual inspections. This approach can reduce the operational cost [1], [2], [3], [4] and enable more frequent inspections to ensure the safety of construction. Practical case studies have validated the feasibility of UAV-based inspection [5], [6], [7], [8], [9]. and specialized UAVs and equipment are constantly developed [10], [11], [12], [13], [14], [15] to better collect

data. Once surface data is collected, the defects can be detected visually or using data processing technologies [16], [17], [18], [19], [20], [21], [22]. In such a context, we aim to use UAVs to effectively and stably collect qualified inspection datasets in practice.

Data quality, especially the data spatial resolution (also known as the ground sampling distance, GSD) is critical to the image-based defect identification [23], [24], [25]. Take crack detection of concrete bridges as an example, the maximum allowable crack width varies according to the standards of countries and regions [26], 0.2 mm for Germany [27] and 0.1 mm for Japan [28]. Meanwhile, according to the literature [29], [30], [31], the minimum

The associate editor coordinating the review of this manuscript and approving it for publication was Halil Ersin Soken¹.

width of detectable cracks on images has been proven to be proportional to the image GSD. Thus, in order to detect cracks above a particular width on a given construction, it is necessary to collect its surface dataset at enough GSD.

Since the relationship between GSD and the sensor's relative pose to objects can be modeled [32], theoretically, we can collect inspection data at the desired GSD by appropriately planning UAV paths according to construction shapes. However, although the importance of data GSD has been recognized, the path planning method that ensures GSD has not been well studied. In previous research [33], [34], we have proposed a coverage path planning (CPP) method that uses targets' 3D models to plan their inspection paths. We validated it in simulations and confirmed that it can inspect targets under ideal conditions. However, practical inspection environments include various disturbances that affect data quality, yet they have not been introduced in our ideal simulations. It's only through validation in the real world that we can ascertain whether our method can still reliably collect desired data. If not, the experiment results can help us to identify crucial factors that require further attention and to determine whether adjustments are necessary to enhance the method's performance.

To further validate our previous algorithm [33], [34] and clarify if it needs enhancement, we implemented it to inspect some surfaces of a bridge in this study. Results indicate that the targets are almost successfully inspected and the GSD of collected data is qualified, yet the positioning errors of UAV resulted in a small percentage of omissions. These positioning errors are inevitable in practice, and handling them in the previous method's framework is challenging. Users must set an ideal parameter to control data redundancy and prevent omission, which is impractical in real inspection. Consequently, we further enhanced our path planning method to enable it to adaptively determine parameters according to limited positioning and orientation errors. The contributions of this study are summarized as follows:

- The practical performance of our previous path planning method for inspection is evaluated. The data quality can be ensured yet errors in coordinate conversion and positioning led to a data omission of 7.49% in the worst case.
- The disturbance factor that should be further handled in practical UAV inspection is clarified as the errors in UAV positioning.
- An enhanced method is proposed, which adaptively adjusts the allowances of paths according to the errors in UAV positioning to prevent data omission.
- Simulations including the disturbances of UAV positioning are conducted. Results show that the enhanced method is robust to position, data quality is ensured and no omission occurs in 1000 times inspection.

The rest of this paper is organized as follows. The section II reviews the works related to inspection path planning. The section III defines the inspection data collection problem we

focus on and briefly introduces our CPP method [33], [34]. The section IV introduces the practical experiment and how we planned the flight paths for the UAV. The experiment results are summarized in section V. We provide quantitative evaluations of the collected data in terms of coverage rate and GSD. In section VI we introduce our enhanced CPP method.

II. RELATED WORKS

The path planning of UAV is a frequent and persistent problem in many applications, including 3D reconstruction [35], [36], SAR [37], [38], [39], surveying [40], [41], communication [42], and construction inspection. There are many commercial software available for path planning, such as DJI GS PRO [43], ArduPilot [44], Pix4D [45], and QGround [46]. They allow users to specify the vertexes of 2D regions of interest on their user interface and they will plan lawnmower scan paths. However, using them to plan paths for construction inspection can be a labor-intensive task. Because constructions usually have complex shapes and many separate surfaces, planning the path for each surface costs numerous man-hours. Moreover, although these software provide features like structure scanning, they can only approximate the targets as cylinders and plan simple circular paths to scan them. Such paths can not guarantee the desired data GSD since the actual inspection distance and angle are uncontrollable. In conclusion, these software are considered unsuitable for the construction inspection.

The above problems can be avoided by planning paths based on the 3D model of target construction. The 3D model can be acquired by laser scanners, digital twins, or BIM. Recently, various model-based inspection path planning methods have been proposed. Almadhoun et al. [47] proposed a method that creates adaptive sensor poses based on the construction's 3D model to navigate the UAV toward areas with poor data quality and coverage. Jing et al. [48] proposed a method to generate discrete sensor poses and planned paths for UAV swarms to reach them. These methods generate sensor pose candidates by random sampling and filtering out those poses which ensure data GSD, and plan a path to visit them. However, the sampling-based approach is hard to guarantee 100% coverage of the inspection target, which will cause defect misses in inspections. Another prevalent approach for CPP is cell decomposition. For example, in the study of Phung et al. [49], they establish a grid for each target surface and plan sensor poses to inspect every occupied cell at the desired GSD, then plan paths to visit them. This approach ensured complete coverage, yet the quantity of sensor poses may not be minimized, which will increase the UAV battery consumption and the man-hours for data post-processing. Besides, Peng and Isler [50] proposed a CPP method for covering large-scale 3D urban constructions. Ivić et al. [51] proposed a Heat Equation Driven Area Coverage (HEDAC) algorithm to address the 3D inspection problem of complex structures. These methods ensured that every point of the target could be inspected via at least one point on the

paths. However, they did not indicate the exact poses that UAVs should collect data, which is indispensable for practical inspection.

Contrarily, our CPP method [33], [34] divides the 3D model into as few clusters as possible and plans poses for each cluster. As a result, the UAV only needs to collect images at sensor poses and complete coverage can be achieved, the sensor poses can also be minimized. We believe that this method is the one most probable to realize the UAV inspection we desire. Therefore, we validated it in real inspection experiments to clarify if it delivers on what it theoretically promises.

III. AUTOMATIC SURFACE DATA COLLECTION USING UAV

A. PROBLEM FORMULATION

Our goal is to use the UAV to automatically collect the construction's surface data, a CCD camera attached to a controllable gimbal is located at the UAV center. The collected data must be qualified to enable technicians to assess the condition of the construction. The data requirements we considered are listed below:

- **Resolution requirement:** Every point on the inspection target should be inspected at the required GSD G^* (mm/pixel) to ensure that defects are discernible. G^* is determined according to the user's needs, smaller value brings better data quality and more details of inspected surfaces.
- **Coverage requirement:** The photoset must cover the inspection target not less than the target coverage rate, which is usually 100% to ensure no defect will be missed in inspections.

We determine a set of inspection positions $P_{insp} = \{p_1, p_2, \dots, p_N\}$ where UAV collects the desired photos, and plan a collision-free path ξ as short as possible to navigate the UAV to visit P_{insp} and collect the desired inspection data. One inspection position $p \in P_{insp}$ is a vector with six dimensions. The first three elements of p represent the camera coordinates and the last three elements represent the direction in which the camera takes a photo.

We supposed that the following information is available for planning P_{insp} and ξ . The inspection target is presented as a set of triangular polygons M_{tar} (STL [52] format) in virtual space \mathbb{V}^3 . The M_{tar} can be a set of planar or uneven surfaces, bridge piers, or dome structures of arch bridges and tunnels (see [33], [34] for more path samples). The size L and resolution I of the image sensor of UAV's camera in the direction of its short edge, and the focal length f are known. A set of ground control points (GCP) on the target are previously measured, and their coordinates in \mathbb{V}^3 and \mathbb{R}^3 are known. We only need to plan ξ in \mathbb{V}^3 since ξ can be easily converted into the real world \mathbb{R}^3 using GCP.

Finally, we consider the following constraints. There exists positioning error E_{pos} of UAVs. The view angle of any point on the target surface should not exceed the threshold β_{max} for ensuring GSD. The view angle is defined as the angle

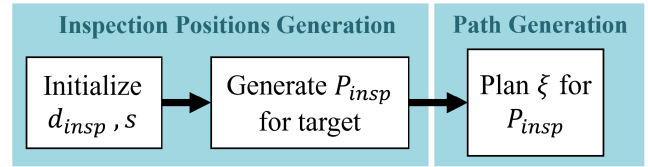


FIGURE 1. Our previous algorithm determines the ideal inspection distance d_{insp} and the corresponding inspection range s to generate P_{insp} , then it plans a collision-free ξ .

between the camera's optical axis and the surface normal. The UAV is modeled as a sphere with a volume V_{UAV} , which is used for the collision check of the planned path.

B. INSPECTION PATH PLANNING

Fig. 1 illustrates the framework of our previous algorithm [33], [34]. It consists of two modules, the inspection positions generation module and the path generation module.

1) INSPECTION POSITIONS GENERATION

Fig. 2 illustrates the complete workflow of generating P_{insp} . Our previous algorithm determines the ideal inspection distance d_{insp} and the corresponding sensor range s for collecting data at G^* in advance (Fig. 2(a)). Then it decomposes (Fig. 2(b)) and recombines M_{tar} into a minimal set of patches and plans one p for every patch to acquire one image (Fig. 2(c)).

The relationship between the actual GSD G and the object distance d can be modeled as Eq. (1) when the sensor faces directly opposite to the object. Therefore, d_{insp} can be determined as Eq. (2) according to G^* and camera specification. Every p will be planned at d_{insp} away from each patch to respect the resolution requirement.

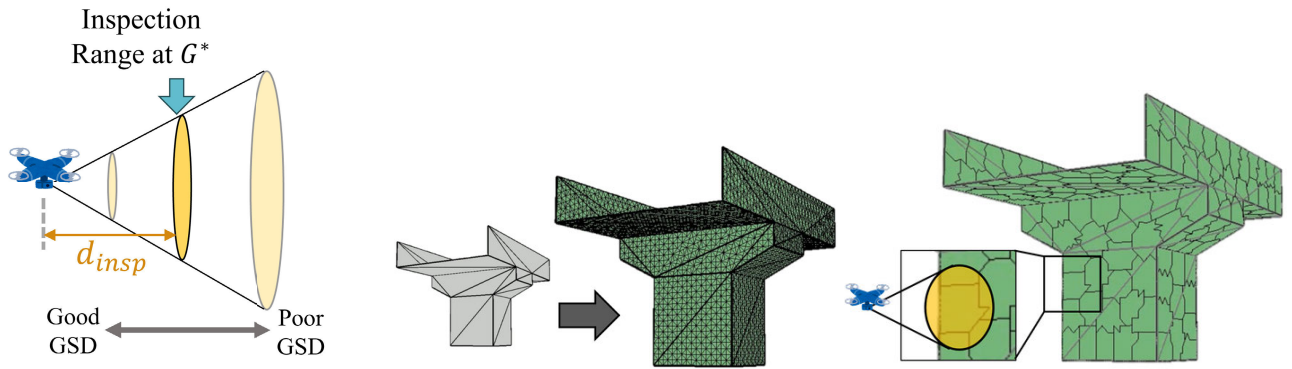
$$G = \frac{d \cdot L}{f \cdot I} \tag{1}$$

$$d_{insp} = \frac{G^* \cdot f \cdot I}{L} \tag{2}$$

Once the d_{insp} is determined, the inspection range s (Fig. 3) will be determined. It indicates how large a region the camera can inspect at d_{insp} , a shorter d_{insp} brings smaller s . Although the actual inspection region is a rectangle (height $H = G^* \cdot I$, width W), we define the s as an oblate inside the rectangle to filter the invalid region caused by lens distortion. The semi-major axis r_{maj} of s is determined by H and α as Eq. (3), and the semi-minor axis r_{min} is proportional to r_{maj} , a low ratio is sufficient for flat target surfaces. The $\alpha \in (0, 1]$ is a parameter that controls the overlap between adjacent photos, users can adjust it to prevent the data omission caused by E_{pos} or further filter the distorted part in photos.

$$r_{maj} = \alpha \cdot H/2 \tag{3}$$

Next, M_{tar} is subdivided into polygons M_{sub} tiny enough to be treated as points. Our method clusters the M_{sub} into k clusters $C = \{c_1, c_2, \dots, c_k\}$ using k-mean++ method [53]



(a) Determine the inspection distance and in- (b) Subdivide the target model into tiny poly- (c) Cluster the tiny polygons into clusters smaller than the inspection range

FIGURE 2. In the inspection positions generation phase, our algorithm decomposes and recombines the inspection target model into patches according to G^* and camera specification.

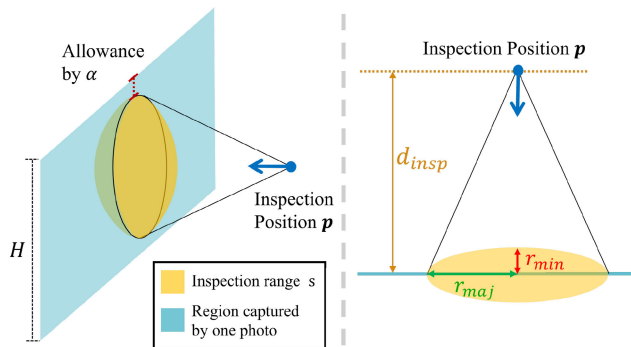


FIGURE 3. The sensor's inspection range s is defined as the yellow oblate intersecting the real image plane (azure) at d_{insp} .

as shown in Algorithm 1. One $p_j(j \in [1, k])$ will be planned for each $c_j(j \in [1, k])$ in the mean normal direction of the polygons in c_j , d_{insp} away from the center of c_j . Then our algorithm checks if all polygons in c_j are contained in the inspection range of p_j , and if the view angle of each polygon is below β_{max} . If not, the algorithm will increase the k and repeat the above steps until all polygons in M_{sub} are successfully inspected.

2) PATH GENERATION

The planning of ξ is treated as a TSP (Travelling Salesman Problem) with obstacle constraints. It is a classic problem and there are numerous existing methods [54], [55], [56], [57] available for addressing it. Optimizing ξ is not the focus of our work in this manuscript although it is crucial in practice. Therefore, we simply adopted an existing collision-free path planning method [58] to plan ξ for demonstration in our previous work [33], [34]. If necessary, ξ can be converted into the format in \mathbb{R}^3 using the information of GCP.

IV. INSPECTION EXPERIMENTS

This section introduces the experiment contents, conditions, and the UAV inspection platform we used for implementing

Algorithm 1 Inspection Position Generation

Input: $M_{tar}, G^*, L, I, f, \alpha, \beta_{max}$

Output: $P_{insp} = \{p_1, p_2, \dots, p_N\}$

- 1: $d_{insp}, s, r_{maj}, r_{min} \leftarrow Initialize(G^*, L, I, f, \alpha)$
- 2: $M_{sub} \leftarrow Subdivision(M_{tar})$
- 3: Initialize the k
- 4: **while** True **do**
- 5: $C = \{c_1, c_2, \dots, c_k\} \leftarrow Clustering(M_{sub}, k)$
- 6: $p_j(j \in [1, k]) \leftarrow Determine(C, d_{insp})$
- 7: **if** all polygons in M_{sub} are inspected **then**
- 8: $P_{insp} = \{p_j | j \in [1, k]\}$
- 9: Breaks
- 10: **end if**
- 11: $k++$
- 12: **end while**

our path planning method. Then, we demonstrated how to prepare paths using our method for an actual UAV inspection. The details of parameter tuning and countermeasures for countering the disturbance factors are provided.

A. EXPERIMENT CONTENTS

The verification experiments were performed on a real-size bridge model (Fig.4) in the Fukushima robot test field. A prepared STL model of this bridge is available for path planning. There are test pieces(Fig.5) sporadically embedded on the bridge surface for simulating various cracks and other deterioration defects. The crack widths and their positions are known.

We select two surfaces on the bridge (Fig.6) as the target of the experiment 1 and 2 respectively. Although our method can plan a 3D inspection path for the entire bridge, it is impossible to execute it in one flight due to the battery limitation. The inspection of a large-scale target would have needed to be divided into several tasks in advance and the path for each task should planned respectively. Besides, this bridge consists of a

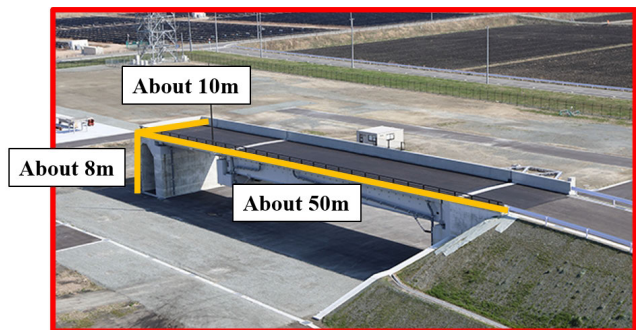


FIGURE 4. A real-size bridge model is built in the test field for the inspection experiments using robots).



FIGURE 7. Our UAV inspection system uses a total-station-based localization system to enable UAV executing flight paths in GNSS-denied environments and collect photos.

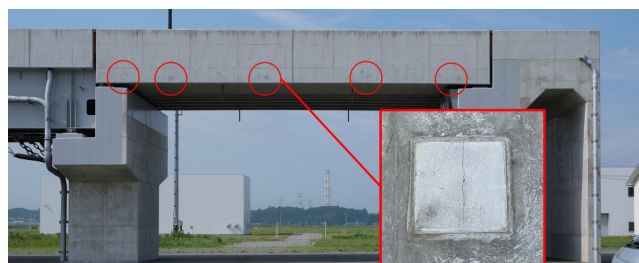


FIGURE 5. Examples of crack test pieces on experiment 2 surface.

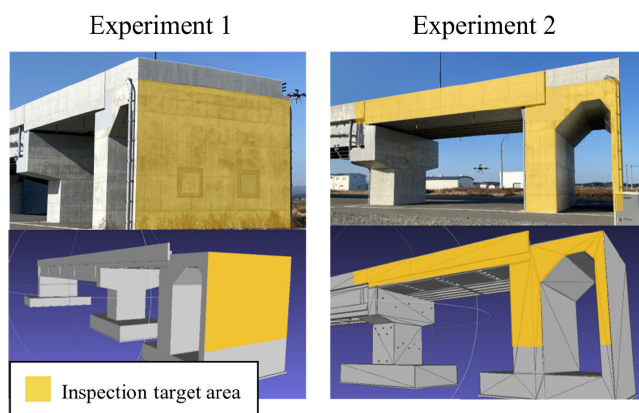


FIGURE 6. The positions of inspection target surfaces on STL model and real bridge are shown.

set of planar surfaces, inspecting other surfaces is essentially the same as repeating these experiments. Therefore, we did not inspect the entire bridge and these two experiments are considered sufficient to verify our method.

The experiment goal is to collect the image sets of target surfaces at $G^* = 0.3 \text{ mm/pixel}$ and the coverage rate requirement is 100%. According to the case study [29], [30], [31], the images should be collected with the GSD lower than $3 \sim 5$ times as the width of the crack desired to be detected. Thus, we set such a conservative G^* to ensure that those cracks wider than 0.1 mm are detectable. Considering the effect of possible E_{pos} , the data will be collected at 0.316 mm/pixel in the worst case.

Note that there are some inconsistencies between M_{tar} and the actual bridge which was previously unknown till we

arrived at the field. The inspection paths and data analysis are based on M_{tar} .

B. UAV INSPECTION SYSTEM

The UAV inspection system we integrate to execute inspection data collection is shown in Fig.7. This UAV platform is developed by DENSO Corporation [59], it is specialized for topographical surveying and bridge inspection. It has six motors whose rotation axis directions can be controlled individually. It can tilt the motors for greater resistance to side winds, enabling stable autonomous flight and spot inspections in windy environments (gusts of up to 10 m/s). A commodity camera (Sony a6000) is mounted 280 mm over the center of the UAV, its direction is controlled by a camera gimbal. We adopted a 50 mm lens with mere distortion. The specifications of the UAV and camera are summarized in Table 1.

Generally, UAVs need GNSS (Global Navigation Satellite System, e.g., GPS) to localize themselves and execute flight paths. However, the GNSS signal will be blocked and reflected near the constructions [60]. Thus, we used our total-station-based localization system [12] instead of GNSS to realize highly accurate UAV position estimation with several centimeters. The localization system is set on the position where the UAV is always visible (Fig.7). A ground station is used to process the planned path files and control the UAV to execute the paths.

It should be noted that navigating the UAV to arrive at a point exactly is impossible in practice. As an alternative, the UAV will be considered to have arrived at a waypoint if it has kept a distance from the waypoint lower than a threshold for a short while. In this study, we set the threshold to 200 mm for the balance positioning accuracy and operation efficiency. Correspondingly, an inevitable E_{pos} of up to 200 mm is introduced, which is the most probable reason that finally caused data omission in experiments.

C. PATH PLANNING FOR IMPLEMENTATION

This section introduces the details of planning inspection paths for experiments 1 and 2. User-defined parameters α and β_{max} are necessary for path planning. We set two α values as

TABLE 1. Specification of UAV and camera.

UAV specification		
Speed Limitation	Battery Duration	UAV Volume V_{UAV}
set to 0.6 m/s	set to 8 min	sphere with 1 m radius
Camera specification		
Sensor Size	Resolution	Focal Length f
25.1×16.7 mm	6000×4000 pixels	50 mm

TABLE 2. Inspection path planning parameters.

	$\alpha = 0.8$	$\alpha = 0.5$
d_{insp}	3.84 m	
H	1.20 m	
W	1.60 m	
r_{maj}	0.48 m	0.30 m
r_{min}	0.048 m	0.030 m

0.8 and 0.5 respectively to counteract the effects of E_{pos} and prevent data omission. We expected that the $\alpha = 0.5$ could ensure full coverage of the target in the worst case. The $\alpha = 0.8$ is a value that balances the inspection position quantity and the data overlap. We did not further narrow the α for distortion filtering since the lens used has a negligible distortion. The β_{max} is unnecessary to be considered since the inspection targets all planar surfaces. All the necessary parameters d_{insp} , H , W , and the size of s (r_{maj} and r_{min}) are determined as shown in Table 2.

The P_{insp} and ξ are planned according to the above parameters. It took about 5 minutes for the path planning of experiment 1 ($\alpha = 0.8$) and less than 5 hours for experiment 2 ($\alpha = 0.5$) using the processor CPU i7-7700K (4.2Ghz). The runtime is considered acceptable for the practical inspection operation. Some adjustments to P_{insp} and ξ are adopted for implementing them in the real world. The complete preparation process for experiments is shown in Fig.8.

In the real world, unpredictable positioning failure and wind disturbance [61] are inevitable and they may result in collisions. Therefore, we excluded those inspection positions in P_{insp} within 1 m above the ground for safety. The corresponding areas of excluded positions will not be considered in the data evaluation. Therefore the actual target areas are slightly different from Fig.6 and their bottom edges are uneven as shown in the subsection V-B later.

Besides, the execution time of paths is also significant to practical inspection. Therefore, we further optimized the planned ξ using the 2-opt algorithm [62] after the original path planning in section III-B2. The path lengths were shortened by up to half to respect the battery duration limitation. Under these adjustments, the details of the paths for final implementation are listed in Table 3.

The coordinates in \mathbb{R}^3 of 22 distinguishable corners (some of them are shown in Fig.9) on the bridge are previously

given. We associated the corresponding points on M_{tar} and used these point pairs as GCP in path conversion. The Helmert transformation [63] and the open-source cartographic projections and coordinate transformations library “pyproj” [64] were used. Besides, since the ξ is essentially the trajectory that the camera should reach, we modified it into UAV flight paths. An offset was added to all ξ to cope with the difference between the camera position and the UAV center.

V. RESULTS AND DISCUSSION

All the experiments and paths were successfully performed. The achieved coverage rate and the object distances at which the photos are collected are summarized in Table 4. The results show that our method [33], [34] can collect data at G^* in practice, yet 100% coverage is not achieved. We infer that there are two major reasons for the omissions. The first one is some biases were introduced to the paths in the coordinate conversion phase. The second reason is that the setting of α was inappropriate so the effect of E_{pos} was not well counteracted. Through the experiments, we identified that the E_{pos} is the crucial disturbance factor that hinders desired data collection, and the shortage of our method in coping with E_{pos} is also reflected.

A. DATA QUALITY EVALUATION

Since it is impossible to precisely calculate the G for every pixel of each collected photo, in this study the quality of each dataset is estimated by inspection distances shown in Table 4 and the visibility of cracks on test pieces.

We calculated the G of the collected photos based on their inspection distance and Eq.(1), the GSD ranged from 0.278 to 0.316 mm/pixel and did not exceed the tolerance (up to 0.316 mm/pixel is allowed). The mean inspection distances are very close to $d_{insp} = 3.84$ m. Fig.10 shows a 150 mm wide test piece with a 0.3 mm crack extracted from collected photos. Its edge is supposed to occupy 500 pixels at $G^* = 0.3$ mm/pixel, but it occupies 502 pixels on the collected image, which means the G of this photo is better than G^* . Besides, the 0.3 mm crack also does not exceed one line of pixels. These quantitative metrics indicate that the G of the collected photos are not poorer than G^* .

Because our essential purpose is benefiting defect detection, we further analyzed the collected photos that contain crack test pieces. At $G^* = 0.3$ mm/pixel, the cracks with

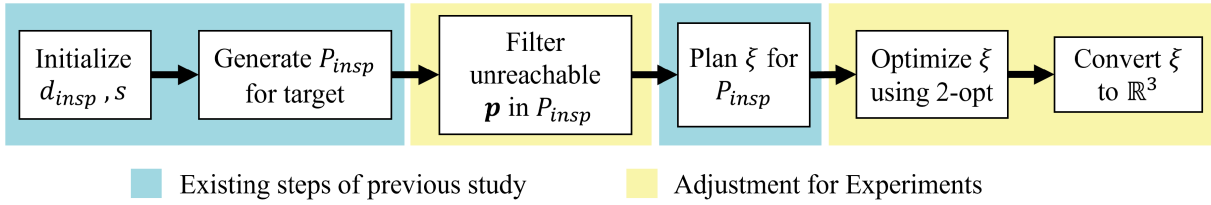


FIGURE 8. The flow of paths preparation for implementation is shown, some adjustments (yellow) are inserted in the original steps (azure) of our method.

TABLE 3. Details of implemented inspection paths.

	Experiment 1		Experiment 2	
	$\alpha = 0.8$	$\alpha = 0.5$	$\alpha = 0.8$	$\alpha = 0.5$
Inspection positions N	140	368	171	385
Path length	93.1 m	153.4 m	97.9 m	145.8 m
Area of target	73 m ²		91 m ²	

TABLE 4. Implementation results of experiments.

	Experiment 1		Experiment 2	
	$\alpha = 0.8$	$\alpha = 0.5$	$\alpha = 0.8$	$\alpha = 0.5$
Coverage achieved	92.51%	98.32%	99.22%	99.84%
Mean inspection distance	3.82 m	3.83 m	3.90 m	3.87 m
Maximum inspection distance	3.99 m	4.02 m	4.04 m	4.02 m
Minimum inspection distance	3.56 m	3.64 m	3.74 m	3.71 m

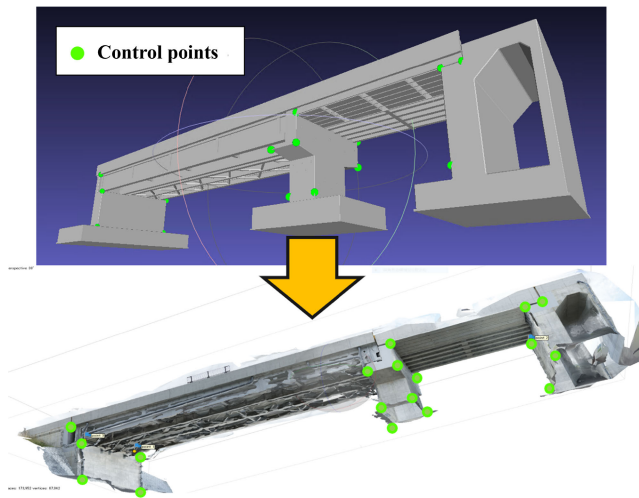


FIGURE 9. Some corners (green) on the bridge are selected and measured as the ground control points, part of them are shown in figure.

widths down to 0.06 ~ 0.1 mm are supposed to be detectable in the photos [29], [30], [31]. Fig.11 shows some samples of inspected cracks. The 0.3 mm and 0.2 mm cracks are clearly visible as expected. Even the 0.05 mm crack is visible, which is almost the limit of G^* . We checked all the photos containing cracks and confirmed that the cracks are all visible.

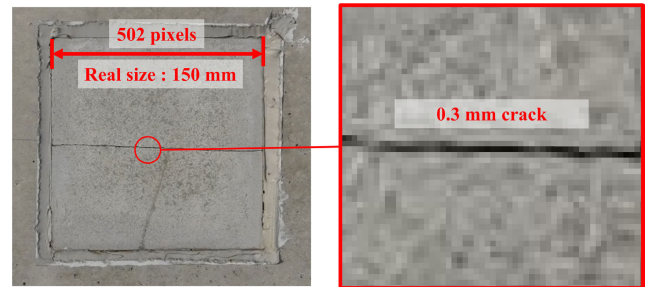


FIGURE 10. The edge of the test piece (150 mm wide) occupies 502 pixels, which means the GSD of photos is 0.299 mm/pixel and is better than G^* .

Based on the above evidence, we have confirmed that the collected data met the GSD requirement and the defects in data are detectable. Our method [33], [34] can ensure the data quality.

B. COVERAGE EVALUATION

The coverage of each dataset is defined as the ratio of the inspected area and the target area. We extracted the valid part (the round area) from every collected image and then stitched them together. The achieved coverage rates of all datasets are shown in Table 4. The target and inspected areas are shown in Fig.12 and Fig.13, for experiments 1 and 2 respectively.

Although our CPP method guarantees full coverage in theory, omissions occurred in all datasets. We classify the

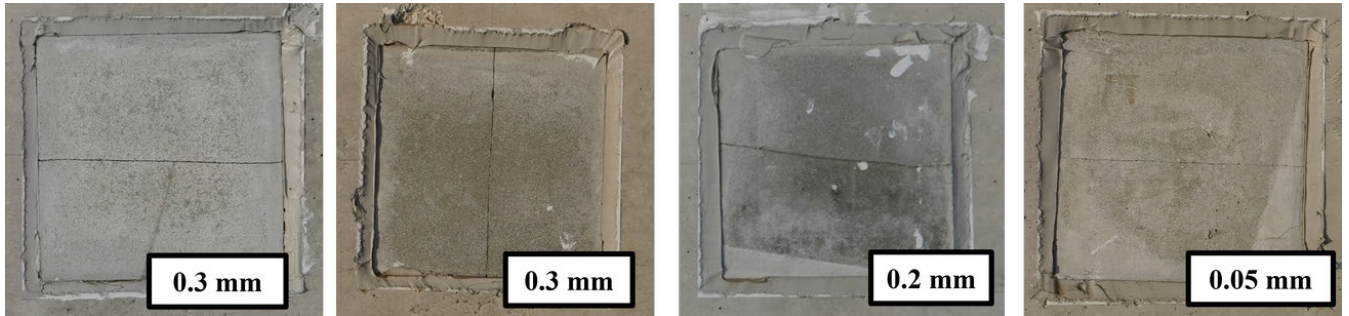


FIGURE 11. Captured cracks on collected photos are clearly visible, which indicates the GSD of data satisfied requirements.

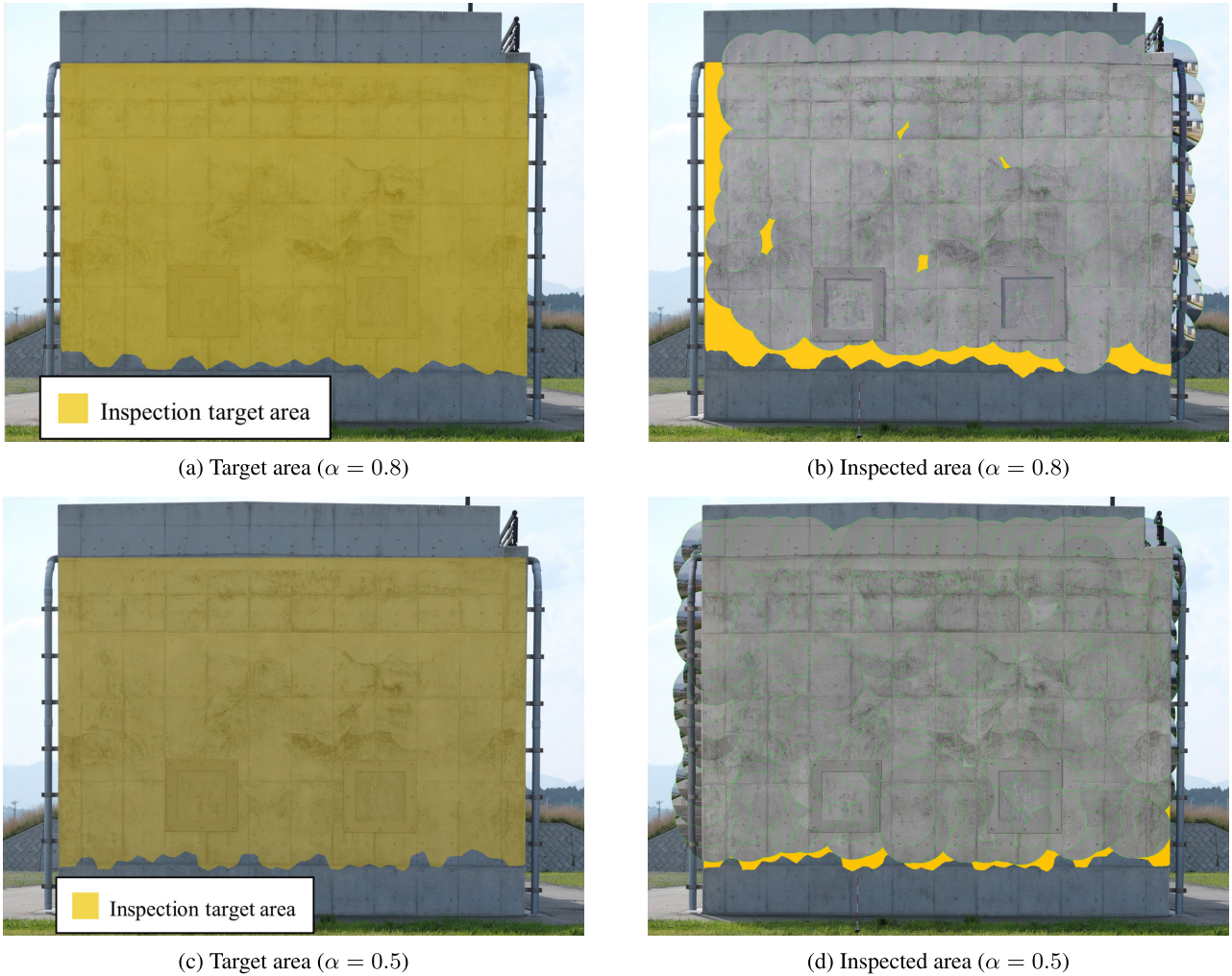


FIGURE 12. Coverage results of experiment 1, the valid parts of collected photos are stitched together and overlaid on inspection target.

omissions as two patterns, the banded distributed omissions on the edges and scattered omissions. The former is considered caused by systematic errors and the latter indicates the shortage of this CPP method.

The banded distributed omissions exist in all results. In Fig. 12, the top area is inspected though it does not belong to the target area, while omission occurs on the bottom. Similarly, in Fig. 13, the distribution of the background in the stitched image is unbalanced. The upper-right area

that does not exist in the model M_{tar} has been almost completely inspected. They are probably caused by the integral biases of inspection paths relative to the actual bridge. The possible reasons may be coordinate conversion errors or localization system initialization errors, rather than the design flaws of the path planning method. Since we did not know the M_{tar} is slightly different from the real bridge before experiments, we may have incorrectly associated the corners on M_{tar} and those on the real bridge (see section IV).

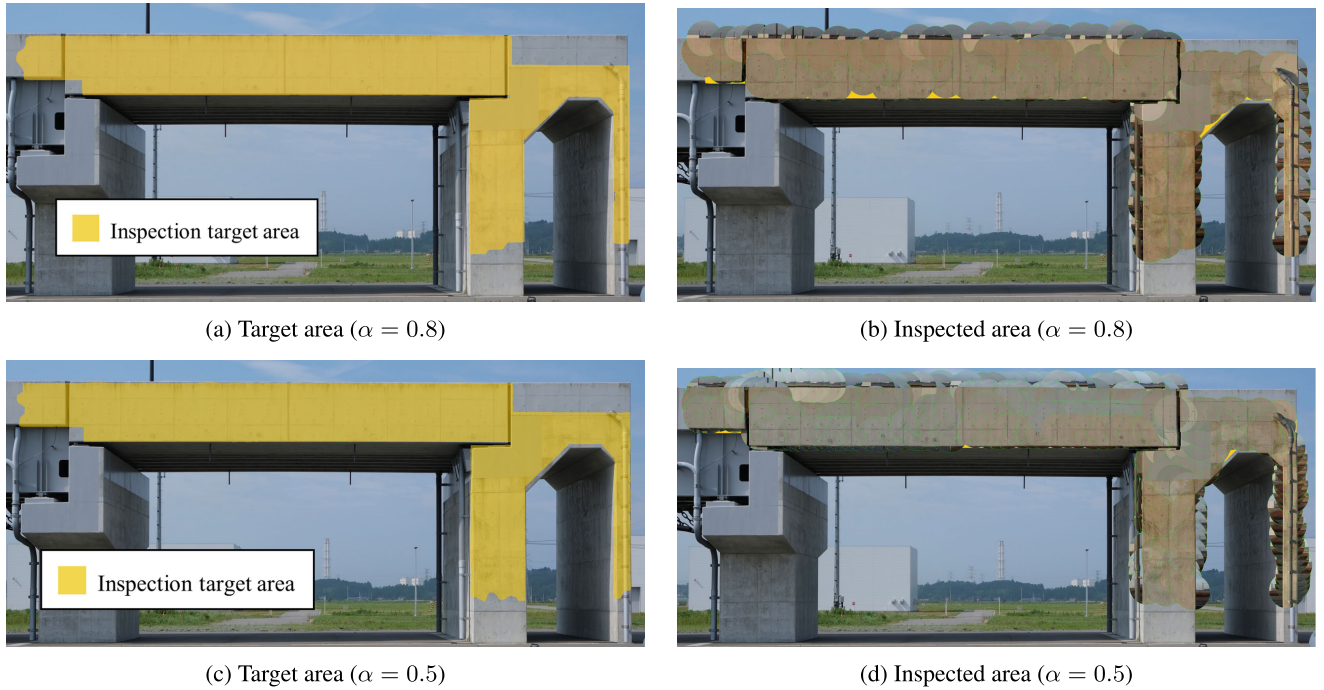


FIGURE 13. Coverage results of experiment 2, the valid parts of collected photos are stitched together and overlaid on inspection target.

Thus, the wrong GCP information resulted in biases in path conversion.

Regarding the scattered omissions, the scattered omissions only occur in the results of $\alpha = 0.8$. The prerequisite for nonomission inspection is that the occurred E_{pos} should be lower than the allowance introduced by α . The α is set as 0.8 and 0.5, which corresponds to allowances of 120 and 300 mm respectively. However, the problem in waypoint arriving judgment may cause an E_{pos} of up to 200 mm as we mentioned in section IV-B. It is reasonable to infer that the actually occurred E_{pos} in experiments exceeded the allowance of $\alpha = 0.8$ and it finally caused the omission. On the contrary, the paths of $\alpha = 0.5$ almost achieve full coverage as expected.

In conclusion, based on the results of $\alpha = 0.5$, we confirmed that our method [33], [34] can achieve full coverage in the practical inspection. The strict 100% coverage can be achieved as long as the systematic errors are eliminated and E_{pos} 's effect is canceled by setting appropriate α . However, setting an appropriate α for each specific case is very challenging, inappropriate α will cause data omission or redundant data. Since E_{pos} 's effect has not been modeled before, there is no guideline for determination α and its setting only depends on experiences and trials. This shortage ruins the method's utility in practice and further enhancement is necessary.

VI. ENHANCED INSPECTION PATH PLANNING

A. ADAPTIVE DETERMINATION OF THE INSPECTION DISTANCE AND RANGE

Through the practical inspection, our previous method's shortage is confirmed. Users must set an appropriate α to

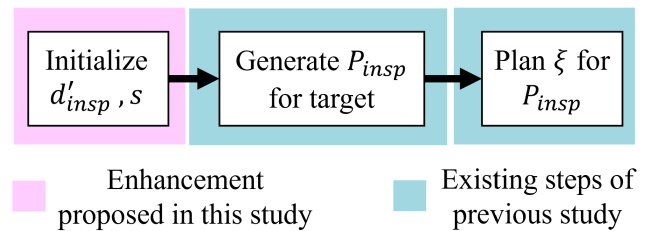


FIGURE 14. The enhanced method optimizes the inspection distance and range, and it plans P_{insp} and ξ like the previous method.

adjust the data overlap and prevent omission caused by E_{pos} while it is impractical. To realize stable collections of qualified inspection data, we modeled the effects of those inevitable positioning and orientation errors in practice and then enhanced our CPP method to handle them.

Fig. 14 illustrates the logic of the enhanced method. The only difference between the enhanced and previous methods is the initialization, the rest steps of these two methods are the same. The enhanced method discards the α . It adaptively optimizes the inspection distance and range using a novel guideline to cancel the effects of the above errors. As a result, data completeness and quality can be strictly ensured.

We expand the problem statement based on the practical experiments as follows. We suppose that when the camera visits an inspection position p , the actual camera position may fall within or on an ellipsoidal space centered at p due to a limited E_{pos} . The ellipsoid has semi-axes of lengths a and b that are parallel to the inspection surface ($a \geq b$), and the length of the semi-axis perpendicular to the surface is denoted

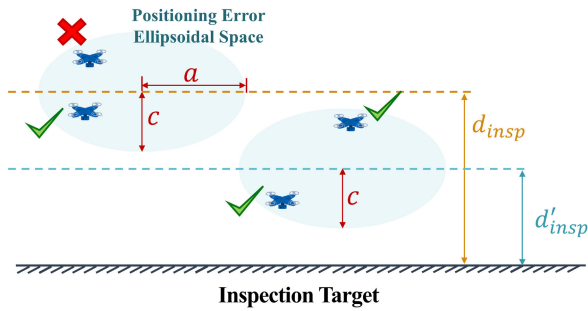


FIGURE 15. Adjustment of inspection distance is introduced to eliminate the effects of positioning error.

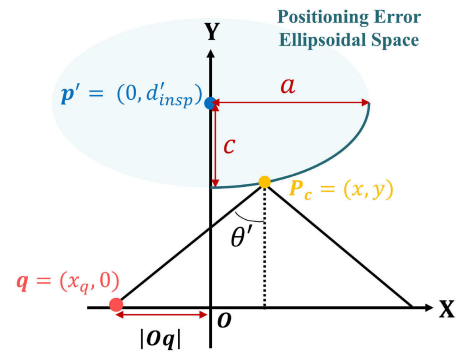


FIGURE 17. The minimum of $|Oq|$ can be found on $a - c$ axes plane.

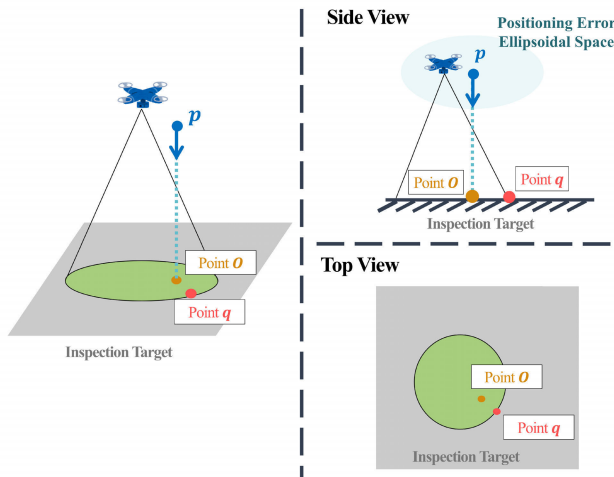


FIGURE 16. We define q as the point on the edge of the actually inspected region (green) that is closest to the projection O of p on the target surface when positioning error occurred.

as c . Besides, there exists an unpredictable orientation error E_{ori} in the range of $[0, \Delta\theta]$ between the planned direction and the real direction in which the camera takes a photo. Our goal is still to inspect given targets at G^* without omission in such a condition.

The first change of the enhanced method is that we shortened the inspection distance to strictly respect the resolution requirements in such a condition, as shown in Fig.15. In the previous method, the inspection distance d_{insp} is determined as Eq.(2), it is the distance that is just enough for G^* . Even a slight E_{pos} in the direction of the optical axis or a slight E_{ori} can affect the quality of data. Therefore, we updated the inspection distance d'_{insp} as Eq.(4) to ensure the data GSD in even the worst case. Since the GSD deteriorates with the changes in view angle [32], a term $\cos(\Delta\theta)$ is introduced to compensate for E_{ori} .

$$d'_{insp} = (d_{insp} - c) \cdot \cos(\Delta\theta) \quad (4)$$

Correspondingly, the s of one p that can be inspected by a single photo at the new distance d'_{insp} should also be modified for full coverage path planning (see section III-B).

TABLE 5. Inspection simulation conditions.

Coverage requirement	100%
GSD requirement G^*	0.3 mm/pixel (up to 0.316 mm/pixel is allowed)
Sensor size L	16.7 mm
Resolution I	4000 pixels
Focal length f	50 mm
α	0.8 & 0.5
semi-axes length a	200 mm
semi-axes length b	200 mm
semi-axes length c	200 mm

Considering an inspection position p is planned d'_{insp} away from the inspection surface. The region actually inspected by p changes according to the happened E_{pos} and E_{ori} . Thus, the s should be defined in the planning phase as the region that will always be inspected regardless of how E_{pos} and E_{ori} change.

We simplified the definition of new s to an extremum problem. Firstly, we replace the original camera FOV $2\theta = 2 \arctan(L/F)$ by a narrowed FOV $2\theta' = 2(\theta - \Delta\theta)$ to eliminate the effects of E_{ori} . The region inspected by the camera will always be round and its center and radius depend on the camera's real position. Let point O be the projection of p on the surface, the q is the closest point to O on the edge of the inspected region (Fig.16). q varies according to the positioning error that actually occurred. We can define the r_{maj} of s as the minimum of $|Oq|$ and plan paths as before to ensure full coverage.

Obviously, $|Oq|$ decreases when the actual camera position is further away from p in the parallel direction of the surface or is closer to the surface. Therefore, only those camera positions simultaneously exist on the positioning error ellipsoidal's surface and its $a - c$ axes plane need to be discussed. We consider such a planar coordinate system (Fig.17). The origin is O and the X-axis is parallel to the major axis of the positioning error ellipsoidal. $p' = (0, d'_{insp})$ denotes as the projection of p on $a - c$ axes plane, the $q = (x_q, 0)$ locates on X-axis. The worst case of camera position $p_c = (x, y)$ corresponding to minimum $|Oq|$ exists on the elliptical arc of $x \in [0, a]$, $y \in [0, c]$ (satisfies Eq.(5)).

TABLE 6. Simulation results (experiment 1).

	Previous ($\alpha = 0.8$)	Previous ($\alpha = 0.5$)	Proposed
Full coverage count	672/1000	998/1000	1000/1000
Inspection positions number	170	444	285

TABLE 7. Simulation results (experiment 2).

	Previous ($\alpha = 0.8$)	Previous ($\alpha = 0.5$)	Proposed
Full coverage count	725/1000	998/1000	1000/1000
Inspection positions number	287	655	468

TABLE 8. Details of parameters for path planning.

	Previous ($\alpha = 0.8$)	Previous ($\alpha = 0.5$)	Proposed
Inspection distance	$d_{insp} = 3.84$ m	$d_{insp} = 3.84$ m	$d'_{insp} = 3.64$ m
Inspection range r_{maj}	0.48 m	0.30 m	0.37 m

The x_q can be expressed as a function $f(x)$ about x , as Eq.(6). The minimum $|\mathbf{O}q|$ can be found by differentiating the Eq.(6). It is worth noting that, counter-intuitively, the camera position corresponding to the minimum $|\mathbf{O}q|$ does not always be the two ends of the elliptical arc, $(0, d'_{insp} - c)$ and (a, d'_{insp}) . Finally, the new r_{maj} is defined as Eq.(7). By planning the inspection positions using the d'_{insp} and new r_{maj} in the existing CPP framework, the effects of E_{pos} and E_{ori} can be complete eliminated full coverage and data quality can be strictly ensured in theory.

$$\frac{x^2}{a^2} + \frac{(y - d'_{insp})^2}{c^2} = 1 \quad (5)$$

$$x_q = f(x) = x + c \cdot \tan(\theta') \cdot \sqrt{1 - \frac{x^2}{a^2}} - d'_{insp} \cdot \tan(\theta') \quad (6)$$

$$r_{maj} = d'_{insp} \cdot \tan(\theta') - \sqrt{a^2 + c^2 \cdot \tan^2(\theta')} \quad (7)$$

when $\frac{df(x)}{dx} = 0$

B. SIMULATIONS

We compared the performance of the previous and enhanced CPP method through the inspection simulation. The simulation conditions are set as close as possible to our real experiments to acquire objective results. The inspection targets are those in Fig.6 and the simulation conditions are summarized in Table 5, which is the same as the experiments. To simulate the E_{pos} we encountered in experiments, a Gaussian distribution ($\mu = 0, 3\sigma = 200$ mm) is added to the actual UAV position to replicate the waypoint arriving judgment problem. The semi-axes lengths a, b, c of the enhanced method are set to $3\sigma = 200$ mm correspondingly.

We planned the inspection paths respectively and the simulation results of two targets are summarized in Table 6 and Table 7. Each path is repeated one thousand times and the count of achieving full coverage is provided. The positioning errors varied every time.

Obviously, compared to the previous method's two cases, the enhanced method can achieve full coverage better. Similar to the result of practical experiments, the previous method ($\alpha = 0.8$) failed to achieve 100% coverage in many cases. Reducing the α to 0.5 helps increase the full coverage count yet omission still exists. On the other hand, the enhanced method is more robust to E_{pos} since its effects are well quantified and counteracted, it achieved the goal of full coverage every time.

Furthermore, the enhanced method realized full coverage with fewer inspection positions. The inspection distance and inspection range determined for each condition are shown in Table 8. In the previous method, the user has to significantly narrow the inspection range to avoid data omission. In the case of $\alpha = 0.5$, the inspection range is reduced to the half of camera's real range and the inspection positions increase over 2 times. On the contrary, the enhanced method adaptively optimizes both the inspection range and distance according to inspection conditions, so that it only needs to introduce the bare minimum of data overlap necessary for the E_{pos} .

In conclusion, the enhancement properly solved the challenge of E_{pos} and made our CPP method more suitable for practical inspection.

VII. CONCLUSION

In this study, we implemented our resolution-aware path planning to collect the inspection data of a bridge to validate its practical effectiveness. The results show that this method

basically delivers what it promises in theory, it can ensure the GSD of inspection data. However, the inevitable positioning errors in practice led to data omission. The positioning errors can not be well handled in the previous method's framework. Therefore, we enhanced this path planning method to realize adaptively adjustment of the inspection path. Through the simulation, we confirmed that the enhanced method achieved stable data collection even though the positioning errors existed. The enhanced method prevented omission while minimizing data redundancy.

On the other hand, there still remain some limitations of our path planning methods. How to shorten the total running time by approaches such as subdividing the path planning tasks or setting the k appropriately is a valuable topic. Another limitation of our method framework is that it can only plan inspection positions directly opposite the target, which is not suitable for inspection when the workspace is narrow. This problem can be addressed by inspecting from a tilted direction with a higher zoom ratio, yet no CPP method can realize such flexible planning. Meanwhile, it is confirmed that the UAV took too long to reach one waypoint, further reducing the number of inspection positions will significantly benefit the inspection efficiency. Fusing adjacent inspection positions into one and planning multiple inspection directions is a feasible choice.

In addition, the optimization of the inspection paths is a crucial problem that needs further discussion. Various disturbances in the inspection environment need to be considered, such as the spatial-variant wind field, the changing light conditions, the influence of the steel structural parts on the UAV's magnetometer, etc. How to model them in the cost function for path optimization is a crucial challenge. We are going to address these challenges in the future works.

ACKNOWLEDGMENT

The author Weitong Wu would like to express his sincere gratitude to Prof. Yuki Funabara, Prof. Shinji Doki, and Prof. Kae Doki, who have been closely involved and provided unwavering support throughout the entire research and manuscript preparation process. Their stringent requirements and meticulous guidance have been instrumental in the successful publication of this article.

He would also like to thank Satoru Yoshikawa, Tetsuji Mitsuda, and Jingyu Xiang, who have offered substantial support and valuable insights during the research phase. Without their assistance, this study could not have been completed.

REFERENCES

- [1] S. Bang, Y. Hong, and H. Kim, "Proactive proximity monitoring with instance segmentation and unmanned aerial vehicle-acquired video-frame prediction," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 36, no. 6, pp. 800–816, Jun. 2021.
- [2] B. Alsadik and F. Nex, "The rise in UAV inspections for civil infrastructure," GIM Int., Twente, The Netherlands, Tech. Rep., 2021.
- [3] M. Rashidi and B. Samali, "Health monitoring of bridges using RPAs," in *EASEC16*, Springer, 2021, pp. 209–218.
- [4] J. L. Wells, B. Lovelace, and T. Kalar, "Use of unmanned aircraft systems for bridge inspections," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2612, no. 1, pp. 60–66, Jan. 2017.
- [5] X. Zhong, X. Peng, S. Yan, M. Shen, and Y. Zhai, "Assessment of the feasibility of detecting concrete cracks in images acquired by unmanned aerial vehicles," *Autom. Construct.*, vol. 89, pp. 49–57, May 2018.
- [6] S. Chen, D. F. Laefer, E. Mangina, S. M. I. Zolanvari, and J. Byrne, "UAV bridge inspection through evaluated 3D reconstructions," *J. Bridge Eng.*, vol. 24, no. 4, Apr. 2019, Art. no. 05019001.
- [7] J. Seo, L. Duque, and J. Wacker, "Drone-enabled bridge inspection methodology and application," *Autom. Construct.*, vol. 94, pp. 112–126, Oct. 2018.
- [8] S. Dorafshan, R. J. Thomas, and M. Maguire, "Fatigue crack detection using unmanned aerial systems in fracture critical inspection of steel bridges," *J. Bridge Eng.*, vol. 23, no. 10, Oct. 2018, Art. no. 04018078.
- [9] A. P. Tomiczek, T. J. Whitley, J. A. Bridge, and P. G. Ifju, "Bridge inspections with small unmanned aircraft systems: Case studies," *J. Bridge Eng.*, vol. 24, no. 4, Apr. 2019.
- [10] C. J. O. Salaan, Y. Okada, S. Mizutani, T. Ishii, K. Koura, K. Ohno, and S. Tadokoro, "Close visual bridge inspection using a UAV with a passive rotating spherical shell," *J. Field Robot.*, vol. 35, no. 6, pp. 850–867, Sep. 2018.
- [11] T. Ikeda, S. Minamiyama, S. Yasui, K. Ohara, A. Ichikawa, S. Ashizawa, A. Okino, T. Oomichi, and T. Fukuda, "Stable camera position control of unmanned aerial vehicle with three-degree-of-freedom manipulator for visual test of bridge inspection," *J. Field Robot.*, vol. 36, no. 7, pp. 1212–1221, Oct. 2019.
- [12] M. Hirai and Kato, "Precise autonomous flight of UAV under the non-GNSS," *J. Jpn. Soc. Mech. Eng.*, vol. 121, no. 11, pp. 24–25, 2018.
- [13] K. Kimura, K. Maeda, K. Asa, Y. Funabara, S. Doki, and K. Doki, "UAV localization system with subsidiary UAVs for visual inspection," *Trans. Soc. Instrum. Control Eng.*, vol. 56, no. 7, pp. 370–378, 2020.
- [14] Y. Li, M. M. Karim, and R. Qin, "A virtual-reality-based training and assessment system for bridge inspectors with an assistant drone," *IEEE Trans. Hum.-Mach. Syst.*, vol. 52, no. 4, pp. 591–601, Aug. 2022.
- [15] Y. Hong, Y. Funabara, S. Doki, and D. Kae, "Efficiency improvement of infrastructure inspection by maneuvering assistance of an unmanned aerial vehicle using augmented reality," in *Proc. Robot. Mechatronics Conf.*, 2018, Paper no. 2A1-B04.
- [16] B. Batchelor and S. Cotter, "Detection of cracks using image processing algorithms implemented in hardware," *Image Vis. Comput.*, vol. 1, no. 1, pp. 21–29, Feb. 1983.
- [17] H.-G. Sohn, Y.-M. Lim, K.-H. Yun, and G.-H. Kim, "Monitoring crack changes in concrete structures," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 20, no. 1, pp. 52–61, Jan. 2005.
- [18] T. Yamaguchi and S. Hashimoto, "Fast crack detection method for large-size concrete surface images using percolation-based image processing," *Mach. Vis. Appl.*, vol. 21, no. 5, pp. 797–809, Aug. 2010.
- [19] S. Kong, J. Fan, Y. Liu, X. Wei, and X. Ma, "Automated crack assessment and quantitative growth monitoring," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 36, no. 5, pp. 656–674, May 2021.
- [20] X. Ji, Z. Miao, and R. Kromanis, "Vision-based measurements of deformations and cracks for RC structure tests," *Eng. Struct.*, vol. 212, Jun. 2020, Art. no. 110508.
- [21] I.-H. Kim, H. Jeon, S.-C. Baek, W.-H. Hong, and H.-J. Jung, "Application of crack identification techniques for an aging concrete bridge inspection using an unmanned aerial vehicle," *Sensors*, vol. 18, no. 6, p. 1881, Jun. 2018.
- [22] H. Chu, W. Wang, and L. Deng, "Tiny-crack-net: A multiscale feature fusion network with attention mechanisms for segmentation of tiny cracks," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 37, no. 14, pp. 1914–1931, Nov. 2022.
- [23] N. Gehri, J. Mata-Falcón, and W. Kaufmann, "Automated crack detection and measurement based on digital image correlation," *Construct. Building Mater.*, vol. 256, Sep. 2020, Art. no. 119383.
- [24] H. S. Munawar, A. W. A. Hammad, A. Haddad, C. A. P. Soares, and S. T. Waller, "Image-based crack detection methods: A review," *Infrastructures*, vol. 6, no. 8, p. 115, Aug. 2021.
- [25] R. G. Lins and S. N. Givigi, "Automatic crack detection and measurement based on image analysis," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 3, pp. 583–590, Mar. 2016.

- [26] M. Flah, A. R. Suleiman, and M. L. Nehdi, "Classification and quantification of cracks in concrete structures using deep learning image-based techniques," *Cement Concrete Compos.*, vol. 114, Nov. 2020, Art. no. 103781.
- [27] H. Wiggerhauser, C. Köpp, J. Timofeev, and H. Azari, "Controlled creating of cracks in concrete for non-destructive testing," *J. Nondestruct. Eval.*, vol. 37, no. 3, pp. 1–9, Sep. 2018.
- [28] K. Maruyama, "JSCE standard specifications for concrete structures 2007, codes in structural engineering-developments and need for international practice," in *Proc. Joint IABSE-Fib Conf.*, vol. 1, Dubrovnik, Croatia, 2010, pp. 315–322.
- [29] F. Ioli, A. Pinto, and L. Pinto, "UAV photogrammetry for metric evaluation of concrete bridge cracks," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 43, pp. 1025–1032, May 2022.
- [30] K. Keisuke and M. Hiroshi, "Verification of digital image resolution and visible crack width of concrete," *J. Civil Eng. Struct. Mater.*, vol. 35, pp. 115–122, 2019.
- [31] Y. Liu, X. Nie, J. Fan, and X. Liu, "Image-based crack assessment of bridge piers using unmanned aerial vehicles and three-dimensional scene reconstruction," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 35, no. 5, pp. 511–529, May 2020.
- [32] J. C. Leachtenauer and R. G. Driggers, *Surveillance and Reconnaissance Imaging Systems: Modeling and Performance Prediction*. Norwood, MA, USA: Artech House, 2001.
- [33] K. Asa, Y. Funabora, S. Doki, and K. Doki, "Evaluation in real world of the measuring position determination for visual inspection using UAV," in *Proc. 44th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2018, pp. 2711–2716.
- [34] K. Asa, "Path planning of UAV for measurement task in visual inspection of infrastructures," M.S. thesis, Nagoya Univ., Nagoya, Japan, 2020.
- [35] B. Hepp, M. Nießner, and O. Hilliges, "Plan3D: Viewpoint and trajectory optimization for aerial multi-view stereo reconstruction," *ACM Trans. Graph.*, vol. 38, no. 1, pp. 1–17, Feb. 2019.
- [36] S. Song, D. Kim, and S. Jo, "Active 3D modeling via online multi-view stereo," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 5284–5291.
- [37] H. Balta, J. Bedkowski, S. Govindaraj, K. Majek, P. Musialik, D. Serrano, K. Alexis, R. Siegwart, and G. De Cubber, "Integrated data management for a fleet of search-and-rescue robots," *J. Field Robot.*, vol. 34, no. 3, pp. 539–582, May 2017.
- [38] A. A. Meera, M. Popovic, A. Millane, and R. Siegwart, "Obstacle-aware adaptive informative path planning for UAV-based target search," in *Proc. Int. Conf. Robot. Autom.*, May 2019, pp. 718–724.
- [39] C. Wang, P. Liu, T. Zhang, and J. Sun, "The adaptive vortex search algorithm of optimal path planning for forest fire rescue UAV," in *Proc. IEEE 3rd Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Oct. 2018, pp. 400–403.
- [40] R. Almadhoun, T. Taha, L. Seneviratne, and Y. Zweiri, "A survey on multi-robot coverage path planning for model reconstruction and mapping," *Social Netw. Appl. Sci.*, vol. 1, no. 8, pp. 1–24, Aug. 2019.
- [41] M. Popović, T. Vidal-Calleja, G. Hitz, J. J. Chung, I. Sa, R. Siegwart, and J. Nieto, "An informative path planning framework for UAV-based terrain monitoring," *Auto. Robots*, vol. 44, no. 6, pp. 889–911, Jul. 2020.
- [42] S. Gong, M. Wang, B. Gu, W. Zhang, D. T. Hoang, and D. Niyato, "Bayesian optimization enhanced deep reinforcement learning for trajectory planning and network formation in multi-UAV networks," *IEEE Trans. Veh. Technol.*, 2023.
- [43] *DJI GS Pro*. Accessed: Nov. 2, 2023. [Online]. Available: <https://www.dji.com/ground-station-pro>
- [44] *Ardupilot Documentation*. Accessed: Nov. 2, 2023. [Online]. Available: <https://ardupilot.org/ardupilot/index.html>
- [45] *PIX4Dcapture Pro*. Accessed: Nov. 2, 2023. [Online]. Available: <https://www.pix4d.com/product/pix4dcapture/>
- [46] *Qgroundcontrol User Guide*. Accessed: Nov. 2, 2023. [Online]. Available: <https://docs.qgroundcontrol.com/master/en/>
- [47] R. Almadhoun, T. Taha, D. Gan, J. Dias, Y. Zweiri, and L. Seneviratne, "Coverage path planning with adaptive viewpoint sampling to construct 3D models of complex structures for the purpose of inspection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 7047–7054.
- [48] W. Jing, D. Deng, Y. Wu, and K. Shimada, "Multi-UAV coverage path planning for the inspection of large and complex structures," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 1480–1486.
- [49] M. D. Phung, C. H. Quach, T. H. Dinh, and Q. Ha, "Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection," *Autom. Construction*, vol. 81, pp. 25–33, Sep. 2017.
- [50] C. Peng and V. Isler, "Visual coverage path planning for urban environments," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5961–5968, Oct. 2020.
- [51] S. Ivić, B. Crnković, L. Grbčević, and L. Matleković, "Multi-UAV trajectory planning for 3D visual inspection of complex structures," *Autom. Construct.*, vol. 147, Mar. 2023, Art. no. 104709.
- [52] L. Roscoe, "Stereolithography interface specification," 3D Syst. Inc., Cary, NC, USA, Tech. Rep. 2020-10, vol. 27, 1988.
- [53] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," Stanford, CA, USA, Tech. Rep., 2006.
- [54] S. Huang, R. S. H. Teo, and K. K. Tan, "Collision avoidance of multi unmanned aerial vehicles: A review," *Annu. Rev. Control*, vol. 48, pp. 147–164, 2019.
- [55] A. Vashisth, R. S. Batth, and R. Ward, "Existing path planning techniques in unmanned aerial vehicles (UAVs): A systematic review," in *Proc. Int. Conf. Comput. Intell. Knowl. Economy (ICCIKE)*, Mar. 2021, pp. 366–372.
- [56] I. Khoufi, A. Laouiti, and C. Adjih, "A survey of recent extended variants of the traveling salesman and vehicle routing problems for unmanned aerial vehicles," *Drones*, vol. 3, no. 3, p. 66, Aug. 2019.
- [57] D. Popescu, F. Stoican, G. Stamatescu, O. Chenaru, and L. Ichim, "A survey of collaborative UAV-WSN systems for efficient monitoring," *Sensors*, vol. 19, no. 21, p. 4690, Oct. 2019.
- [58] K. Setoguchi, S. Higashino, and K. Mitake, "Path planning for small rotary-wing UAVs for disaster site imaging (in Japanese)," *J. Astrophys. Aerosp. Technol.*, vol. 15, pp. 23–32, 2016.
- [59] *Denso XDC02*. Accessed: Dec. 26, 2023. [Online]. Available: <https://www.g-mark.org/en/gallery/winners/9e0d5434-803d-11ed-af7e-0242ac130002?companies=43280d13-c8b8-49d7-b503-0043e8dfc0b9>
- [60] B. Ma, Z. Liu, F. Jiang, W. Zhao, Q. Dang, X. Wang, J. Zhang, and L. Wang, "Reinforcement learning based UAV formation control in GPS-denied environment," *Chin. J. Aeronaut.*, vol. 36, no. 11, pp. 281–296, Nov. 2023.
- [61] B. Ma, Z. Liu, W. Zhao, J. Yuan, H. Long, X. Wang, and Z. Yuan, "Target tracking control of UAV through deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, 2023.
- [62] G. A. Croes, "A method for solving traveling-salesman problems," *Oper. Res.*, vol. 6, no. 6, pp. 791–812, Dec. 1958.
- [63] G. A. Watson, "Computing Helmet transformations," *J. Comput. Appl. Math.*, vol. 197, no. 2, pp. 387–394, Dec. 2006.
- [64] *PROJ Coordinate Transformation Software Library*, Open Source Geospatial Found., Chicago, IL, USA, 2022.



WEITONG WU received the B.E. degree from the School of Automation and Electrical Engineering, University of Science and Technology Beijing (USTB), China, in 2018, and the M.E. degree from the Department of Electrical Engineering, Electronics, and Information, Nagoya University (NU), Japan, in 2021, where he is currently pursuing the Ph.D. degree. His research interests include robot decision planning, path planning, intelligent control systems, modeling, and optimization.



YUKI FUNABORA (Member, IEEE) received the B.E., M.E., and Ph.D. degrees in electrical engineering and computer science from Nagoya University (NU), Japan, in 2007, 2009, and 2012, respectively. In 2012, he was a Postdoctoral Researcher with the RIKEN Advanced Science Institute. Since 2021, he has been an Associate Professor with NU. His research interests include human-cooperative robots, soft robotics, intelligent control, soft computing, and system design.



SHINJI DOKI (Senior Member, IEEE) received the B.E., M.E., and Ph.D. degrees in electronic-mechanical engineering from Nagoya University (NU), Japan, in 1990, 1992, and 1995, respectively. Since 2012, he has been a Professor with NU. His research interests include control, modeling, and signal/information processing and its application for motor drive systems. He was a recipient of the IEEE IECON '92 Best Paper Award and the paper awards from FANUC FA and Robot Foundation and the Institute of Electrical Engineers of Japan.



SATORU YOSHIKAWA received the M.S. degree from the Graduate School of Frontier Sciences, The University of Tokyo, in 2009. In 2009, he joined DENSO Corporation. He has been engaged in research and development of power conversion control for in-vehicle chargers, motion control for robots and UAVs, and image analysis AI.



TETSUJI MITSUDA received the M.S. degree in mechanical engineering from the Nagoya Institute of Technology, in 2004. In 2004, he joined DENSO Corporation. After working on the development of automobile engine control systems, he is currently engaged in the development of digital twin systems utilizing robots and AI.



KAE DOKI (Member, IEEE) received the Ph.D. degree in electrical and electronic engineering from the Graduate School of Engineering, Nagoya University, in 2002. In April 2002, she became an Assistant Researcher with the Department of Electrical Engineering, Faculty of Engineering, Aichi Institute of Technology. After working as a Lecturer, she became an Associate Professor with the Department of Mechanical Engineering, Aichi Institute of Technology, in 2007, and an Associate Professor and a Professor with the Department of Electrical Engineering, Aichi Institute of Technology, in 2012 and 2020, respectively.

Her research interests include robotics, intelligent control systems, and intelligent information processing. She is a member of the Robotics Society of Japan, the Society of Instrument and Control Engineers, and the Institute of Electrical Engineers of Japan.



JINGYU XIANG was born in Shanghai, China, in 1985. He received the B.S., M.S., and Ph.D. degrees in mechanical science and engineering from Nagoya University, Nagoya, Japan, in 2008, 2010, and 2014, respectively. From 2013 to 2014, he was a Visiting Researcher with the Clemson University International Center for Automotive Research. Since 2014, he joined SOKEN Inc. His research interests include autonomous driving, autonomous mobile robots, 3D modeling, and cyber physical system (CPS).

...