2025 IEEE/SICE International Symposium on System Integration (SII)
January 21-24, 2025, Munich, Germany

Preliminary Study on Task Division for UAV-Based Visual Inspection of Large Structures with Multiple Flights Using 3D Urban Models

Masaya Haneda[1], Yuki Funabora[1], Shinji Doki[1]

Abstract— **This paper addresses the problem of task division based on 3D urban models for visual inspection using a UAV with multiple flights. When inspecting large structures for damage detection, UAVs are often flown multiple times to obtain high-resolution data from all areas. Research is progressing on coverage path planning (CPP) methods for data collection in a single flight, and the 3D models of the target required for this can be readily obtained. However, the automation of task division for each flight has not been extensively studied. Because the scale of the target is large, it is necessary to execute a task division method that optimizes the performance efficiency of UAVs within a practical calculation time. This paper presents a task division method for data collection by multiple flights of a UAV based on the decomposition of the 3D urban models. Forming a framework in which data collection tasks are divided based on the 3D mesh decomposition, and then the CPP method is applied to each task, this enables efficient inspection of large-scale structures. In this paper, we implement three basic methods based on the strategies of avoiding going far, equalizing the amount of each task, and reducing unnecessary movement as preliminary research. Each method applies to objects on a scale of several hundred meters, and evaluates their performance in automated data collection.**

I. INTRODUCTION

The deterioration and aging of infrastructures such as bridges are critical problems in Japan and others [1], [2]. It is needed to monitor their conditions and detect damages through visual inspections to prevent serious accidents such as collapses.

To satisfy this requirement, many researchers have proposed coverage path planning(CPP) methods to generate inspection path of structures by UAVs [3]–[6] to automate data collection. The shape of the structures required for these methods can be obtained from urban 3D models as triangular mesh [7], [8].

This paper specifically targets inspections aimed at detecting damages in their early stages. In Japan, crack detection of 0.2 mm in width is required [9]. The scale of damages detectable in image data is proportional to the spatial resolution of the surface [10]–[12]. Some methods focus on this point and perform CPP based on the target geometry, the sensor, and the spatial resolution required for the acquired data [13], [14]. They are excellent methods in terms of guaranteeing damage detection.

Now consider early detection of anomalies in objects with a size of several hundred meters, such as a road junction on a bridge. Since early detection of anomalies such as the above requires high-resolution data collection, the viewpoint set that allows measurement of the entire area is very dense. Increasing the density of viewpoints makes CPP more difficult in two ways. The first is the increase in computation time. The CPP is NP-hard, and the computation takes a lot of time, making it impractical for high-resolution data collection on large targets. The second is that the area that can be inspected in one flight is not large. The path length increases as the viewpoint density increases for the same region on target. The flight time of the UAV is several tens of minutes. When detecting a 0.2 mm wide crack, the size of structure that can be inspected is about tens of meters in length, width, and height [14], making it impossible to inspect the entire area. However, urban 3D models are not divided by inspectable regions in a single flight and cannot be utilized as is.

To solve these two problems, we take the approach of dividing the input triangular mesh into sub-regions to be inspected on each flight. As shown in Fig. 1, the target is divided into sub-regions, and apply CPP method for each mesh. In this paper, we apply three methods experimentally and identify what problems exist as an initial study. The contributions of this paper are as follows.

1) Proposal of a framework for collecting high-resolution data through multiple flights for the inspection, based on the decomposition of 3D urban models and applying CPP to each sub-region.
2) Implementation of three methods based on the strategies of avoiding going far, equalizing each task, and reducing unnecessary movement, that can be considered as intuitive optimization elements.
3) Evaluation based on the perspective of automating data collection for inspections, and identifying areas that need to be improved.

II. RELATED WORKS

The main focus of this paper is on task allocation based on mesh decomposition for large structures. Most studies of CPP focus on making efficient the inspection of targets where the entire area can be measured in a single flight [3]–[6], [13], [14], and not enough research has been done on this issue. One of the characteristics that makes CPP problems difficult is that they require work optimization that takes into account the complexity of the target shape. In single UAV work, this is often addressed by optimizing the path. However, in planning for multiple flights, optimization is also required for the assignment part. This is the area to be examined in this paper. An example of a manual assignment

[1]Department of Information and Communication Engineering, Graduate School of Engineering, Nagoya University Fro-cho Chikusa-ku Nagoya Aichi Japan haneda.masaya@nagoya-u.jp, funabora@nagoya-u.jp

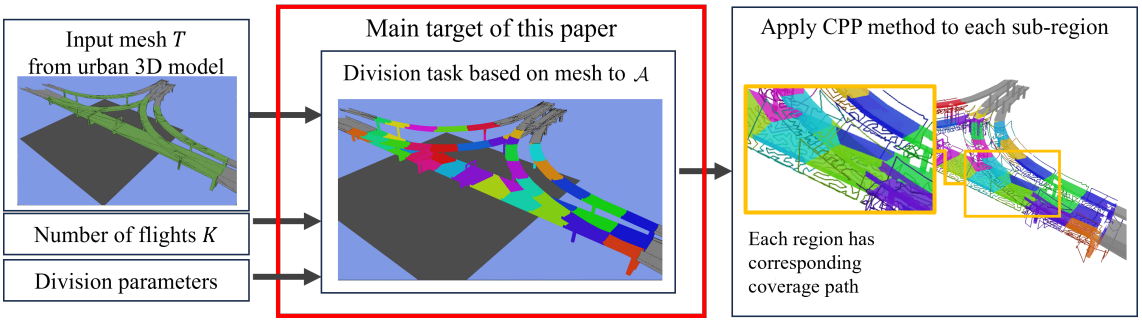


Fig. 1: Algorithm overview.

is the method proposed by Jacob J et al. [15] to divide based on 2D satellite photos. However, it is difficult to manually assign the optimal values for complex shapes. In addition, there are many bridges in various places, for example, there are 730,000 in Japan alone [1], and manually allocate them all is too overwhelming task.

A problem similar in nature is the existence of algorithms for multiple UAV inspection tasks. One simple method of allocating tasks in the CPP is to calculate the optimal path for the entire target and divide the path [16]. However, this is not a practical policy because it is computationally expensive and requires enormous computation time for CPP, especially for high-resolution measurements on large objects. Ivić et. al proposed a multi-UAV trajectory planning method based on ergodicity [17]. This method can be regarded as simultaneously optimizing and allocating routes. However, there are still issues in achieving full coverage, such as a mix of areas where 20 measurements are taken and areas where none are taken. Wei et al. proposed a method [18] to optimize task assignment and route generation by genetic algorithm based on a Coverage Probabilistic Roadmap. This method deals with data collection for objects on the scale of several hundred meters. On the other hand, it only deals with data collection with a spatial resolution that is about 10 times coarser than this paper. The number of viewpoints is about 100 times greater, and there is a possibility that the solution will not converge within a practical time.

In contrast to these methods, the one proposed in this paper is based on the concept of determining assignments at a stage before route planning. If only the assignment of the area of responsibility is required, it may be accomplished with a relatively small amount of computation. Then, by solving the CPP for each subregion, the CPP for the entire area can be achieved. Even with large targets as input, this approach allows for planning in a practical amount of time. After the assignment, we use a CPP method based on spatial resolution [13] in this paper because spatial resolution is important for damage detection, but it is possible to use any method that is appropriate for the problem at hand.

III. PROBLEM STATEMENT

In this section, we first check the 3D mesh characteristics of the joint, which is the target structure of this paper, and describe the problem to be solved.

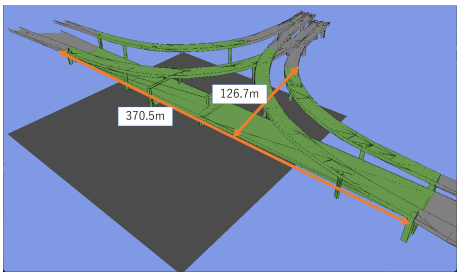


Fig. 2: Inspection target 3D model (*Hakozaki* junction in Japan).

TABLE I: Features of *Hakozaki* junction.

Size	[370.5, 126.7, 30.7] m
Area of surface	34456.2 m^2
Number of truangular	1085

A. Characteristics of inspection target

We deal with CPPs that assume multiple flights based on 3D urban models. In this paper, we use PLATEAU [7], which are 3D urban models of bridges and buildings provided by the Japanese Ministry of Land, Infrastructure, Transport and Tourism.

As a sample of large and complex structures, we use the concrete parts of the *Hakozaki* junction in Japan as a sample, and the mesh of the relevant part was extracted from PLATEAU. The target area is the green region shown in Fig. 2, and quantitative information is shown in Table I. The target area includes five roads. Note that it is difficult to automatically divide the region into each road due to the shared components, such as piers. We apply the task division and CPP to this structure and evaluate.

B. Problem formulation

The problem to be solved in this paper and the method of evaluation are described. The goal of this research is to ensure that the workload for each flight is as even as possible and that the work is carried out efficiently as a whole.

The input data of task division is a set of triangles T that represent the surface shape of the target as a triangular mesh. The entire area of T will be inspected on k times flights. This k is adjustable according to the battery constraints of UAV, etc. The area of the partition result is some sets of triangles, $A = \{A_j\}_{j=1}^{k}$. A_j indicates the area to be inspected during

Algorithm 1 Clustering by distance method algorithm

Input: T, k, r
Output: $A = \{A_j\}_{j=1}^{J}$
1: $T' \leftarrow Subdivide(T, r)$
2: $A \leftarrow VoronoiClustering(T', k)$

a single flight, and j is treated as the task ID. The union of all of A represents the same shape as T.

From the perspective of automating data collection, the evaluation is carried out in two stages. The first stage is based on the results of the task division and checks the variation in the area of each region of A. The second stage is based on the results of applying the CPP method and checks the sum of path lengths corresponding to flight time and the variation of working flight time.

IV. TASK ALLOCATION METHOD

In this section, an explanation is given of how to perform decomposition taking into account the target geometry. Three different methods are implemented, based on the strategies of avoiding going far, equalizing each task, and reducing unnecessary movement. These algorithms are applied in the stage before CPP in the system shown in Fig. 1.

A. Clustering by distance method

The first method is a geometric distance-based clustering of the target, based on the strategy of avoiding going far. To offset the effects of polygon density and equalize workload, a clustering method that weights areas [19] is used.

The process is shown in Algo 1. The input is a set of triangles T representing the surface shape of the target and the number of flights k. First, subdivide each triangle in T by repeatedly bisecting each until they are all smaller than r. The r is a parameter to be adjusted based on the computing capacity and the scale of the target. Next, the target is decomposed by distance-based clustering of T'.

Subdivision allows for any sub-region of the target surface to be represented by a subset of T'. By grouping neighboring triangles, UAVs are avoiding traveling far in a single flight. The main factor of computation time in this allocation algorithm is the clustering part $O(|T'| \cdot K)$, $|T'|$ means the number of subdivided triangles.

B. Directional division method

The second method is to divide a certain area with almost the same area in a certain direction, based on the strategy of equalizing the amount of each task. This method is based on the expectation that by making the area to be inspected the same, the work time for each flight will be divided as evenly as possible.

The process is shown in Algo 2. The input is a set of triangles T representing the geometry of the target, the number of flights k, and vector $\boldsymbol{d}$ indicating the reference for the direction of the split. The total area of target s is used as an intermediate variable. The subdivision process is performed the same as method A. Next, for each triangle

Algorithm 2 Directional division method algorithm

Input: T, k, r, $\boldsymbol{d}$
Output: $A = \{A_j\}_{j=1}^{J}$
1: $T' \leftarrow Subdivide(T, r)$
2: $T'_{\mathrm{sort}} \leftarrow TriangularSort(T', \boldsymbol{d})$
3: $s \leftarrow CalculateArea(T'_{\mathrm{sort}})$
4: $A \leftarrow EqualizeAreaDivision(T'_{\mathrm{sort}}, \frac{s}{k})$

Algorithm 3 Face division method algorithm

Input: T, θ_{th}
Output: $A = \{A_j\}_{j=1}^{J}$
1: $\{\mathbf{p}_i\}_{l=1}^{L} \leftarrow PairingTrianglesSharingVertex(T)$
2: $U \leftarrow InitializeDisjointSetDS(T)$
3: **for** $l = 1$ to L **do**
4: **if** $CalcNormalAngleDiff(\mathbf{p_l}) < \theta_{th}$ **then**
5: $U.union(p_l)$
6: **end if**
7: **end for**
8: $A \leftarrow CreateSetsOfSameRootTriangles(U)$

in T', sort according to the centroid position and vector $\boldsymbol{d}$. After calculating the total area, take out the polygons from T'_{sort} in order, add up the areas, and group them A_j each time they exceed $\frac{s}{k}$.

The above process achieves an equal-area partition based on $\boldsymbol{d}$. The main factor of computation time in this allocation algorithm is the sorting part $O(|T'| \log |T'|)$.

C. Face division method

The third method is to have the UAV take charge of the same face, based on the strategy of reducing unnecessary movement. In many cases, when UAVs change the surface to be inspected, they do not collect data but only move for long periods, which could be reduced. This method is based on the idea that polygons that are adjacent to each other and have a sufficiently small angle difference exist on the same face. By recursively applying this idea, the same face can be obtained. In this paper, we use disjoint-set data structure [20] to implement this process.

The process is shown in Algo 3. The input is a set of triangles T and the threshold parameter θ_{th} of the angle between the normals of adjacent triangles to determine whether they are the same face or not. First, create a list $\{\mathbf{p}_i\}_{i=1}^{j}$ of pairs of adjacent polygons. Next, initialize the disjoint-set data structure U based on T. Each of the included polygons is treated as an independent set. Calculate the angle difference for each pair based on the adjacent pair $\mathbf{p}_i$, and if it is less than or equal to θ_{th}, then union the sets to which the polygons belong. After determining whether or not to union all $\{\mathbf{p}_i\}_{i=1}^{j}$, U can be used as A, result of face division.

If the UAV collects data according to the obtained A, the time for changing the surface to be inspected is 0. It should be noted that in this method, the number of divisions corresponding to the times of flights cannot be directly determined by the user. The main computation time is the

TABLE II: Used parameter for division.

k	30
r	0.45 m
θ_{th}	60°

phase of determining whether vertices are shared between polygons and computing the normal angle, which is $O(|T|^2)$.

V. EVALUATION

In this section, the evaluation is performed using a scenario that assumes the inspection of the *Hakozaki* Junction in Japan(3D mesh data is obtained from PLATEAU [7]). After the assignment, generate paths by the CPP method considering spatial resolution [14]. Evaluation will be based on the evenness of the area corresponding to the task assignment, the variation of each path length time, and the total flight time. These evaluations were conducted on a desktop computer powered by an Intel(R) Core(TM) i9-10900K CPU.

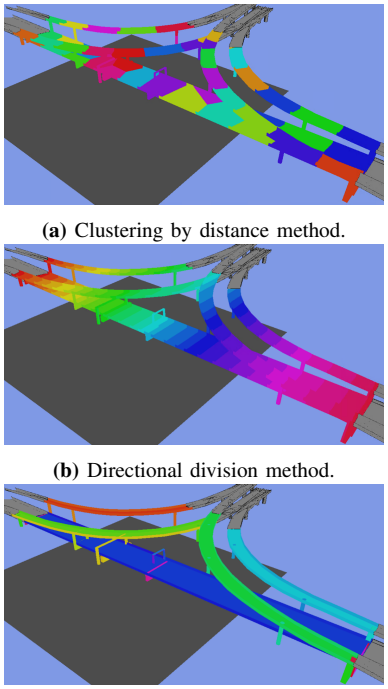
A. Evaluation of division results

First, check the area of each sub-region after task division. The parameters in Table II were used for task division. The discussion in this paper will be based on $k = 30$ times flights. For $\boldsymbol{d}$ in Method B, it was determined to be a vector along the longitudinal direction of the thick road.

The allocation result is shown in Fig. 3. Color-coded for each sub-region obtained by task division. Method A groups spatially close areas together, and in method B in the directional division, the object is divided like a ribbon. Method C based on face division is broken down into each side of the road, including the underside and sides. Each method took 40.5 seconds, 3.0 seconds, and 0.1 seconds to process, respectively. Method C is the quickest, but this is because the subdivision of the polygon is not necessary and the input polygon was relatively large. In the case of input data is smaller triangles, the calculation time O($|T|^2$) is expected to be longer than for methods A and B.

The area of each sub-region obtained by division is shown in Fig. 4. First, a comparison of Methods A and B is made. While Method B has all equal areas, the area variation is confirmed in Method A. As an example, the 27th task corresponds to the task of measuring only a part of the piers, and its area is extremely small. Method A has a problem in that it is difficult to control the area of each task. Although the weighting of the area is intended to make the area uniform, the distance-based cost is thought to be dominant, resulting in this outcome.

For method C, the number of divisions is separated into 71 planes. The main reason is that there are several surfaces corresponding to relatively small members such as Pier and joint components. Because of their distance from each other, it is difficult to deal with such a situation as putting together a sub-region with a small area. For the largest result 46th task, which corresponds to the underside of the bridge, the entire area cannot be inspected in a single flight, and further division is necessary. For these reasons, it is not appropriate to use the method of dividing the inspection task into separate tasks for each surface. From the above, we can assume that methods A and B are practical in terms of task division.



(a) Clustering by distance method.

(b) Directional division method.

(c) Face division method.

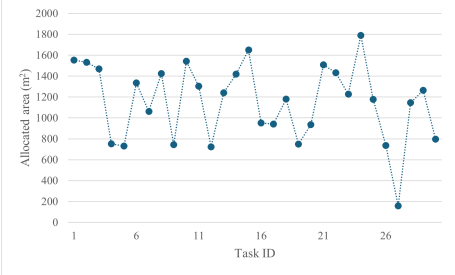
Fig. 3: Visualized segmentation results.

B. Evaluation of CPP results

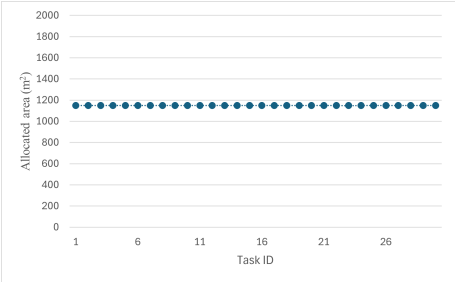
Next, apply the CPP method [13] to the results of Methods A and B for each sub-region. Based on the crack width to be detected being 0.2 mm [9], and the required spatial resolution on crack detection in image [21], the data 's spatial resolution parameter is set to 1 mm/pixel. The parameters of the camera intended for use are given in Table III. A coverage path is obtained by generating viewpoints based on various parameters and performing path length-based optimization using the SA method [22].

Planned paths are shown in Fig. 5, and to evaluate the overall work, statistics for each flight about methods A, and B are shown in Table IV. Note that in this viewpoint generation, the viewpoint is generated so that the object is inspected orthogonally to the surface, so there is a possibility that joints and intricate areas cannot be measured. The method of path generation based on the spatial resolution of the currently acquired data is based on the policy of preventing data degradation by assuming orthogonal measurements. This is an issue that needs to be resolved in the future, but the difference in the inspection area is less than 0.1% of the total, so it does not hurt the evaluation. In addition, since this is not the main point of this paper, we will proceed with the evaluation as is.

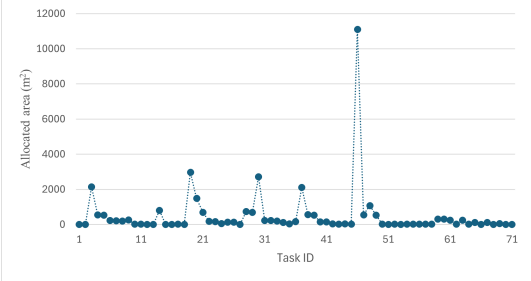
For evaluation, estimate the flight time of the UAV. One of the UAV inspections involves working [23] at a travel speed of 0.8 m/s. In this case, based on the difference in path length of 2125m, method A is estimated to complete the work in



(a) Clustering by distance method.



(b) Directional division method.

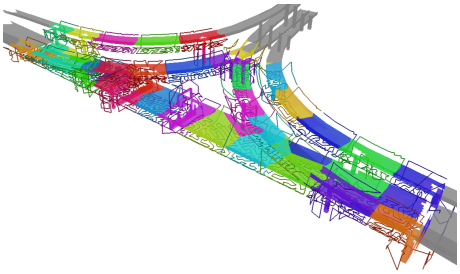


(c) Face division method. (Note that the scales are different.)

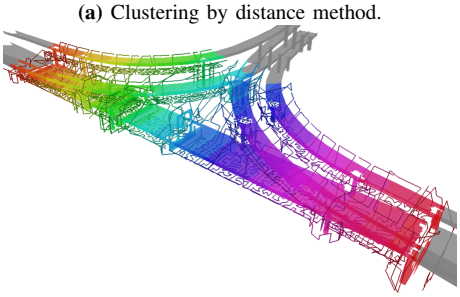
Fig. 4: Plot of the area of each sub-region after division

TABLE III: Camera parameter for input of CPP.

focal length	25 mm
image resolution	5000 x 5000 pixel
image sensor size	15.6 x 15.6 mm



(a) Clustering by distance method.



(b) Directional division method.

Fig. 5: Visualized CPP result.(Not inspected area are deleted.)

TABLE IV: Statistics summary of each CPP result.

	method A	method B
Number of viewpoint	4380	4543
Sum of path length (m)	14087	16212
Maximum of path lengths (m)	690.9	762.3
Standard deviation of each path length(m)	135.2	79.5
Sum of inspected Area (m^2)	29289	29267
Standard deviation of each inspected area(m^2)	279.1923	156.0355
Correlation between each path length and corresponding area	0.82	-0.21

44.3 minutes less flight time. In addition, UAV batteries can generally be used continuously for around 20 minutes, and based on the longest path in both methods which is smaller than 960 m which is the limit of the path length that can be assumed based on the battery, can be completed in a single flight. From these results, we can say that the method is capable of assignments that take into account the complexity of the target. The correlation coefficients between path length and inspection area for each flight are shown in Table IV. From the table, there is a strong correlation for method A, while there is no correlation for method B. From these points of view, it can be said that the same UAV can be assigned to geometrically close areas of a structure to reduce the amount of time spent on worthless travel. On the other hand, the standard deviation of the length of the path and the inspection area for each flight is smaller for method B. This shows that equalizing the size of the area in charge contributes to equalizing the tasks.

The computation time is also checked. Check the computation time, especially for the viewpoint generation section, which is the main cause of computation time in this case. 7.65 hours for Method A and 2.75 hours for Method B. The computational order of the viewpoint generation part of the CPP method [13] used in this study is shown as $O(AREA(T)*N_v)$, using the number of N_v required for the whole area measurement. Here, the N_v part can be converted by the request spatial resolution s_{req} and $AREA(T)$, and can be expressed as $O(AREA(T) \cdot \frac{AREA(T)}{s_{\mathrm{req}}})$. In these experiments, we have split the problem, and the computation order can be shown as $O(k \cdot \frac{AREA(T)}{k} \cdot \frac{AREA(T)}{ks_{\mathrm{req}}})$ with the number of flights k. From the comparison of the computation volume order, it can be said that the computation time could simply be reduced to $1/30$. It can be said that the task division process, which takes less than a few minutes, has achieved a significant reduction in calculation time.

From the above results, it can be said that the division proposed in this paper shows that planning for work based on multiple flights is feasible in practical computation time. It also suggested that, although there is still room for improvement, an automatic decomposition can achieve the optimization of the equalization of tasks and the efficiency of overall work.

VI. CONCLUSIONS

We investigate a task division method based on the mesh decomposition of the 3D urban models. By combining this with the CPP method, it can be used in a planning framework for inspecting large-scale structures using multiple flights. In this paper, the three basic methods were implemented and evaluated. Method A is to group geometrically close regions, based on the strategies of avoiding going far. Method B is to divide the object into equal areas along a certain direction, based on the strategies of equalizing each task. Method C is to divide the object based on faces, based on the strategies of reducing unnecessary movement. After dividing a large complex structure using these methods, the CPP method was applied and evaluated. The first method reduced unnecessary travel, while the second method achieved an even division of labor. The third method was found to be unsuitable in terms of flight sharing, but we anticipate that there may be situations where it can be used for planning algorithms to make UAV operations efficiently. These methods are the first steps towards optimization, equalizing the work of each UAV and increasing the efficiency of the overall operation.

Looking ahead, there is an issue for improvement of task division based on decomposition. Each method is implemented based on a single strategy, but it will be necessary to implement a method that can optimize for multiple purposes, such as equalization and reducing work time. It may be possible to solve the problem of reducing the number of operations that occur when changing the surface of the measurement target in the method C strategy by changing the algorithm structure or cost function. In addition, there is room for improvement in this method concerning equalization of flight time for each flight. Furthermore, by considering collision avoidance between UAVs, it may be possible to introduce this method to simultaneous operations of multiple UAVs.

As a by-product, it was also confirmed that the constraint of facing the target surface squarely to avoid data degradation, about the task of collecting high-resolution data, interferes with full coverage. To date, there has been insufficient research into methods of planning that include angled photography, on the premise of guaranteeing the spatial resolution of the collected data. We recognize that this problem should also be solved on the premise of collecting data that has sufficient spatial resolution to guarantee anomaly detection.

REFERENCES

[1] T. Ministry of Land, Infrastructure and Tourism, *WHITE PAPER ON LAND, INFRASTRUCTURE, TRANSPORT AND TOURISM IN JAPAN, 2023*, 2023.

[2] "Structurally deficient bridges — 2021 infrastructure report," https://infrastructurereportcard.org/cat-item/bridges-infrastructure/ (accessed Jul. 11, 2024).

[3] C. Feng, H. Li, M. Zhang, X. Chen, B. Zhou, and S. Shen, "Fc-planner: A skeleton-guided planning framework for fast aerial coverage of complex 3d scenes," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 8686–8692.

[4] C. Cao, J. Zhang, M. Travers, and H. Choset, "Hierarchical coverage path planning in complex 3d environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3206–3212.

[5] H. W. Tong, B. Li, H. Huang, and C. Wen, "Uav path planning for complete structural inspection using mixed viewpoint generation," in *2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2022, pp. 727–732.

[6] N. Bolourian and A. Hammad, "Lidar-equipped uav path planning considering potential locations of defects for bridge inspection," *Automation in Construction*, vol. 117, p. 103250, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580519309355

[7] T. Ministry of Land, Infrastructure and Tourism, "Plateau [plateau] — a project led by the ministry of land, infrastructure, transport and tourism to develop 3d city models of all of japan and make them available as open data(in japanese)," https://www.mlit.go.jp/plateau/ (accessed Aug. 29, 2024).

[8] OpenStreetMap contributors, "OpenStreetMap," https://www.openstreetmap.org/((accessed Aug. 31, 2024)).

[9] T. Ministry of Land, Infrastructure and Tourism, *The gist of Scheduled Inspections Targeting Bridges(in Japanese)*, 2019.

[10] N. Gehri, J. Mata-Falcón, and W. Kaufmann, "Automated crack detection and measurement based on digital image correlation," *Construction and Building Materials*, vol. 256, p. 119383, 2020.

[11] H. S. Munawar, A. W. 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, 2021.

[12] R. G. Lins and S. N. Givigi, "Automatic crack detection and measurement based on image analysis," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 3, pp. 583–590, 2016.

[13] K. Asa, Y. Funabora, S. Doki, and K. Doki, "Evaluation in real world of the measuring position determination for visual inspection using uav," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, 2018, pp. 2711–2716.

[14] K. Asa, "Path planning of uav for measurement task in visual inspection of infrastructures," Ph.D. dissertation, Nagoya University, 2020.

[15] J. J. Lin, A. Ibrahim, S. Sarwade, M. Golparvar-Fard, Y. Nitta, H. Moirkawa, and Y. Fukuchi, "Bridge inspection with aerial robots and computer vision: A japanese national initiative," in *Proceedings of the 37th International Symposium on Automation and Robotics in Construction (ISARC)*. Kitakyushu, Japan: International Association for Automation and Robotics in Construction (IAARC), October 2020, pp. 1598–1605.

[16] D. Kurabayashi, J. Ota, T. Arai, and E. Yoshida, "Cooperative sweeping by multiple mobile robots," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, 1996, pp. 1744–1749 vol.2.

[17] S. Ivić, B. Crnković, L. Grbčić, and L. Matleković, "Multi-uav trajectory planning for 3d visual inspection of complex structures," *Automation in Construction*, vol. 147, p. 104709, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580522005799

[18] W. Jing, D. Deng, Y. Wu, and K. Shimada, "Multi-uav coverage path planning for the inspection of large and complex structures," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 1480–1486.

[19] M. Haneda, Y. Funabora, S. Doki, and K. Doki, "Shooting position generation with area normalized polygon clustering for drone-based visual inspection of structure (in japanese)," in *Proceedings of 2022 Tokai-Section Joint Conference on Electrical, Electronics, Information, and Related Engineering*. Tokai-Section Joint Conference on Electrical, Electronics, Information, and Related Engineering, 2022, pp. F4–8.

[20] R. Tarjan, "Efficiency of a good but not linear set union algorithm," *Journal of the ACM (JACM)*, vol. 22, no. 2, pp. 215–225, Apr. 1975.

[21] K. Keisuke and M. Hiroshi, "Verification of digital image resolution and visible crack width of concrete (in japanese)," *Journal of Civil Engineering Structures and Materials(in Japanese)*, vol. No.35, pp. 115–122, 2019. [Online]. Available: https://cir.nii.ac.jp/crid/1010286980753935880

[22] C. C. Skiścim and B. L. Golden, "Optimization by simulated annealing: A preliminary computational study for the tsp," in *Proceedings of the 15th Conference on Winter Simulation - Volume 2*, ser. WSC '83. IEEE Press, 1983, p. 523–535.

[23] "Bridge image measurement technology(in japanese)," https://www.mlit.go.jp/road/sisaku/inspection-support/pdf/c/BR010012.pdf (accessed Aug. 29, 2024).