



OPEN

Model-based analysis of multi-UAV path planning for surveying postdisaster building damage

Ryosuke Nagasawa¹, Erick Mas²✉, Luis Moya^{2,3} & Shunichi Koshimura²

Emergency responders require accurate and comprehensive data to make informed decisions. Moreover, the data should be acquired and analyzed swiftly to ensure an efficient response. One of the tasks at hand post-disaster is damage assessment within the impacted areas. In particular, building damage should be assessed to account for possible casualties, and displaced populations, to estimate long-term shelter capacities, and to assess the damage to services that depend on essential infrastructure (e.g. hospitals, schools, etc.). Remote sensing techniques, including satellite imagery, can be used to gathering such information so that the overall damage can be assessed. However, specific points of interest among the damaged buildings need higher resolution images and detailed information to assess the damage situation. These areas can be further assessed through unmanned aerial vehicles and 3D model reconstruction. This paper presents a multi-UAV coverage path planning method for the 3D reconstruction of postdisaster damaged buildings. The methodology has been implemented in NetLogo3D, a multi-agent model environment, and tested in a virtual built environment in Unity3D. The proposed method generates camera location points surrounding targeted damaged buildings. These camera location points are filtered to avoid collision and then sorted using the K-means or the Fuzzy C-means methods. After clustering camera location points and allocating these to each UAV unit, a route optimization process is conducted as a multiple traveling salesman problem. Final corrections are made to paths to avoid obstacles and give a resulting path for each UAV that balances the flight distance and time. The paper presents the details of the model and methodologies, and an examination of the texture resolution obtained from the proposed method and the conventional overhead flight with the nadir-looking method used in 3D mappings. The algorithm outperforms the conventional method in terms of the quality of the generated 3D model.

When a disaster such as an earthquake or tsunami strikes, obtaining a comprehensive picture of the damage situation is one of the most important tasks in disaster response. For instance, to assess the extensive damage produced by the Great East Japan Earthquake of March 11, 2011, responders relied on aerial photographs and satellite remote sensing to avoid delays in information collection caused by the destabilization and limitations of ground transportation¹. Especially when the accident at the Fukushima Daiichi Nuclear Power Plant occurred, satellite remote sensing and unmanned aerial vehicles (UAVs) were crucial platforms to avoid human exposure to radiation on the premises of the power plant². Significant progress has been made in the estimation of building damage in disasters using satellite remote sensing^{3,4}. For instance, Moya et al. (2020)⁵ identified damaged areas based on change detection algorithms, while Moya et al. (2019)⁶ used three dimensional texture features and support vector machine (SVM) to detect collapsed buildings with high accuracy. The proposed method were tested using synthetic aperture radar data, and Lidar-based digital surface model (DSM). Thus, satellite remote sensing provides an initial estimation for constructing damage maps in wide areas; however, the details of the damage to buildings typically remain unclear because of the lack of information about the condition of the buildings' walls and the satellite image resolution. For this reason, increasing attention is being paid to alternative airborne remote sensing systems, such as UAVs. Photos taken by UAVs can be used to construct a 3D model of a particular point of interest (POI) using the structure-from-motion (SfM) and multiview stereo (MVS) methods⁷⁻⁹.

Indeed, UAVs are currently being applied in several fields, such as aerial assessment of social infrastructure, aerial photography, aerial surveying, and pesticide spraying. In addition, UAVs are used for disaster site surveys because of their low cost and low measurement time requirements. Meanwhile, SfM technology has progressed

¹Mox-Motion, Tokyo 161-0033, Japan. ²International Research Institute of Disaster Science, Tohoku University, Sendai 980-8572, Japan. ³Japan-Peru Center for Earthquake Engineering and Disaster Mitigation (CISMID), National University of Engineering, Lima, Peru. ✉email: mas@irides.tohoku.ac.jp

by virtue of advances in computer vision. Currently, SfM techniques are able to extract many characteristic points from overlapping photographic images and simultaneously estimate the three-dimensional (3D) coordinates of elements and characteristic points of a camera image. Similarly, MVS generates high-precision 3D point groups or mesh models via a matching function among multiple photographic images based on the 3D geometric information collected through SfM. SfM and MVS are conducted simultaneously in most commercial software programs. Software tools such as Pix4D¹⁰ and PhotoScan¹¹, among others, have become popular among UAV users. In fact, private companies and individuals can now conduct SfM surveys quite easily. However, in most commercial software packages, the mission planning feature is limited to a few standard configurations, such as grid missions with a nadir-looking camera (overhead flight) or a fly-around of a point of interest. The limited flexibility of these path planning algorithms means that several flights must be made to capture multiple targets in a complex environment¹². This issue is critical in the case of disaster response, in which biases and a lack of information in surveys are of particular concern. Moreover, planning multiple flight missions and paths is a time-consuming process that can delay disaster relief activities. A precise 3D model of a complicated structure can be produced with the aid of commercial softwares. The method applied in these software packages combines the images acquired by the UAV with the images acquired on the ground¹⁰. A large body of work on 3D scanning, path planning and 3D reconstruction is available in the literature, here we briefly present the most salient work in 3D reconstruction and Coverage Path Planning (CPP) algorithms. Then, we developed a pipeline for multi-objective and multi-UAV path planning. Finally, the method is compared to the conventional stripmap nadir-looking flight pattern (hereafter, *the conventional approach*) used in 3D mapping of urban areas¹³.

In summary, we contribute the following: (i) a simple pipeline that allows users to allocate multiple targets and paths for a similar number of UAVs to reconstruct possible damaged buildings in 3D after a disaster; (ii) a general area division method based on the clustering and balanced task approach; (iii) a pipeline that ensures collision avoidance among UAV units and the environment; and (iv) a comparison of the 3D reconstruction quality of synthetic scenes of the proposed pipeline and the conventional approach.

Motivation of the study. The authors' research studies have been focused on exploring methodologies and alternatives to grasp the damage situation of buildings and essential infrastructure in the aftermath of a disaster. Satellite remote sensing technologies are used as the main tools to acquire the damage information within the impacted areas. We previously developed several methods to enhance building damage mapping by using pre- and post-disaster satellite imagery and change detection algorithms⁵, machine learning¹⁴ and deep learning methods as well¹⁵.

There is a great advantage in terms of the level of coverage of satellite images; however, there are limitations regarding the level of detail of the damage that can be grasped from a top-down view. In particular, there is concern regarding the condition of walls, which may have collapsed or have sustained structural damage and may compromise the stability of the building. Thus, to acquire such detailed information a closer look at the POI is necessary. In addition, to avoid the deployment of human teams that might incorporate new risks and demand additional costs and time, UAVs are a feasible alternative for this task. Thus, one of the motivations for this study is to tackle the limitations posed by satellite remote sensing damage estimation by integrating the advantages of UAV reconnaissance capabilities within the damage mapping pipeline.

To this end, the need to speed up the acquisition of images from various damaged structures require the implementation of multiple UAVs into the surveying task. In addition, it is important to consider a suitable workflow for CPP that ensures safe flight and high quality 3D reconstruction of the POIs.

State of the art

3D reconstruction using UAVs. Nex et al.¹³ showcases various 3D reconstruction models from UAV platforms applied to various fields of study. They show a typical pipeline and the approximate time taken to produce a typical UAV-based photogrammetric workflow. It is suggested that "flight planning" takes only 5% of the whole workflow time, while "image acquisition" (20%) and "Digital Surface Model generation" (25%) are the tasks requiring the highest effort time. Thus, it may seem that the task of interest in this study is not significant; however, the estimations above refer to small areas and are based on the flight-planning of one UAV unit following a conventional approach. On the other hand, for wide areas, when increasing the number of UAV units, the complexity of flight-planning increases, thus the necessary time for flight planning might become significant in the workflow¹⁶.

In addition, Zheng et al.^{17, 18} developed a multi-UAV route planning methodology for 3D building model reconstruction. They focused on the use of multi-UAV for the reconstruction of a single building and used an initial model based on information extruded from the building footprint data of the POI. The created volumetric is then used to estimate the drone location to obtain sufficient coverage of the building. The idea is extendable to multiple POIs, and in the case of postdisaster environments, based on the damage mapping conducted from satellite remote sensing, as presented in the previous section, the footprint information can be updated to produce a more reliable initial map. Notice that at this point we are clearly ignoring other features and infrastructure in the urban environment that might pose a collision risk for drones (e.g. poles, electric cables, antennas, ad billboards, etc.). In addition, Hepp et al.¹² proposed trajectory optimization to obtain high-quality 3D reconstruction of large buildings. First, a low-precision model of the area is generated using an overhead flight pattern. Then, they suggest an optimization of viewpoints to obtain the final path that will generate a suitable camera position constellation for sufficient coverage of the POI. A similar approach uses semantic segmentation from an initial flight to optimize the route and ensure a safe flight plan¹⁹.

Coverage path planning. The problem of planning multiple tasks for multiple UAVs can be addressed as a 'divide-and-conquer' approach²⁰, in which the tasks are divided or allocated to each UAV unit and the subproblems are solved individually. In this case, the task is similar to a multiple traveling salesman problem (MTSP). Thus, first, a clustering of the visiting points (drone camera locations) is necessary. Jain et al.²¹ describes data clustering best practices, such as K-means. Previously, Barrientos et al.²² used a path planning algorithm to survey farms. They transformed the farm into a low-resolution, two-dimensional grid and then divided the area equally between the drone units. Next, a search algorithm with backtracking was used to find the paths that minimize the number of turns. Similarly, Bouzid et al.²³ applied rapidly exploring random tree star fixed nodes (RRT*-FN) combined with the genetic algorithm (GA) by assuming that the CPP issue emulates a vehicle routing problem (VRP). In addition, Gao et al.²⁴ used spanning tree coverage (STC) with ant colony optimization (ACO). On the other hand, Ju et al.²⁵ applied a distributed swarm control algorithm to show that the performance of the multi-UAV system is significantly superior to the single-UAV system. They considered the following seven performance metrics: total time, setup time, flight time, battery consumption, inaccuracy of land, haptic control effort, and coverage ratio. Similarly, Ge et al.²⁶ studied path planning for oilfield inspection within a 3D environment. Here, the cost function includes metrics such as: distance, height, time and electricity consumption. They proposed a combined use of metaphor-inspired metaheuristic algorithms like pigeon-inspired optimization (PIO) and fruit fly optimization algorithm (FOA). FOA is applied to account for dynamic obstacles and to increase computation speed. In recent years, Cabreira et al.²⁷ presented a survey of studies focusing on CPP approaches in UAVs. Unfortunately, the CPP problem presented in their review is focused on 2D mapping representations where cellular decomposition is common. Such methods are conventionally used for area mapping; however, their resulting quality is limited by the flight pattern, which is basically a back-and-forth flight along parallel lines while keeping the camera position on a nadir looking angle.

Various CPP methods have been tested when clustering is applied. For instance, Zhao et al.²⁸ studied task clustering and planning using K-means and simulated annealing (SA) to ensure balance in the task assignment. Later, Botteghi et al.²⁹ confirmed that K-means performs well when the priority is to balance the distance and time among UAVs. Similarly, Mansouri et al.³⁰ applied K-means as the clustering method for a collaborative CPP focused on single complex 3D infrastructure. Furthermore, Kong et al.³¹ proposed a fusion of the K-means and Dijkstra algorithms to allocate offline stores for sales staff and optimize the visiting path. This is a traveling salesman problem (TSP) solved by using clustering and shortest path calculations. The algorithm can evenly distribute the workload to different sales staff. In our case, if a sales staff member is understood to be a UAV unit and an offline store is a camera location, then the algorithm can be transferred to our problem. However since points are in space, special care is needed when applying shortest path because trajectories may cross physical objects.

In addition, Bouras et al.³² developed a two-scale algorithm to plan a fleet of UAVs visiting various points of interest to scan an area for 2D ortho mapping. Similar to Bouzid et al.²³, they set the CPP task assuming a VRP solved with SA. Here, UAV units consider a hovering time to take measurements and the flight is set to go to the POI and back to the station. Since their purpose is to map the area of interest without paying attention to specific POIs, these are not given and are calculated from an iterative process of voronoi tessellation and centroid selections. On the other hand, similar to the case we will present here, Xiao et al.³³ focused on the CPP problem considering that the 3D reconstruction will be generated by an available software program. They focused on ensuring image overlap and reducing energy consumption. In contrast, they assume flight with a nadir angle and splitting the task of one model building among the available UAV units.

Furthermore, among recent studies, Ali et al.³⁴ proposed a self-organization, collective motion and control of a UAV swarm. They used Particle Swarm Optimization (PSO) and Multi-agent System (MAS) to design an algorithm towards self-synchronization among UAVs. Similar to this study, they handle the problem in a 3D space and the verification of their method was conducted on a simulation-based approach. PSO, is one of the most used optimization algorithms in swarm research, in this regard, Shao et al.³⁵ improved the PSO algorithm by introducing a chaos-based particle initialization method and adaptive parameter adjustments to improve optimality and rapidity of the algorithm. The improvement proved to outperform other similar PSO methods. In contrast to our study, Shao et al.³⁵ focuses on simple trajectory with start-to-end points without intermediate goals, however similar to our case, they also used a simulated-based verification of their method.

In summary, clustering methods, such as K-means, have been applied to the CPP of multiple UAVs and solved for sub-trajectories as TSP or VRP problems using optimization algorithms. Unfortunately, most scholars applied this to a single structure with multiple UAVs. In contrast, we will explore the application of multiple structures with multiple-UAVs since a damage assessment requires the inspection of multiple targets.

UAV for disaster mapping. Disaster situation awareness can be significantly improved by using UAV imagery. UAVs can be small, light-weight and easily deployed quickly. Thus, search and rescue, damage mapping and recovery monitoring are some of the tasks suitable for UAVs to perform. Moreover, UAV imagery has higher resolution than satellite and airborne photos; these are a significant data-set to map building damage⁷. For instance, Nex et al.⁷ focused on real-time damage mapping using UAVs. However, the flight plan is proposed for one UAV unit and with the purpose of exploring an unknown area. The photos obtained are used to detect damage through machine learning algorithms. In contrast, in this paper, we focus on a second stage of damage mapping, where essential infrastructure require greater details for assessment and a 3D reconstruction of each building is necessary. In this case a strip-map with the nadir-looking angle is not sufficient to obtain a detailed 3D reconstruction. Previous research has focused on 3D reconstruction of damaged buildings by using commercial off-the-shelf UAV units³⁶.

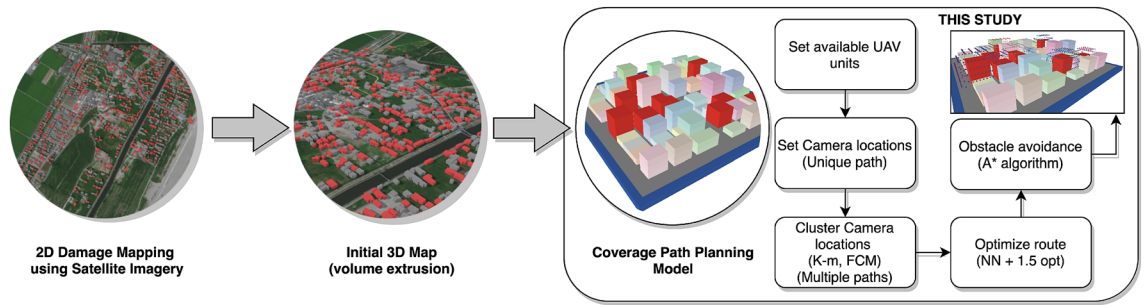


Figure 1. Workflow and scope of this study. Satellite remote sensing generates a 2D damage map. Volume extrusion from building geometries provides an initial 3D map, and the current model calculates the paths for each available UAV unit. *Source:* This study. The maps and virtual environment were developed in ArcGIS Pro and NetLogo3D, respectively.

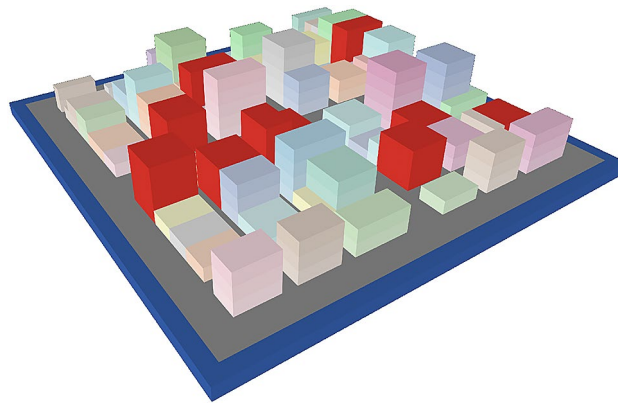


Figure 2. Virtual urban environment created in NetLogo3D. Here, the red volumes represent damaged buildings in need of inspection. *Source:* This study.

Methodology

We summarized the workflow in Fig. 1. Based on the information obtained from building damage mapping using satellite remote sensing an initial 3D map is constructed based on the extrusion of building geometries, as suggested in Botteghi et al.²⁹. Another alternative would be to obtain the initial map through a low resolution and real-time reconstruction to, then, proceed to the CPP model. The CPP model will be the focus of this paper. To inspect all damaged buildings in the area, the necessary camera locations are generated as if a unique path is required (“[Generation of camera location points](#)” section). Next, based on the number of available UAVs (user setting), a cluster algorithm (i.e., *K*-means or Fuzzy *C*-means) is executed to calculate one path per UAV (“[Clustering camera location points](#)” section). The clustered points are joined to generate a unique path per UAV using the nearest neighbor (NN) and are improved with the 1.5-opt approximation algorithm (“[Route optimization](#)” section). Final routes are verified for collision avoidance and are corrected using the A* algorithm to overcome obstacles (“[Obstacle avoidance](#)” section).

For the purpose of evaluating the method, we built a virtual urban environment using NetLogo3D³⁷ (Fig. 2). A demo video can be found in the supplementary data. The configuration is set randomly and the structures to be surveyed are visualized in red. Here, the camera locations and flight paths are calculated.

Generation of camera location points. First, camera locations separated every 2.5 m horizontally and 5 m vertically were generated around the objects of interest. The camera was presumed to face the center of the nearest object. In addition, locations satisfying any of the following conditions were deleted from the candidate list: (i) the distance to the object of interest is less than a certain user-defined value (7.5 m in this case); (ii) a structure exists between the object and the camera; and (iii) the coordinates of the candidate are inside of a structure. We selected a camera visibility angle of 90° horizontally; additionally, the camera aspect ratio was set to 4:3, and the distance between the camera and the object was set to 7.5 m. Moreover, the images taken at each location were required to have a forward overlap ratio of at least 80% and a side overlap ratio of over 50% to ensure sufficient multiplicity.

Clustering camera location points. Since the generated candidates for camera locations are shared among all UAVs, a clustering method for allocating specific locations to specific UAV units is necessary. Clus-

tering is among the most studied problems in machine learning. Methods like SVM or Deep Learning models require training data, however the K -means method does not rely on training data.

K -means. The K -means method is an algorithm for non-hierarchical clustering. Clustering is the process of grouping a series of data. As a type of unsupervised learning method, non-hierarchical clustering is a method for directly classifying similar data into groups using an evaluation function. The K -means method follows a simple algorithm; therefore, it is a typical clustering method that is widely used²¹. Consider a set of camera locations x_i ($i = 1, \dots, n$) to be partitioned in k sets S_j ($j = 1, \dots, k$), where each set is to be allocated to a unique UAV unit. First, a random partition is used to initialize the algorithm by allocating the data into one cluster, thus, creating k clusters. Then, we find the centroids m_j ($j = 1, \dots, k$) of each cluster and update it as the new mean. The distance between each observation x_i and each centroid m_j is calculated to re-sort the data into their closest clusters.

The assignment and update of means can be represented as Eqs. (1) and (2), where t denotes the step which continues until convergence.

$$S_j^{(t)} = \{x_i : \|x_i - m_i^{(t)}\|^2 \leq \|x_i - m_j^{(t)}\|^2, \forall i, 1 \leq i \leq n, \forall j, 1 \leq j \leq k\} \quad (1)$$

$$m_j^{(t+1)} = \frac{1}{|S_j^{(t)}|} \sum_{i=1}^n x_i \quad (2)$$

Next, if the change in m_j is less than a predetermined threshold value, the operation is terminated, otherwise a new assignment of clusters is conducted until convergence occurs.

In mathematical terms, the process described above is applied to minimize the following objective function:

$$\operatorname{argmin}_S \sum_{j=1}^k \sum_{i=1}^n \|x_i - m_j\|^2 \quad (3)$$

The performance of this method is known to strongly depend on the positions of the initial clusters. Therefore, the first application of this method might not yield the best sorting results. In addition, it is assumed that the clusters are spherical and that the number of clusters into which the data should be classified is known, thus classification is difficult if these assumptions are violated.

Fuzzy C -means. When each observation x_i is sorted into just one cluster, such clustering is referred to as hard clustering. By contrast, when data are partially sorted into multiple clusters, such ambiguous clustering is known as a fuzzy clustering. The K -means method belongs to the former type of clustering, while the fuzzy C -means (FCM) method belongs to the latter. The FCM method provides each x_i with an attribution degree w_i ($0 \leq w \leq 1$) with respect to each cluster based on the distance to the centroid c_j .

The algorithm starts at $t = 0$ with a random assignment of attribution degree coefficients. While there is no convergence (the change in coefficients' values between two iterations is lower than a threshold), the algorithm loops on computing the centroid of each cluster $c_j^{(t)}$ ($1 \leq j \leq c$). In the FCM method the centroid is the weighted mean of the camera location points with the degree of attribution to the cluster as the weight [Eq. (4)]. In addition, a hyper-parameter m is defined to control the *fuzziness* of clusters. Next, the partition matrix or attribution degree matrix w_{ij} is calculated from Eq. (5).

$$c_j^{(t)} = \frac{\sum_{i=1}^n (w_{ji}^{(t-1)})^m x_i}{\sum_{i=1}^n (w_{ji}^{(t-1)})^m} \quad (4)$$

$$w_{ij}^{(t)} = \frac{1}{\sum_{k=1}^c \left(\frac{\|x_i - c_j^{(t-1)}\|}{\|x_i - c_k^{(t-1)}\|} \right)^{\frac{2}{m-1}}} \quad (5)$$

The value of m is usually set to a real number between 1.3 and 2. In the limit $m = 1$, this method reduces to the K -means method of hard clustering. In mathematical terms, this method aims to minimize the following objective function:

$$\operatorname{argmin}_C \sum_{i=1}^n \sum_{j=1}^c (w_{ij})^m \|x_i - c_j\|^2 \quad (6)$$

Route optimization. With camera locations clustered and sorted to each UAV unit, the next step is to create optimum paths for each UAV within its clustered points. Then, the multiple traveling salesman problem (MTSP) is solved in this study. Here, each UAV visits all camera location points in its cluster only once and then returns to the starting point. For all UAVs, the starting point and the end point after the flight were assumed to

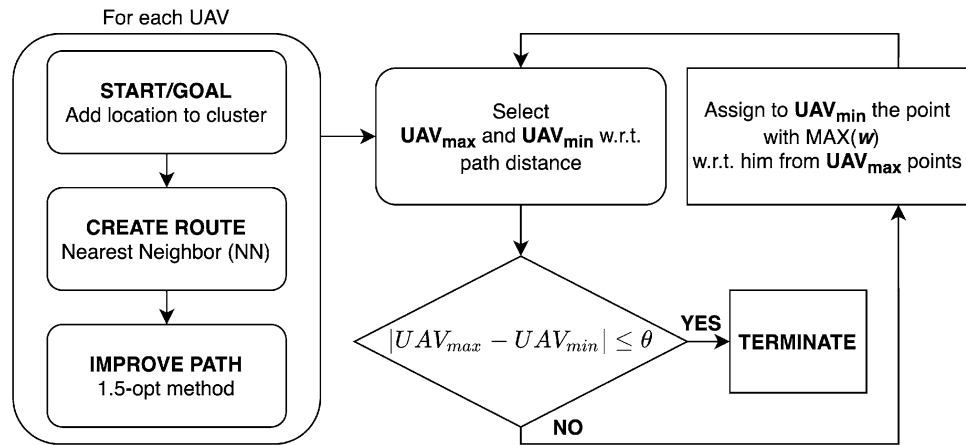


Figure 3. Process to calculate optimum paths for UAVs. Source: This study.

be the same based on practical considerations. The MTSP for the UAV routes was solved to select one solution for which the cost was minimal and uniform across all devices.

The UAV routes are created using the nearest neighbor (NN) method and are improved with the 1.5-opt algorithm (hereafter *NN method*). Next, we select the UAVs with maximum and minimum distance cost (UAV_{max} and UAV_{min} in Fig. 3) using Eqs. (7) or (8) when optimizing by distance or flight time, respectively. When the difference in the distance costs exceeds a given threshold (θ), the camera locations of UAV_{max} are inspected. Among them, the maximum attribution degree point w.r.t UAV_{min} ’ cluster is re-assigned to UAV_{min} .

$$f_i = \sum_{k=1}^n d_k \tag{7}$$

$$f_i = t_h(n - 1) + \sum_{k=1}^n t_k \tag{8}$$

$$t_k = \begin{cases} \frac{d_k}{v_{max}} + \frac{v_{max}}{a} & \left(d_k > \frac{v_{max}^2}{a} \right) \\ 2\sqrt{\frac{d_k}{a}} & \left(d_k < \frac{v_{max}^2}{a} \right) \end{cases} \tag{9}$$

where f_i is the cost function in step i ; notice that this can be the flight distance or flight time. When optimizing by flight distance, d_k denotes the distance between camera locations k and $k + 1$. When optimizing by flight time, t_k denotes the flight time between camera locations k and $k + 1$, and t_h denotes the hovering time at a point of acquisition. In both cases, n is the total number of acquisition points. Since each UAV is presumed to accelerate to its maximum speed with uniform acceleration and to hover at the camera location, t_k can be calculated as Eq. (9), where v_{max} is the maximum speed of a UAV, and a represents the uniform acceleration of a UAV.

Obstacle avoidance. The routes found through the procedure presented in the preceding subsections may pass through obstacles when connecting the home location to the first camera location of each path. Thus, the *A* algorithm* was implemented for obstacle avoidance in the cases where such a situation is present. We selected the *A* algorithm* for finding a path between two points because it is more efficient than the Dijkstra method. Therefore, when two connected points of a path cross an obstacle, adjacent intermediate points are generated between these two and an alternative route is constructed for that segment.

Finally, through the steps described above, the CPP for multiple UAVs is generated. Figure 4 presents an example of the output of the UAV flight routes of our example case of a virtual urban environment. Each point represents a camera location, and the edge color corresponds to the particular UAV to which that point is assigned. The points are linked by edges in accordance with the flight progression.

Model-based analysis

Analysis of clustering methods. To confirm the benefits of the FCM sorting method used in the proposed approach, it was compared to the *K-means* method. The environment shown in Fig. 2 was used for the comparative analysis. Five UAVs were used to survey 10 randomly selected structures at the site. After camera location points were generated for each structure, these were clustered using the *K-means* and FCM, optimizing the distance or flight time as explained in “**Obstacle avoidance**” section. In the *K-means* clustering case, the camera location points did not have attribution degree matrices, therefore, the cost of the TSP was minimized instead for each UAV. After a 100 trials of the above-described procedures, the results were examined for com-

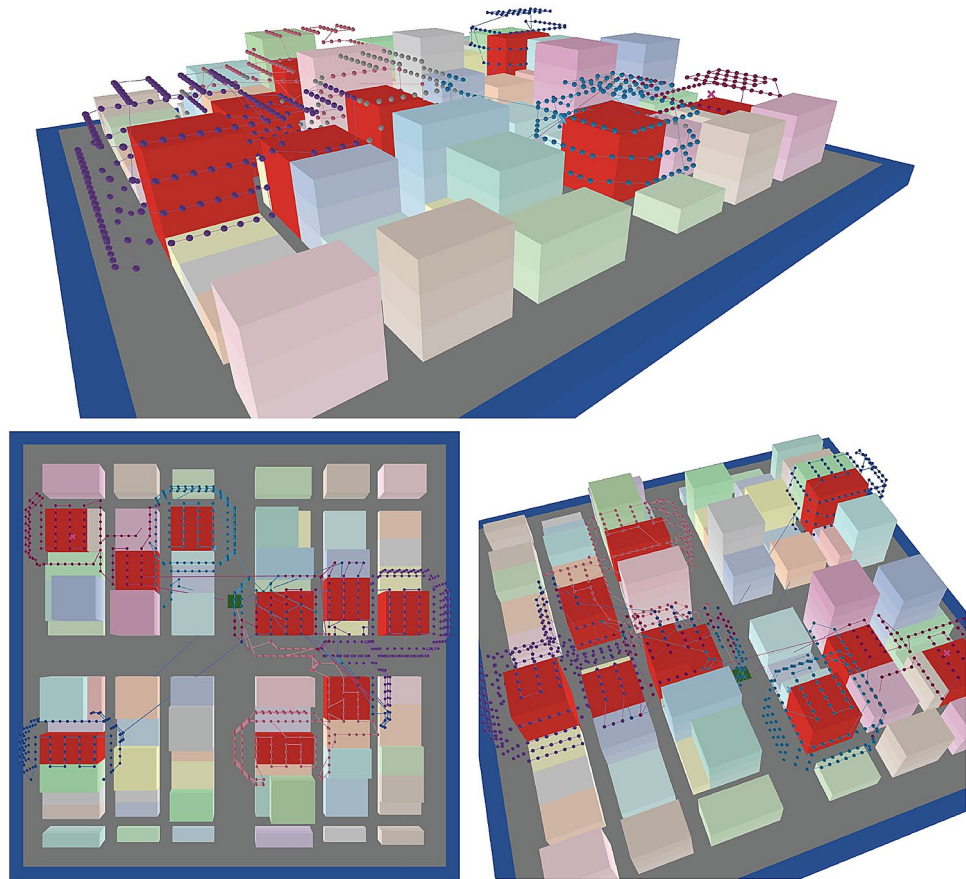


Figure 4. Final UAV routes produced using the proposed method. The green square at the center of the area indicates the home point for the UAVs. *Source:* This study, using NetLogo3D.

parison. The cost functions provided in both Eqs. (7) and (8) were considered. The parameters were as follows: the constant m in the FCM method was set to 2, the UAV hovering time t_h was assumed to be 2s, the maximum UAV speed v_{max} was 5m/s, and the uniform UAV acceleration a was $2m/s^2$. The total cost F was given as shown in Eq. (10). In addition, as another measure to evaluate clustering methods, the subtour averaging coefficient ρ is considered, as shown in Eq. (11).

$$F = \sum_{i=1}^5 f_i \quad (10)$$

$$\rho = \frac{\max f_i - \min f_i}{\max f_i} \quad (11)$$

Figure 5 shows the cost functions for each UAV in the FCM method. Figure 5—left shows the cost function calculated using Eq. (7), whereas Fig. 5—right shows the results with Eq. (8). These results suggest that the redistribution of the camera location points among the UAVs based on the attribution degree function of the FCM method drives the cost functions for all UAVs to approach the average value. Tables 1 and 2 present the average values of the cost functions given by Eqs. (7) and (8), respectively. These tables show that the FCM method yielded a smaller subtour averaging coefficient and more uniform solutions, whereas the K -means method found a slightly lower total cost. One reason for this result is that with the hard clustering of the K -means method, long paths between structures might not be generated. Another is that f_{min} increased more rapidly than f_{max} decreased with redistribution based on the attribution degree, as shown in Fig. 5. However, the $\max f_i$ column in Table 2 shows that the K -means method resulted in a cost of approximately 20 min on average, whereas redistribution with the FCM method reduced the cost to less than 15 min. The current capacity limit of UAV batteries is typically 10–30 min, depending on the equipment. The termination of UAV operation is determined by the UAV with the maximum cost. Therefore, it is better to have a smaller maximum cost. From the discussions presented above, we can conclude that the clustering of the camera location points in accordance with the FCM method is quite effective in making the costs for multiple UAVs minimal and uniform when solving the MTSP.

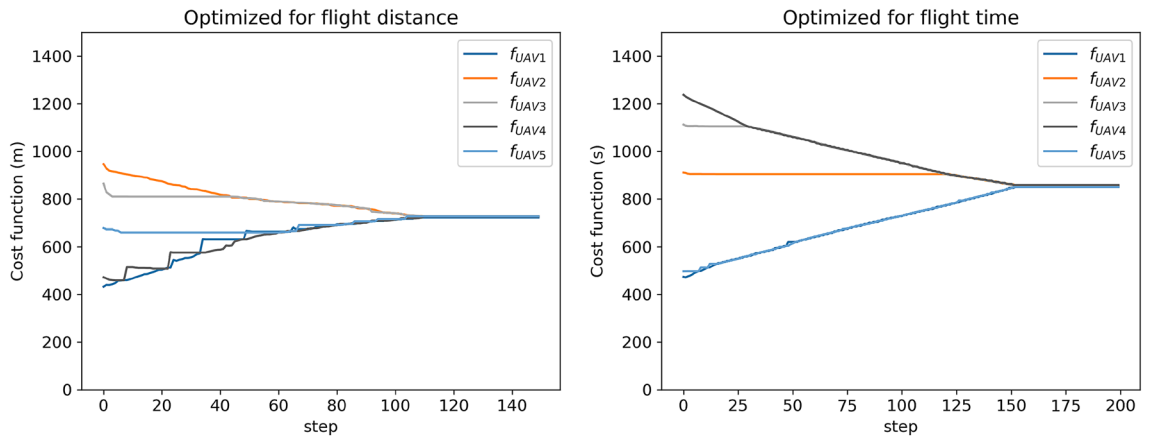


Figure 5. Changes in the cost functions f_i given by Eq. (7) (left) and Eq. (8) (right) with the redistribution of camera location points based on the degree of attribution.

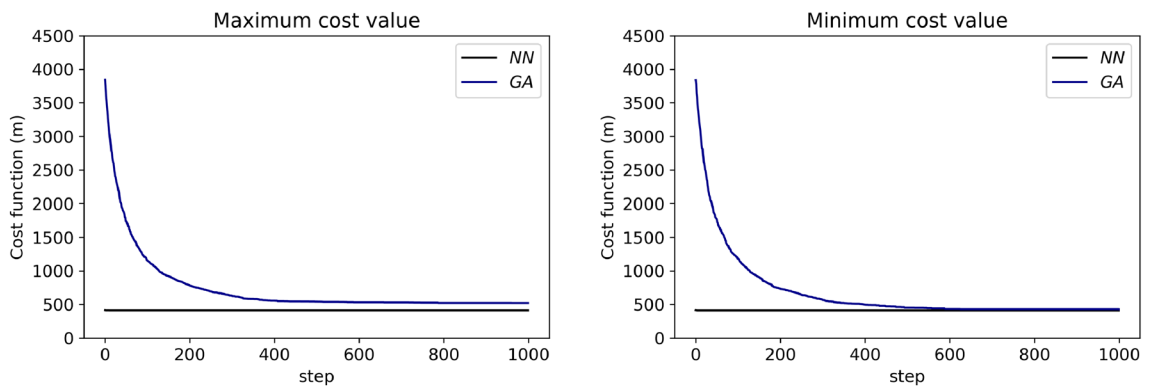


Figure 6. Convergence processes of the NN and GA methods resulting in convergence to the maximum cost value (left) and the minimum cost value (right).

	F	ρ	$\max f_i$	$\min f_i$
<i>K</i> -means	3236	0.545	902	405
Fuzzy <i>C</i> -means	3633	0.022	735	719

Table 1. Average values of evaluation functions when the flight distance is used as the cost function.

	F	ρ	$\max f_i$	$\min f_i$
<i>K</i> -means	4167	0.626	1218	455
Fuzzy <i>C</i> -means	4275	0.015	861	847

Table 2. Average values of the evaluation functions when the flight time is used as the cost function.

Analysis of the route optimization method. To examine the effectiveness of the proposed workflow including the NN method, the precision of the path solution is compared with that achieved using the genetic algorithm (GA). We use the same virtual environment presented in Fig. 2. The GA parameters used here are presented in Table 3. Because both methods use random numbers, to ensure a fair comparison, 100 trials were conducted using both methods. The cost function given in Eq. (7) was used here.

The results are presented in Table 4. Figure 6 illustrates the convergence of the cost function for both methods. Figure 6—left shows the case with the maximum converged value of the cost function among the 100 trials for each method, whereas Figure 6—right shows the case with the minimum converged value. Table 4 shows that the NN method produced more precise results. Basically, the time for one step is shorter in the 1.5-opt method than in the GA method. Ultimately, the method proposed in this study is approximately five times faster than the GA method. Figure 6—right demonstrates that even the initial solution provided by the NN method is more precise

Parameter	Value
Crossover design	One-point crossover
Number of genomes	300
Tournament rate	1%
Mutation rate	10%

Table 3. Parameters of the GA.

	NN method	GA method
$\min f_i$	410.9	428.2
$\max f_i$	411.7	522.0
Processing time (s)	29.3	146.7

Table 4. Precision of the TSP solutions obtained using the NN and GA methods.

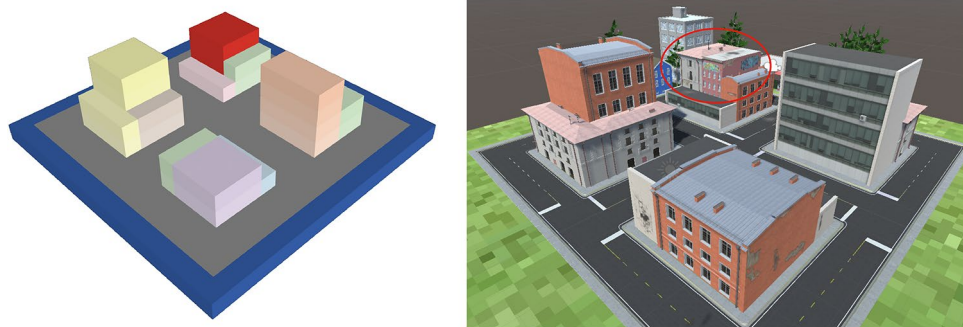


Figure 7. Left—Paths are generated to survey the red building from the NetLogo3D environment. Right—Blocks generated with Unity3D for quality assessment. *Source:* This study, using NetLogo3D and Unity 3D.

than the final solution of the GA method. Moreover, with the 1.5-opt method, convergence was attained in 20 steps, whereas the GA method took 600 steps to reach convergence. Therefore, the combination of the NN and 1.5-opt methods is superior to the GA method in terms of both precision and convergence speed. The reason might be that the combination of the NN and 1.5-opt methods may be particularly well suited to the current problem under examination, in which the camera location points are distributed regularly but with some deviations. A method in which the initial solution is obtained via the NN method but the subsequent improvement is achieved using the GA method requires more time for convergence than the current method. Improving the initial solution using the 1.5-opt method might be more compatible with the NN method than improvement via the GA method.

Analysis of the 3D reconstruction output quality. The proposed path generation method was examined in terms of its ability to enhance a more precise 3D model than that produced by a conventional overhead flight. We develop another virtual urban environment, as shown in Fig. 7, which was also generated using the Unity3D game engine³⁸, as shown in the right panel of the same figure. The red oval indicates the object to be surveyed. A flight route for one UAV was designed using the proposed method, including the coordinates of the camera location points, the camera orientation, and the order of the camera location points to be visited. The routes are shown in Fig. 8.

These flight routes were used in Unity3D to photograph the structure. The UAV took pictures at every camera location point on the flight path. The camera parameters are presented in Table 5. Jiang et al. (2020)³⁹ evaluated the performance of six SfM software packages. Among them, Pix4D demonstrated the highest accuracy and, therefore, is used in this study to generate our 3D models for both flights, and their respective results are compared. Table 6 presents the parameters applied in Pix4D. In addition, as a comparative examination, the graffiti and damage details are illustrated in Figs. 9 and 10 to examine the precision of their reproduction.

For the drawings of the walls shown in Fig. 9 both methods produced similar results, however, our results show better quality for the details, such as the window and the edge of the roof. This is expected since the overhead flight cannot acquire images of those spaces. Our results in the second case (Fig. 10) show even greater enhancement than the standard approach. Even on a shady side where only a few tie points could be extracted, the proposed method precisely reproduced cracks and graffiti.

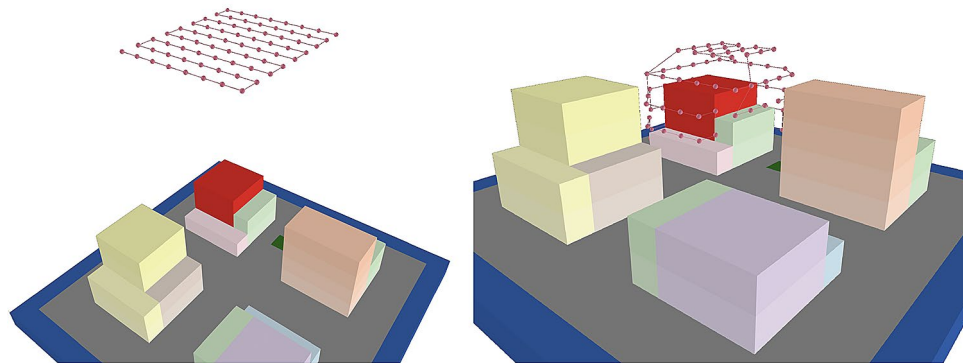


Figure 8. Left—Overhead flight route. Right—The path generated by this study. *Source:* This study, using NetLogo3D.

Discussion and future work

There is room for improvement of the proposed method, in particular when generating camera location points and redistributing those using the attribution degree. Regarding the former, the authors have proposed a rule-based camera location points generation method with sufficient overlap and sidelap ratios to achieve fast generation of camera location points. As a result of this method, erroneous extraction of tie points can occur. Additionally, a tendency toward very poor estimation of the camera position and orientation for short distances and multiple photographs is apparent. Therefore, the optimal photography points for the target objects must be generated after solving the network designing problem¹². Regarding the redistribution of camera location points, the results show that the UAV costs when applying FCM are more uniform than those achieved by the *K*-means method. However, the *K*-means method is superior in terms of the total and average costs. In the examination performed above, the redistribution of the camera location points was performed for five UAVs. If camera location points are to be sorted among more UAVs, groups of clusters that are not mutually close will tend to appear, and long paths might be generated through redistribution using the same cost function and degree of attribution settings used in the present study. To avoid this pitfall, one might renew the centroid m of the clusters in the algorithm used for the MTSP by providing a weight coefficient w_i for the Euclidean norm of the *K*-means method depending on the cost calculated for the MTSP. In this way, the method proposed herein could be extended to more UAVs, allowing it to yield better results in terms of the total and average costs compared to the method for solving the TSP for each UAV in the *K*-means clustering. Another shortcoming of the present proposal is the practical problem that it cannot respond quickly to a cost greater than the threshold cost. This shortcoming derives from the fact that a long computation time is necessary to achieve convergence of the UAV costs. Only in the final stage of processing can it be known whether the cost exceeds the threshold cost. To overcome this difficulty, one might predetermine the number of UAVs corresponding to the generated number of camera location points. However, if the converged cost value exceeds the threshold value, then the calculations must still be repeated for an increased number of UAVs.

A method that can uniformly minimize the costs for several tens of UAVs would be useful for developing methods for surveying much wider areas. The ability to provide highly precise 3D information about disasters over a wide area is expected to contribute greatly to the preparation of countermeasures to mitigate the effects of future disasters. In addition, the present study provides a model-based analysis using a virtual environment, however, real experiments should be performed in future work to further evaluate the application of the proposed workflow.

Concluding remarks

We have proposed a path planning method for multiple UAVs to aid in 3D building damage surveys or disaster situations. The proposed methodology combines the fuzzy *C*-means method for assigning camera location points to each UAV with a route optimization algorithm for calculating the visit order of the camera location points for each UAV by solving the multiple traveling salesman problem. Finally, the selected paths are corrected to avoid obstacles between the camera location points using the A* algorithm. To assess the effectiveness of the proposed method, a comparative study of the quality of 3D reconstruction models was conducted, where the enhanced results confirm the superiority of our proposed approach over a conventional overhead flight mapping technique. We believe, the proposed method can improve reconnaissance capabilities in increasingly complex urban disasters.



Figure 9. Comparison of two graffiti drawings. The top panel in each pair of images shows the results from the method proposed in this study, while the bottom panel shows the results of the overhead flight method. *Source:* This study, using Unity 3D.

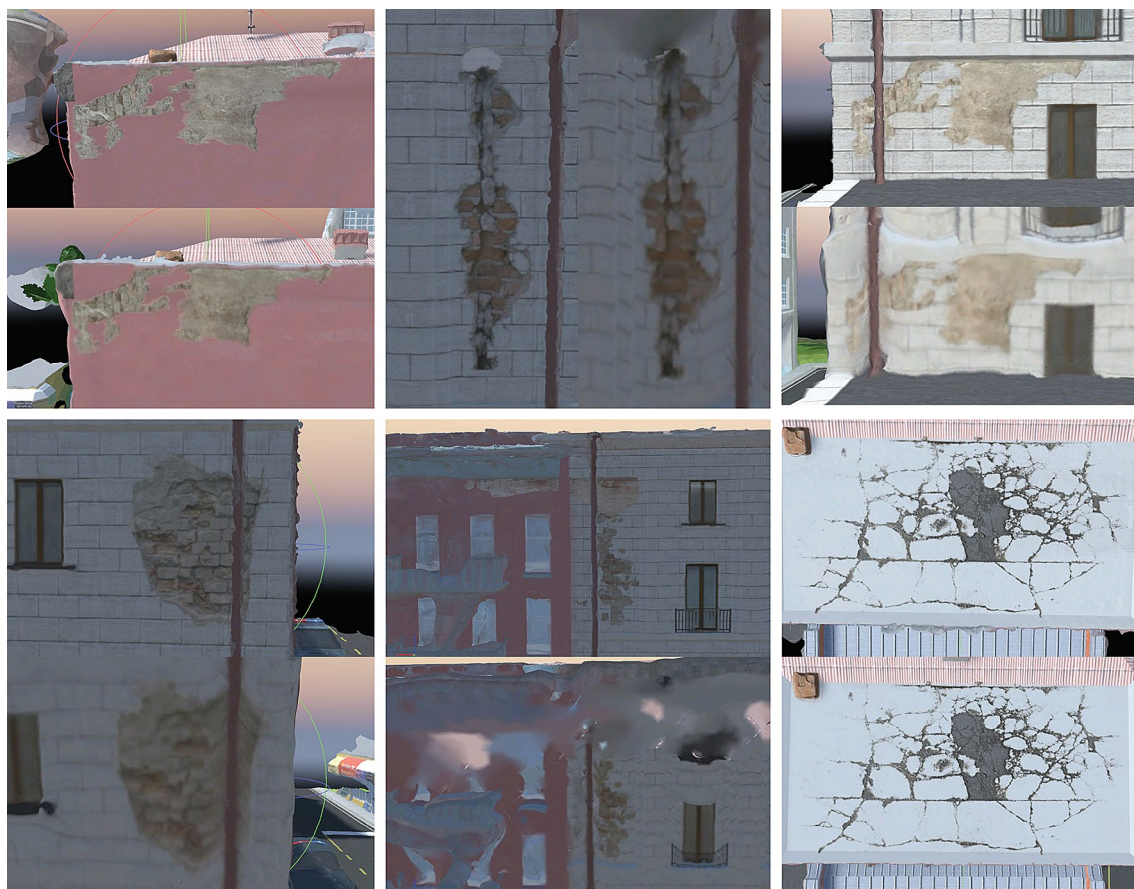


Figure 10. Comparison of the extraction of six damage features. The top or left panels in each pair of images shows the results from the method proposed in this study, while the bottom or right panel shows the results of the overhead flight method. *Source:* This study, using Unity 3D.

Parameter	Value
Visibility (vertical)	70
Size of sensor x	70
Size of sensor y	51
Focal length	36.418
Aspect ratio	4:3
Resolution	1440 × 1080

Table 5. Camera parameters.

Parameter	Proposed method	Conventional method
Number of images	145	144
Quality	Standard	Standard
Coordinate system	Orthogonal	Orthogonal
Horizontal accuracy (m)	0.01	0.01
Vertical accuracy (m)	0.02	0.02

Table 6. Parameters of Pix4D.

Received: 16 December 2020; Accepted: 31 August 2021

Published online: 20 September 2021

References

- Rathje, E. & Adams, B. The role of remote sensing in earthquake science and engineering: Opportunities and challenges. *Earthq. Spectra* **24**(2), 471–492. <https://doi.org/10.1193/1.2923922> (2008).
- Morita, T. *et al.* Decision making support of UAV path planning for efficient sensing in radiation dose mapping. In *2018 IEEE 42nd Annual Computer Software Applied Conference (COMPSAC)*, 333–338. <https://doi.org/10.1109/compsac.2018.00053> (2018).
- Koshimura, S., Moya, L., Mas, E. & Bai, Y. Tsunami damage detection with remote sensing: A review. *Geosciences* **10**(5), 177. <https://doi.org/10.3390/geosciences10050177> (2020).
- Tahir, A. *et al.* Swarms of unmanned aerial vehicles—A survey. *J. Ind. Inf. Integr.* **16**, 100106. <https://doi.org/10.1016/j.jii.2019.100106> (2019).
- Moya, L. *et al.* Detecting urban changes using phase correlation and L1-based sparse model for early disaster response: A case study of the 2018 Sulawesi Indonesia earthquake-tsunami. *Remote Sens. Environ.* **242**, 111743. <https://doi.org/10.1016/j.rse.2020.111743> (2020).
- Moya, L. *et al.* 3D gray level co-occurrence matrix and its application to identifying collapsed buildings. *ISPRS J. Photogramm. Remote Sens.* **149**, 14–28. <https://doi.org/10.1016/j.isprsjprs.2019.01.008> (2019).
- Nex, F., Duarte, D., Steenbeek, A. & Kerle, N. Towards real-time building damage mapping with low-cost UAV solutions. *Remote Sens. (Basel)* **11**, 287. <https://doi.org/10.3390/rs11030287> (2019b).
- Li, M. *et al.* Application and typical case of unmanned aerial vehicles combined with building 3D modeling. *J. Phys. Conf. Ser.* **1746**, 012012. <https://doi.org/10.1088/1742-6596/1746/1/012012> (2021).
- Verykokou, S. *et al.* UAV-based 3D modelling of disaster scenes for Urban Search and Rescue. In *2016 IEEE International Conference on Imaging Systems and Techniques (IST)*, 106–111. <https://doi.org/10.1109/ist.2016.7738206> (2016).
- Pix4D: Pix4Dcapture. (23 Apr. 2021). <https://www.pix4d.com/product/pix4dcapture>
- Agisoft LLC, Agisoft PhotoScan. (23 Apr. 2021). <http://www.agisoft.ru/products/photoscan>
- Hepp, B., Niefßner, M. & Hilliges, O. Plan3D: Viewpoint and trajectory optimization for aerial multi-view stereo reconstruction. *ACM Trans. Graph* **38**(1), 1–17. <https://doi.org/10.1145/3233794> (2018).
- Nex, F. & Remondino, F. UAV for 3D mapping applications: A review. *Appl. Geomat.* **6**, 1–15. <https://doi.org/10.1007/s12518-013-0120-x> (2014).
- Okada, G., Moya, L., Mas, E. & Koshimura, S. The potential role of news media to construct a machine learning based damage mapping framework. *Remote Sens. (Basel)* **13**, 1401 (2021).
- Bai, Y. *et al.* Pyramid pooling module-based semi-siamese network: A benchmark model for assessing building damage from xBD satellite imagery datasets. *Remote Sens.* **12**, 4055 (2020).
- Kolling, A., Kleiner, A. & Rudol, P. Fast guaranteed search with unmanned aerial vehicles. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* 6013–6018. <https://doi.org/10.1109/iros.2013.6697229> (2013).
- Zheng, X., Wang, F., Xia, J. & Gong, X. The methodology of UAV route planning for efficient 3D reconstruction of building model. In *2017 25th International Conference on Geoinformatics*, 1–4. <https://doi.org/10.1109/geoinformatics.2017.8090905> (2017).
- Zheng, X., Wang, F. & Li, Z. A multi-UAV cooperative route planning methodology for 3D fine-resolution building model reconstruction. *ISPRS J. Photogramm. Remote Sens.* **146**, 483–494. <https://doi.org/10.1016/j.isprsjprs.2018.11.004> (2018).
- Koch, T., Körner, M. & Fraundorfer, F. Automatic and semantically-aware 3D UAV flight planning for image-based 3D reconstruction. *Remote Sens. (Basel)* **11**, 1550–1578. <https://doi.org/10.3390/rs11131550> (2019).
- Schermer, D., Moeni, M. & Wendt, O. A hybrid VNS/Tabu search algorithm for solving the vehicle routing problem with drones and en route operations. *Comput. Oper. Res.* **109**, 134–158. <https://doi.org/10.1016/j.cor.2019.04.021> (2019).
- Jain, A. K., Murty, M. N. & Flynn, P. J. Data clustering: A review. *ACM Comput. Surv. (CSUR)* **31**, 264–323. <https://doi.org/10.1145/331499.331504> (1999).
- Barrientos, A. *et al.* Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots. *J. Field Robot.* **28**, 667–689. <https://doi.org/10.1002/rob.20403> (2011).
- Bouaid, Y., Bestaoui, Y. & Siguerdidjane, H. Guidance-control system of a quadrotor for optimal coverage in cluttered environment with a limited onboard energy: Complete software. *J. Intell. Robot. Syst.* **95**, 707–730. <https://doi.org/10.1007/s10846-018-0914-5> (2019).
- Gao, C. *et al.* Optimal multirobot coverage path planning: Ideal-Shaped Spanning Tree. *Math. Probl. Eng.* **2018**, 1–10. <https://doi.org/10.1155/2018/3436429> (2018).
- Ju, C. & Son, H. I. Multiple UAV systems for agricultural applications: Control, implementation, and evaluation. *Electronics* **7**, 162. <https://doi.org/10.3390/electronics7090162> (2018).
- Ge, F., Li, K., Han, Y., Xu, W. & Wang, Y. Path planning of UAV for oilfield inspections in a three-dimensional dynamic environment with moving obstacles based on an improved pigeon-inspired optimization algorithm. *Appl. Intell.* **50**, 2800–2817. <https://doi.org/10.1007/s10489-020-01650-2> (2020).
- Cabreira, T. M., Brisolara, L. B. & Ferreira, P. R. Survey on coverage path planning with unmanned aerial vehicles. *Drones* **3**, 4. <https://doi.org/10.3390/drones3010004> (2019).
- Zhao, J. & Zhao, J. Study on multi-UAV task clustering and task planning in cooperative reconnaissance. In *2014 Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics*, Vol. 2, 392–395. <https://doi.org/10.1109/IHMSC.2014.196> (2014).
- Botteghi, N., Kamilaris, A., Sinai, L. & Sirmacek, B. Multi-agent path planning of robotic swarms in agricultural fields. *ISPRS Ann Photogramm Remote Sens Spatial Inf Sci* **1**, 361–368. <https://doi.org/10.5194/isprs-annals-V-1-2020-361-2020> (2020).
- Mansouri, S. S. *et al.* Cooperative coverage path planning for visual inspection. *Control Eng Pract* **74**, 118–131. <https://doi.org/10.1016/j.conengprac.2018.03.002> (2018).
- Kong, S. *et al.* Multi-objective path planning based on K-D fusion algorithm. *J. Phys. Conf. Ser.* **1828**, 012037. <https://doi.org/10.1088/1742-6596/1828/1/012037> (2021).
- Bouras, A., Bouaid, Y. & Guiatni, M. Two-scale algorithm to plan coverage paths for multi-UAVs. In *2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-being (IHSH)*, 193–198. <https://doi.org/10.1109/IHSH51661.2021.9378755> (2021).
- Xiao, S., Tan, X. & Wang, J. A simulated annealing algorithm and grid map-based UAV coverage path planning method for 3D reconstruction. *Electronics* **10**, 853. <https://doi.org/10.3390/electronics10070853> (2021).
- Ali, Z. A., Han, Z. & Masood, R. J. Collective motion and self-organization of a swarm of UAVs: A cluster-based architecture. *Sensors* **21**, 3820. <https://doi.org/10.3390/s21113820> (2021).
- Shao, S., Peng, Y., He, C. & Du, Y. Efficient path planning for UAV formation via comprehensively improved particle swarm optimization. *ISA Trans* **97**, 415–430. <https://doi.org/10.1016/j.isatra.2019.08.018> (2020).
- Ferwom, A., Tran, J., Ufkes, A. & D'Souza, A. Initial experiments on 3D modeling of complex disaster environments using unmanned aerial vehicles. *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics* 167–171. <https://doi.org/10.1109/ssrr.2011.6106781> (2011).

37. Wilensky, U. NetLogo. <http://ccl.northwestern.edu/netlogo> (Center for Connected Learning and Computer-based Modelling, Northwestern University, Evanston, IL, 1999).
38. Haas, J. K. A history of the unity game engine. Interactive Qualifying Project. Worcester Polytechnic Institute. <https://web.wpi.edu/Pubs/E-project/Available/E-project-030614-143124/> (2014).
39. Jiang, S., Jiang, C. & Jiang, W. Efficient structure from motion for large-scale UAV images: A review and a comparison of SfM tools. *ISPRS J. Photogramm.* **167**, 230–251. <https://doi.org/10.1016/j.isprsjprs.2020.04.016> (2020).

Acknowledgements

This study was partly funded by the Japan Society for the Promotion of Science (JSPS) Kakenhi Program (17H06108 and 21H05001); the Core Research Cluster of Disaster Science; the Tough Cyberphysical AI Research Center at Tohoku University, and the National Program for Scientific Research and Advanced Studies (PROCIENCIA/CONCYTEC - PERU) [contract number 038-2019 FONDECYT-BM-INC-INV].

Author contributions

Conceptualization, E.M., R.N., S.K.; Methodology, R.N., E.M., L.M., S.K.; Analysis, R.N., E.M.; Resources, S.K., E.M.; Writing, E.M., R.N., L.M.; Supervision, E.M., S.K.; Project administration, S.K.; Funding acquisition, S.K. All authors have read and agreed to the published version of the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1038/s41598-021-97804-4>.

Correspondence and requests for materials should be addressed to E.M.

Reprints and permissions information is available at www.nature.com/reprints.

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



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2021