Received: 30 August 2018 | Revised: 22 May 2019 | Accepted: 1 July 2019

DOI: 10.1111/coin.12232

SPECIAL ISSUE ARTICLE

WILEY Computational Intelligence

Ground object recognition and segmentation from aerial image-based 3D point cloud

Katsuya Ogura[1] | **Yuma Yamada**[1] | **Shugo Kajita**[1] | **Hirozumi Yamaguchi**[1] | **Teruo Higashino**[1] | **Mineo Takai**[1,2]

[1]Graduate School of Information Science and Technology, Osaka University, Osaka, Japan

[2]Computer Science Department, University of California, Los Angeles, California

Correspondence
Katsuya Ogura, Graduate School of Information Science and Technology, Osaka University, Osaka 565-0871, Japan.
Email: k-ogura@ist.osaka-u.ac.jp

Present Address
Katsuya Ogura, Osaka University, 1-5 Yamadaoka, Suita, Osaka 565-0871, Japan

Funding information
Empirical Research and Development for Solving Regional Issues by Data Utilization, Grant/Award Number: 200; National Institute of Information and Communications Technology

Abstract

Several attempts have been made to grasp three-dimensional (3D) ground shape from a 3D point cloud generated by aerial vehicles, which help fast situation recognition. However, identifying such objects on the ground from a 3D point cloud, which consists of 3D coordinates and color information, is not straightforward due to the gap between the low-level point information (coordinates and colors) and high-level context information (objects). In this paper, we propose a ground object recognition and segmentation method from a geo-referenced point cloud. Basically, we rely on some existing tools to generate such a point cloud from aerial images, and our method tries to give semantics to each set of clustered points. In our method, firstly, such points that correspond to the ground surface are removed using the elevation data from the Geographical Survey Institute. Next, we apply an interpoint distance-based clustering and color-based clustering. Then, such clusters that share some regions are merged to correctly identify a cluster that corresponds to a single object. We have evaluated our method in several experiments in real fields. We have confirmed that our method can remove the ground surface within 20 cm error and can recognize most of the objects.

KEYWORDS

drone, outdoor recognition, point cloud, segmentation, 3D objects

1 | INTRODUCTION

Recently, Structure from Motion (SfM)[1] techniques, which are known as a sort of "photogrammetry methods", have been more attractive. They generate 3D point clouds that contain 3D coordinates and the colors using multiple aerial images and such information, ie, photographing angles obtained from global positioning system (GPS) and sensors. These techniques are expected to facilitate shape grasping tasks, ie, ground object surveying.

The SfM can be utilized in various research fields. For example, wireless sensor networks are often deployed in forest and mountainous areas to monitor the ecosystem. In such a system, sensed data are aggregated to cloud servers via wireless communications. For covering wide areas, wireless stations should carefully be placed considering severe signal attenuation by trees and other objects. Simulation techniques are often used for the optimal placement design of wireless stations, and SfM techniques with aerial images taken by a drone are helpful in building the model of such environment into the simulation environment.

The SfM can also be utilized for understanding the damage by disasters. When a large-scale earthquake occurs in city areas, many streets are closed and traffic congestion frequently occurs due to mass of walking people. In such an extreme situation, it is not easy to fully investigate the damage of buildings (eg, collapse of houses), but drones are often useful to quickly capture images from the sky. By visualizing and comparing the 3D survey results before and after the disaster, we can efficiently grasp building collapse and inclination. The 3D object information can also be used for simulation of wireless communication to estimate performance in real field. We can perform more precise simulations that reproduce radio signal attenuation by individual objects appearing in the target field.

However, point clouds generated by SfM methods usually contain 3D coordinates and color information only. From such a mass of points, we need to identify 3D points that correspond to each object (building, vehicle, tree, etc.) to reproduce the situations on the ground for such purposes, ie, situation awareness, analysis, and simulations.

To tackle this issue, in this paper, we propose a ground object recognition method from a 3D point cloud that captures the heights of ground surface. Basically, we rely on the existing tools to generate such a 3D point cloud from aerial images, and our method tries to give semantics to each set of clustered points. In the proposed method, firstly, such points that correspond to the ground surface are eliminated using the elevation data from the Geographical Survey Institute. Next, we apply an interpoint distance-based clustering and noise filtering method according to the point density of each cluster. We also apply color-based clustering to divide adjacent objects. Then, such clusters that share some regions are merged to correctly identify a point cluster that corresponds to each object. Finally, a filtering method is applied based on the knowledge on the sizes of objects.

We have evaluated our method in the following four case studies. Firstly, to evaluate the fundamental performance of our method, the ground surface removal accuracy was evaluated using a real single object (vehicle) on the plane ground. We have used images taken by a drone from different angles. The error of the estimated object height after ground removal was less than 20 cm. Secondly, we have verified the capability of building recognition using real photos provided by ESRI. As a result, the 3D point group that corresponds to the building was correctly identified even with trees around the building. Thirdly, assuming a disaster scenario, we have tried to detect the height difference of houses before and after the houses collapse. This often happens in Japan and other countries by severe earthquakes, and fast recognition of house damage levels is quite significant for timely disaster relief. Finally, we have applied our method to recognize a single

story building as well as a number of trees in a mountainous area, which was the more challenging case among the four. We have confirmed that the ground surface was mostly removed even under undulating situations, and most part of the building was correctly identified.

We would like to note that, in our conference paper,[2] we have presented a preliminary algorithm for ground surface removal. In this paper, we extend the algorithm so that we further use color information for better accuracy. We have also evaluated the impact of accuracy of the terrain data on the object recognition.

2 | RELATED WORK

Object detection by drone

Many attempts have been made to grasp the situation using drones in various regions and cases. For example, Arifin et al[3] present a method of detecting and tracking objects with specific shapes and colors in drone flight. This method applies some algorithms such as Hough transformation to the images taken by drones. In addition, Chen et al[4] provide an algorithm for detecting and tracking objects by a small drone. The algorithm uses the inertial data from the drone to assist a series of processes to be done in the drone with limited power. Yasuoka et al[5] apply machine learning to indoor images and enables to recognize indoor objects such as home electronics in the room. van Gemert et al[6] use a drone for nature conservation. They attempt to identify animals and count them by machine learning processing on images and videos taken by drones. Maria et al[7] propose a method for vehicle detection from videos captured by drones in an urban environment.

2.1 | Ground surface extraction for point cloud analysis

In our method, given a 3D point cloud, we have to eliminate those points corresponding to the ground surface to identify objects on it. For this purpose, several methods have been investigated so far. For a flat ground area, we can treat the ground surface as a single, large plane and can extract it by applying the random sample consensus (RANSAC) algorithm.[8] Schnabel et al[9] propose an improved version of the RANSAC algorithm to extract basic features such as planes from a given point cloud to pursue reliability and efficacy. Zhang et al[10] propose a method to extract the ground surface by scanning the horizontal direction of the point cloud with a certain window size and calculating the relative positions based on the interpoint height difference in the window.

2.2 | Object classification for point cloud

There are several studies on a method for determining attributes of objects represented by point cloud. Golovinskiy and Funkhouser[11] propose a method to identify the target object by using the graph, which is generated based on the interpoint distance after removing the ground surface from the point cloud. Ramiya et al[12] focus on urban environment and classify each point in the point cloud generated by light detection and ranging (LiDAR) into five types (Road, Lawn, Flat Roof Building, Gable Roof Building, and Tree).

2.3 | Our contribution

The conventional methods to remove the ground surface are not sometimes applicable to undulating plane cases because they employ only the information about the distribution of points in

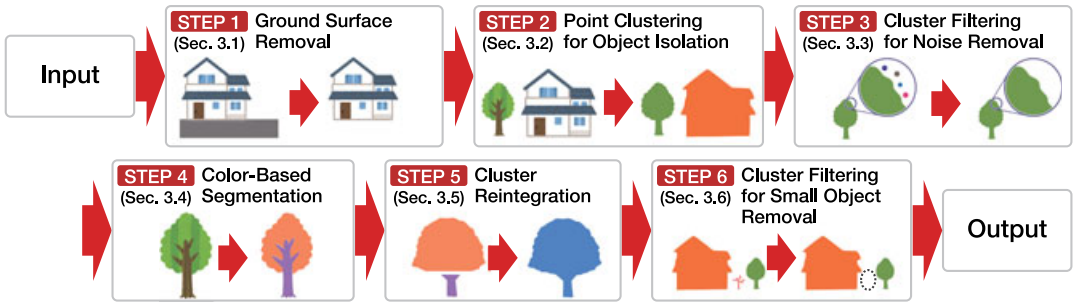

FIGURE 1 Workflow of our methodology [Color figure can be viewed at wileyonlinelibrary.com]

the point clouds. On the contrary, our method extracts the ground surface based on the data provided by the Geographical Survey Institute. It enables to extract the ground surface with higher accuracy even in mountainous field. Regarding the extraction of the point group that corresponds to each object from a given point cloud, our method tackles a challenging problem where we deal with *3D point clouds generated from aerial photographs*. Due to the limitation of SfM, such a point cloud often includes noisy data, eg, the shade of building roofs may generate sparsely distributed points, which are to be eliminated. As a result, such a case may happen where one building is recognized as multiple parts by simply applying clustering methods. We provide more intelligent and data-specific processing such as noise filtering based on the cluster size and shape, and we have proved our method outperforms the others in real field data as well as those in some controlled environment.

3 | GROUND OBJECT RECOGNITION

To divide the point cloud into real-world object units, we provide six procedures: (1) ground surface removal, (2) points clustering based on interpoint distance, (3) filtering point clusters, (4) points clustering based on color information, (5) point cluster reintegration, and (6) filtering small objects. The workflow is shown in Figure 1. After the removal of the ground surface (step 1), we apply interpoint distance-based clustering to recognize ground objects (step 2). The clustering result usually contains small "noise clusters" because the original point cloud generated from drone aerial images contains a lot of noises. We apply our noise cluster filtering to solve this problem (step 3). Then, since the interpoint distance-based clustering cannot perfectly distinguish multiple objects close to each other, we further apply color-based clustering to divide a cluster that covers adjacent multiple objects (step 4) into corresponding ones. On the contrary, due to the lack of points by roof shade, etc, a single object is likely to be recognized as multiple parts. To avoid this undesirable situation, we reintegrate clusters based on the intercluster overlapping in a two-dimensional (2D) plane (step 5). Finally, we remove such small clusters (such as shrubberies), which are not informative (step 6).

There are some tools for point cloud generation by integrating aerial photograph images from different angles based on SfM technology. We have chosen one of typical Open Source Software (OSS) tools, named OpenDroneMap,[13] but we should note that our method does not depend on particular tools.

In this section, we describe the details of our proposed method.

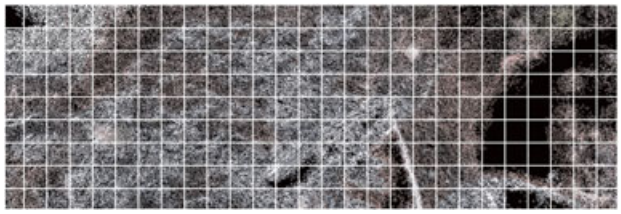

FIGURE 2 Comparison between point cloud and elevation data in density [Color figure can be viewed at wileyonlinelibrary.com]

3.1 | Ground surface removal

To distinguish objects on the ground, it is necessary to remove the *continuity* of point distribution between objects at different positions in the point cloud. To this end, we extract the points corresponding to the ground surface to identify those for ground objects.

Our approach to extract the ground surface uses the numerical elevation model[14] provided by the Geospatial Information Authority of Japan. The numerical elevation model provides the elevation data for every 5×5 m square. This elevation data represents the height from the sea level to the ground surface. When the difference between the z coordinate value of each point and the elevation data just below is smaller than a certain threshold value, it is estimated as the ground surface. Specifically, this estimation is performed in the following two steps.

3.1.1 | Elevation data interpolation

The density of the sampling points in the elevation data is largely insufficient (Figure 2). This is because the point cloud generated from images are distributed in units of a few centimeters, whereas the numerical elevation model is only provided every 5×5 m square. Therefore, we interpolate the elevation data at arbitrary 2D coordinates based on the surrounding sample points so that we can compare the z coordinate value of each point with the elevation data at the same xy-coordinates.

h_P is the interpolation of the elevation at the coordinates $P(p_x, p_y)$ on any plane. We derive h_P from the four peripheral points provided by the numerical elevation model using linear interpolation.

Let (l, t), $(l + D, t)$, $(l, t + D)$, and $(l + D, t + D)$ denote the XY-coordinates of the four points around this point, and h_A, h_B, h_C, and h_D ($l \leq p_x < l + D, t \leq p_y < t + D$) denote the elevation at these points, respectively. D represents the granularity of the numerical elevation model. For example, the numerical elevation data is $D = 5$ [m]. We derive the interpolated elevation of h_P as follows.

$$h_P = (1 - r_x)(1 - r_y)h_A + r_x(1 - r_y)h_B + (1 - r_x)r_y h_C + r_x r_y h_D, \tag{1}$$

where

$$r_x = \frac{p_x - l}{D}, \quad r_y = \frac{p_y - t}{D}.$$

3.1.2 | Threshold definition based on histogram

In our approach, if the difference between the z coordinate value of each point and the corresponding elevation data is smaller than a certain threshold value, the point is estimated as that on the ground surface. In this section, we describe how to define the threshold based on the relative value of the z value of the point and the elevation data after interpolation.

When the point cloud is generated from images taken by a drone, the origin of the z coordinate in each point may not match that of the elevation value in the numerical elevation model due to

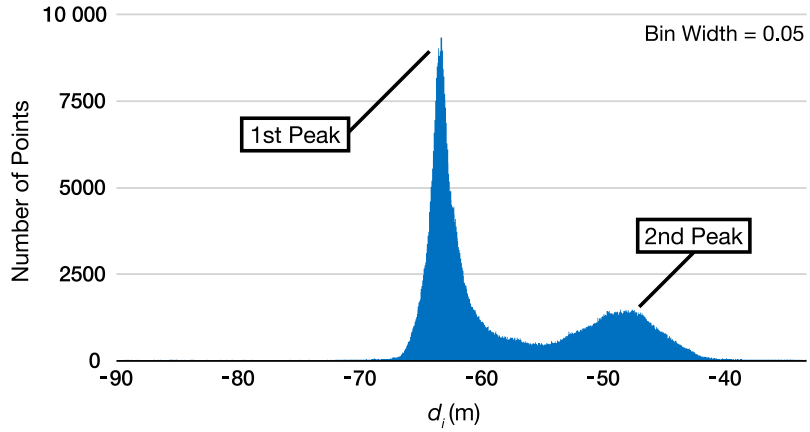

FIGURE 3 An example of a histogram that summarizes the distribution of differences in z coordinates (d_i) [Color figure can be viewed at wileyonlinelibrary.com]

the altitude error of the drone. Therefore, we derive the difference d_i between the z coordinate value and the elevation value $h_{(p_x,p_y)}$ for all points $P_i(p_x, p_y, p_z)$ in the point cloud and estimate the ground surface by comparing d_i with a threshold obtained by the distribution of d_i.

$$d_i = p_z - h_{(p_x,p_y)}. \tag{2}$$

The following explains how to determine the threshold value. If the point cloud includes the ground surface, we may obtain the distribution of d_i in a histogram like Figure 3.* We can see the highest peak called the 1st peak at rather small d_i value, which shows such points that represent the ground surface. Generally, aerial images capture the ground surface, and that is the reason why the histogram forms such a peak, and d_i at the peak means the maximum likelihood estimation of the errors. Then, the next peak with larger d_i but lower height than the 1st, called the 2nd peak, consists of points representing objects such as buildings. Considering the fact, we set the threshold including the 1st peak to extract the ground surface. We note that both the elevation data and point clouds contain errors and this motivates us to determine the threshold adaptively to the observed data. We assume that the error distribution follows the normal distribution and set the threshold based on the assumption.

When setting the threshold, we consider d_i of the 1st peak as the average μ' and derive the pseudostandard deviation σ' from only the samples on the left side of the peak. We do not use the right side of the 1st peak because it consists of not only those points representing the ground surface but also points representing objects. The following equation reflects the above consideration.

$$\sigma' = \sqrt{\frac{1}{2N+M} \times 2 \sum_{d_i \le \mu'} (d_i - \mu')^2} \tag{3}$$

$$\simeq \sqrt{\frac{1}{N} \sum_{d_i \le \mu'} (d_i - \mu')^2}. \tag{4}$$

We note that N and M are the number of points that satisfy $d_i < \mu'$ and $d_i = \mu'$, respectively. We regard that the right side of the 1st peak has the same distribution as the left side, and we derive σ' from the expression (3). We use the approximated expression (4) because M is much smaller than N when the bin width is sufficiently small. The threshold is given by $\mu' + z\sigma'$ ($z > 0$), where z is a parameter. We set about 2.8 to z based on our experience.

*This histogram is for the point cloud used in the verification in Section 4.4.

3.2 | Point clustering for object isolation

For clustering of points, the method uses the Euclidean Cluster Extraction method,[15] which performs clustering based on the interpoint distance. This is based on the fact that two points of different objects have some distances between them.

The Euclidean Cluster Extraction method needs to set the threshold value, d_{th}. Basically, we set a starting point, search the points in the region within d_{th} from the starting point, and add the points in the same group. We perform this process for all points in the group. When we do not find new points belonging to the same group, we regard the group as a cluster. We can also set the minimum number of points in a cluster, which enables to exclude isolated points that do not have points within the threshold.

The Euclidean Cluster Extraction method can correctly separate a cluster that corresponds to each object when the distance between objects is larger than d_{th}. Obviously, with larger d_{th}, neighboring objects are likely to be recognized as the same object; and with smaller d_{th}, two points representing the same object may be separated. We take an approach of producing more clusters by smaller d_{th} and finally integrating clusters close to each other into a single one. Therefore, we set about 0.3 to 0.5 m to d_{th}. We also set 3 to the minimum number of points in a cluster.

3.3 | Cluster filtering for noise removal

Generally, point clouds include some points irrespective of the actual objects, which may often greatly affect recognition performance. Outlier points are such points. They are mostly distributed in low-density space. Thus, most outliers do not form clusters and are removed in the Euclidean Cluster Extraction method (Section 3.2), but some outliers may form small clusters. Fortunately, the clusters by outlier points are mostly composed of at most several points, and accordingly, the size is considerably small, which can easily be eliminated.

On the other hand, a part of the ground surface may slightly remain after the ground surface removal of Section 3.1 depending on the ground condition (eg, heavy snow) and the accuracy of ground surface removal. They are called "Residue" point. Clusters by residue points have similar z coordinates close to the ground level but may have a planar spread, which results in larger clusters. This makes hard to distinguish them from the object clusters. Therefore, for noise removal, we observe the values in the z-range and the area of the distribution region to determine the types of the clusters.

The region of a cluster is regarded as a convex hull formed by the 2D point clouds when focusing on only x, y components of the points (Figure 4). A convex hull is a convex polygon with the smallest region enclosing a set of points. The convex hull is uniquely determined for a set of points, and in the 2D plane, the time complexity is $O(n \log h)$, where n is the number of input points and h is the number of points in the hull.[16] The following value P_{range} is used for noise determination of each cluster

$$P_{\text{range}} = \min\left(\sqrt{S}, z_{\max} - z_{\min}\right), \tag{5}$$

where S is the area of the convex hull for the xy planar distribution region of a cluster. $z_{\max}$ and $z_{\min}$ are the maximum and minimum z values of all the points in the cluster, respectively. In comparison between the area and the z-width, we use the square root of S and adjust a base unit to z-width.

The filtering removes clusters whose volume is within P_{range}^3 or clusters whose height is within P_{range}. Based on our experience, we set about 0.3 m to P_{range}.

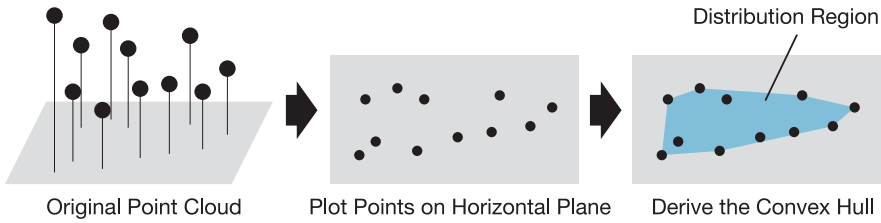

FIGURE 4 Distribution region of point cloud [Color figure can be viewed at wileyonlinelibrary.com]

3.4 | Color-based segmentation

Interpoint distance-based clustering (Section 3.2) cannot distinguish adjacent objects. Therefore, our method applies additional clustering based on color information. Color-based clustering divides objects with different colors, for example, house and trees into multiple clusters.

Our method uses HLS color space, which consists of three components of Hue, Lightness, and Saturation. On the ground, such points that correspond to an object with a single color have different lightness (darkness) of colors depending on how the light hits the object. The HLS color space represents the difference in color according to how the light hits as the value of Lightness component. The original color of the object gathers on the axis defined by the Hue and Saturation components. Therefore, the difference in colors with objects appears more clearly than in red, green, and blue space, whose components are based on three primary colors of light. In our method, clustering is performed by the original color of the object. We apply the k-means++[17] method to the plane composed of the two components of Hue and Saturation. The k-means++ method is needed to select the number of clusters to divide points beforehand. However, it is difficult to manually set this for each cluster. Our method uses gap statistics[18] to automatically select the number of clusters.

After the color-based clustering, our method applies interpoint distance-based clustering to points in each color segment. This is because a cluster generated by the color-based clustering does not always correspond to a single object. For example, let us assume that several trees are adjacent to a house in different directions. The color-based clustering distinguishes the house and the trees but may not be able to separate the multiple trees. Our method uses the interpoint distance to distinguish these objects.

3.5 | Cluster reintegration

The above clustering may divide a point set that corresponds to a single object into several clusters. Therefore, it is necessary to "repair" such segmented clusters to find the correct correspondence between an object and a cluster.

Firstly, our method merges clusters divided by color-based clustering and then merges clusters divided by clustering based on interpoint distance.

3.5.1 | Reintegration of color-separated clusters

The reason that such a situation occurs is that an object usually consists of multiple colors. For example, a house has different colors for the roof, walls, and door. Therefore, it is basically natural, but we would like to give semantic to the set of clusters as "a house".

Most of the outside objects (eg, trees and buildings) are on the ground, and accordingly, it is very rare that different objects spatially overlap. Therefore, our approach makes use of the degree

of overlap of clusters in the horizontal plane and merges the clusters with a certain overlapped space.

Technically, we use 2D convex hull to represent the region of a point cluster. S_A is the area of the convex hull of the *xy* region of the point cluster P_A. The following formula determines the overlap ratio of two point clusters P_A and P_B

$$R_{(A,B)} = \frac{S_{A\cap B}}{\min(S_A, S_B)}, \tag{6}$$

where $S_{A\cap B}$ represents the area of the overlap region of the two convex hulls. $R_{(A,B)}$ indicates how much the smaller cluster overlaps with the counterpart. When $R_{(A,B)}$ exceeds a certain threshold (about 0.4), P_A and P_B are regarded as those coming from the same object and they should be integrated into single one. We apply this for each pair of clusters, which were regarded as a single cluster before the color-based clustering but as separated ones after it, to recover the original distance-based cluster information.

3.5.2 | Reintegration of other isolated clusters

The point density of point clouds generated from aerial images is often much lower than that by LiDAR. This is very natural as LiDAR emits infrared beams with high frequency to different directions. Due to the cost of LiDAR hardware itself and the cost of obtaining super-accurate positions and directions of the LiDAR on the flying drone at every moment, LiDAR-based geo-survey is quite expensive. On the contrary, image-based survey is cost efficient but low-density data should be handled. Such low-density feature may separate points representing a single object into multiple clusters by the previous procedure in Section 3.2. Particularly, the distribution of points in blind spots is very sparse, which makes matter worse. For example, in hardwood trees, aerial images often catch only the upper leaves of the tree and the root, and points are divided into two clusters due to the occlusion by the leaves. These multiple clusters need to be integrated into a single cluster.

The conditions for the reintegration are based on the same principle as that of Section 3.5.1. A cluster is compared with all clusters and judged whether to merge each cluster or not. We set the threshold to about 0.9 based on our experience.

3.6 | Cluster filtering for small object removal

Point clusters become larger after the above reintegration. Large clusters corresponding to large objects such as buildings and forests are useful in the applications and services explained in the introduction, whereas small clusters (such as shrubberies) are not informative. Therefore, we remove such small clusters.

The filtering method is the same as that in Section 3.3. To leave the large objects, we set the filtering threshold to about 1.0-3.0 m, which is larger than that in Section 3.3. We may use appropriate values according to the application purposes.

4 | EVALUATION

We have examined our method in the following four case studies. In the first study, the object is a real vehicle on the flat ground and images are taken by our drone. We observed the accuracy of ground surface removal. The second one targets a house located in a residential area. The images are from ESRI. We observed the effect of clustering, reintegration, and filtering. The third

FIGURE 5 Vehicle on the ground. A, One of the images; B, Input point clouds; C, After removing the ground surface; D, After applying our method [Color figure can be viewed at wileyonlinelibrary.com]

study also targets houses but the geography is modeled by plastic models and more Japanese scenes are assumed. We observed the effect of the procedure after clustering. We also prepared the point clouds for collapse of houses assuming disasters such as severe earthquake. The fourth study targets a few buildings but larger than those houses in the previous cases. The most significant characteristic is the buildings are in the forest, which is the severest case among the four. We verified the result of ground surface removal under undulating situations. We also compared the results in such undulating situations with those in the previous two cases where the ground is mostly flat. We used OpenDroneMap[13] to generate point clouds. We removed outlier points scattered at positions irrelevant to the object beforehand. We used Point Cloud Library[19] for the processing of point clouds.

4.1 | Vehicle standing on flat ground

We applied the ground surface removal processing (Section 3.1) to a vehicle on a flat ground and evaluated the accuracy. We placed a vehicle on the ground at a drone airfield in Nose Town, Osaka Prefecture and photographed this using a drone. The point cloud is made from 14 images. Figure 5A shows one snapshot. Figure 5B shows the point cloud generated from these images.

Figure 5C shows the results after the ground surface removal algorithm. Most of the ground surface is cleanly removed, whereas a few points surrounded by yellow line is not removed. The reason is that the points corresponding to snow are slightly above the ground surface. Figure 5D shows the result of applying the procedures in Sections 3.2-3.6 to the point cloud after ground removal. The residue in Figure 5C is removed by filtering, and only the vehicle is identified as one cluster. We confirmed that the filtering procedures successfully remove some remaining points after the ground surface removal.

Table 1 shows the result of comparison of our method with "manual work" regarding the height of points that correspond to the vehicle after the ground surface removal. The physical height of the vehicle is 160.0 cm, which is shown in column (A). The manual work means that we have manually removed such points that correspond to the ground, using the original photo and the colors of those points. We employ this manual work as careful manual processing of points is only the way of detecting the height of the vehicle in the original point cloud, which contains some error from the physical height. The result of manual work is in column (B), where the height is 173.3 cm. Then, the height estimated by our method is 153.7 cm as shown in column (C). We believe that our automatic recognition could recognize the vehicle with acceptable height error in practice as about 11% height does not affect greatly our target scenario where we distinguish

TABLE 1 Vehicle height estimation result

(A) Physical height	(B) Height in point cloud	(C) Height by our method	$\frac{(B)-(C)}{(B)}$ (%)
160.0 cm	173.3 cm	153.7 cm	11.3%

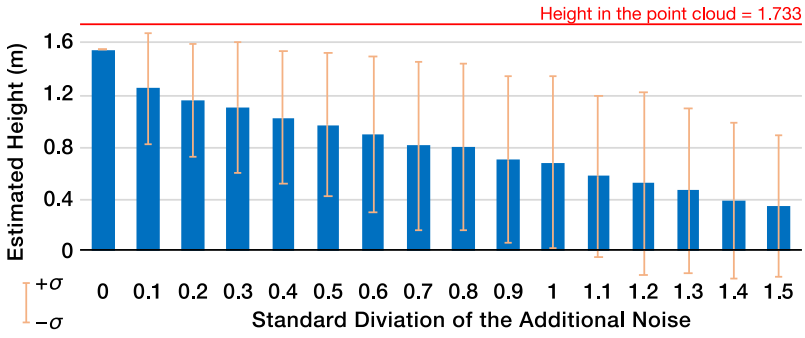

FIGURE 6 *Z*-width of vehicle clusters vs. standard deviation of noise [Color figure can be viewed at wileyonlinelibrary.com]

(A) (B) (C)

FIGURE 7 Buildings on the flat ground. A, Target building; B, Input point cloud; C, After applying our method [Color figure can be viewed at wileyonlinelibrary.com]

artificial and natural objects on the ground surface. For a more critical scenario where a severe earthquake attacks and houses collapse we need to compare the situations before and after the disaster, to quickly recognize how severe the damages of houses are for more efficient rescue operations and so on. We will discuss this scenario and conducted an experiment in Section 4.3.

Next, we confirmed the change in cluster height when we gave noise to the elevation data. In this way, we can measure the influence of elevation data accuracy on our method. Figure 6 shows the height of the vehicle cluster when giving errors with different standard deviations for the elevation data. The given errors are normally distributed with an average of 0. The reduction criterion is the *z*-width when manually removing the ground. The value is the average of the results obtained in 60 trials.

The graph shows that the larger the standard deviation of the error given to the elevation data, the more nonground points are removed. If the standard deviation of the error is 0.9 m or more, the average value of the height reduction of the object exceeds 1 m. The elevation data itself contains an error, whose standard deviation is about 0.3 m. Therefore, when the standard deviation of the total error is over $\sqrt{0.3^2 + 0.9^2}$ = about 0.94 m, the average reduction exceeds 1 m. We note that it may often be difficult to derive the accurate height of an object using elevation data with large errors. However, even in such a case, our method is useful for capturing changes in the height of an object (eg, Section 4.3). This is because the same object has always the same error of the estimated height as long as we use the same elevation data, and accordingly, the difference in the estimated height does not largely depend on the elevation error.

4.2 | A house in residential area on flat ground

We confirmed the effectiveness of our method for a house in a residential area of a flatland. Figure 7A shows the target house. We used 36 aerial images to generate the point cloud. The

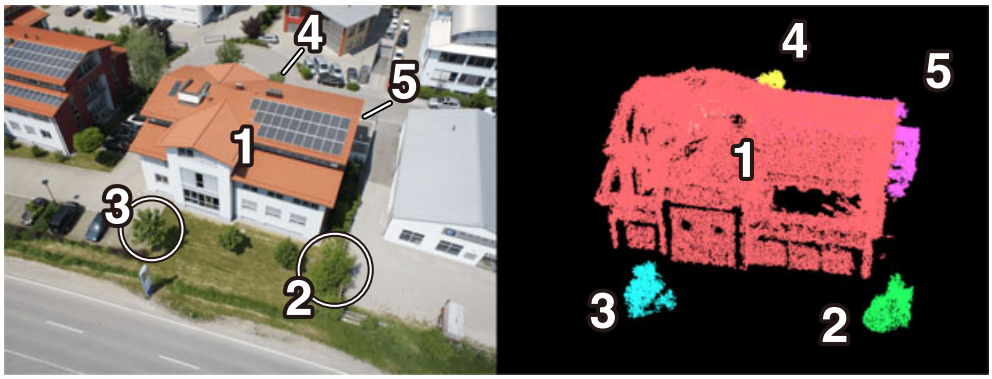

FIGURE 8 The correspondence between actual objects and clusters [Color figure can be viewed at wileyonlinelibrary.com]

images capture the buildings standing on the outskirts of Munich, Germany.[20] In Germany, we can obtain the numerical elevation model, which has 1-5 m errors from the Federal Agency for Cartography and Geodesy,[21] which is insufficient to apply the ground surface removal. Therefore, we did not apply it but applied the plane extraction algorithm by RANSAC. The surrounding points were removed manually. Figure 7B shows the input point cloud. We verify the result of applying the process described in Sections 3.2-3.6.

Figure 7C shows the result of applying our method. Objects on the ground are correctly clustered, including the house and large trees. Figure 8 shows the correspondence between actual objects and clusters. We compared the point cloud after applying the method and the point cloud before reintegration for interpoint distance-based clustering in Section 3.5.1. An asterisk (∗) is added to Figure 9A, which represents a cluster independently clustered from the main cluster of the building. Such clusters are reintegrated with other clusters representing the same building later. Comparison between Figure 9A and Figure 7C shows that the reintegration works well. Furthermore, filtering after reintegration removes small clusters representing small planting, vehicles and so on. We also compared the difference in the results of whether or not color-based clustering was used. Figure 9B shows the result of applying our method without color-based clustering. Interpoint distance-based clustering does not divide objects adjacent to the house, including the

FIGURE 9 Comparison of the results. A, Before reintegration for clustering for object isolation; B, Without clustering based on color [Color figure can be viewed at wileyonlinelibrary.com]

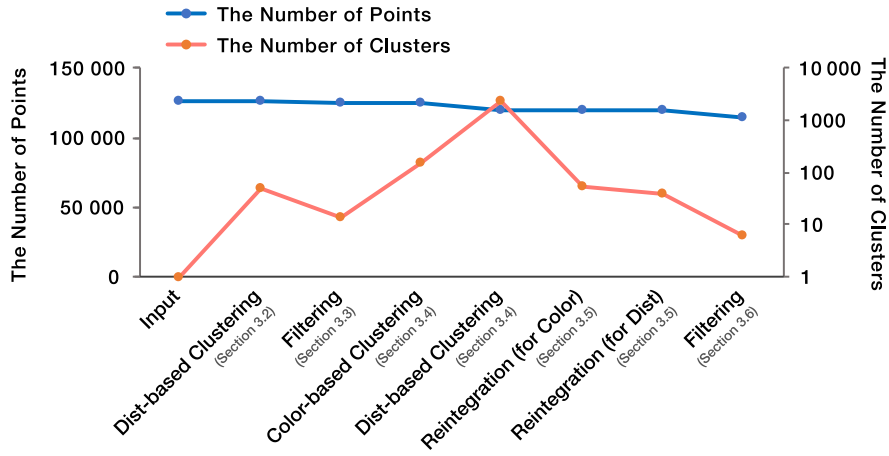

FIGURE 10 Transitions in the number of points and the number of clusters [Color figure can be viewed at wileyonlinelibrary.com]

trees behind it. The result shows color-based clustering contributes to the division of adjacent objects.

Figure 10 shows the number of points and clusters in the point cloud at the point of passing through the process in Sections 3.2-3.6. The noise filtering decreases the number of points from 125 874 to 125 487, which is only about 0.3%, while decreases the number of clusters from 51 to 14, which is about 72.5%. This shows that the noise filtering greatly reduces the number of clusters that affects a load of reintegration without affecting the shape of the point cloud.

4.3 | Residential areas model before/after disaster

We assume our method is used for a simple survey of damage caused by a disaster. We examined the result of applying the method to the point clouds capturing a residential area before and after a disaster. Since it is definitely difficult to prepare point clouds before and after a disaster, we arranged plastic models of 1/150 scale simulating a residential area before and after the collapse of a house, and we took 23 images from various angles. The point clouds were made from the images. We applied the enlargement process to the point clouds so as to obtain an actual scale. There is no elevation data of the model, and the ground surface removal in the method cannot be applied to the point clouds. Therefore, we also used RANSAC to remove the ground surface as in Section 4.2. We examined the method except for the ground surface removal. Figure 11A and Figure 11B show the model simulating the predisaster situation and Figure 11C and Figure 11D shows the model simulating the postdisaster situation. We assumed the case of an earthquake as a disaster and collapse the house surrounded by a red line in Figure 11. We compared the cluster changes of the house before and after the disaster with other houses, which were not damaged.

Figure 12 shows the result of applying the method to each of two point clouds. The assigned numbers refer to the same cluster in Figure 12A and Figure 12B. Barn and fence at No.7 are grouped in the same cluster. This is because the distance between both clusters is too short and both are similar in color.

We examined the clusters corresponding to houses before and after the disaster. In the model, the house of No.4 collapses after the disaster and the roof is on the ground, whereas other houses have no change in the shape. The height of a building can be obtained from the z-width in a

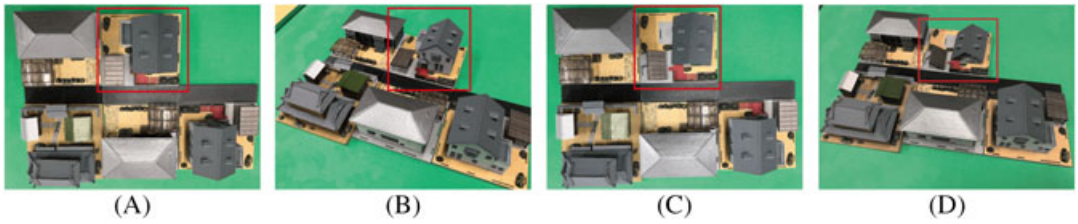

FIGURE 11 Model of a residential area simulating the situation before and after the disaster. A, Before the disaster (from above); B, Before the disaster (from diagonally above); C, After the disaster (from above); D, After the disaster (from diagonally above) [Color figure can be viewed at wileyonlinelibrary.com]

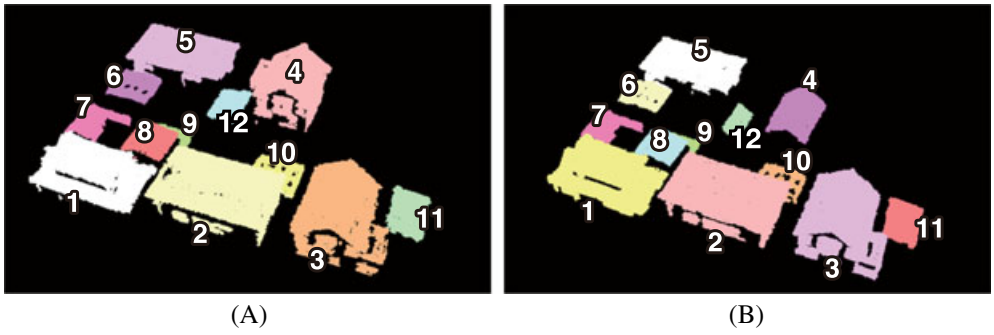

FIGURE 12 Results of applying our method. A, Predisaster situations; B, Postdisaster situations [Color figure can be viewed at wileyonlinelibrary.com]

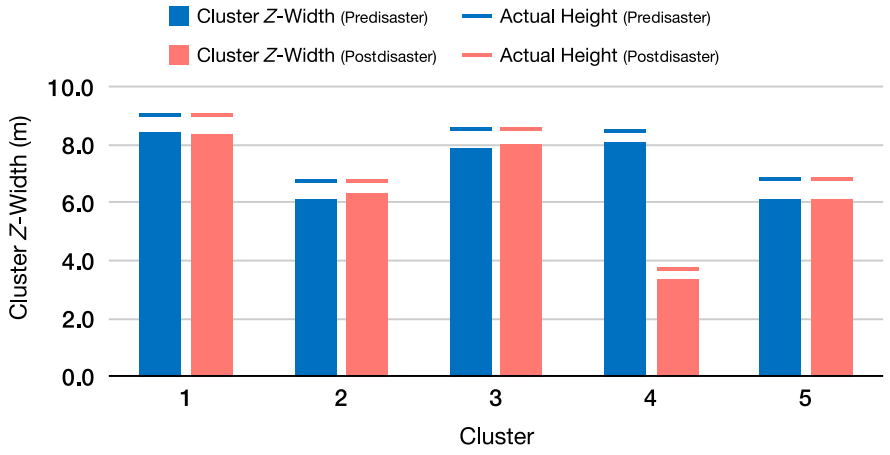

FIGURE 13 Results of applying our method [Color figure can be viewed at wileyonlinelibrary.com]

cluster. Figure 13 summarizes the height of clusters (No.1-No.5) before and after the disaster and the amount of change. The z-width is lower by about 50 cm than the actual height of the model. This is because the RANSAC algorithm considers a part of the building as the ground surface and can remove it. We note that if point clouds capturing the same object are made from different images, those point clouds have different point distributions. This is the reason why the z-width of a cluster is different between point clouds. In fact, the clusters except No.4 have undergone a height change of up to about ±3.5% despite no change after the disaster. On the other hand,

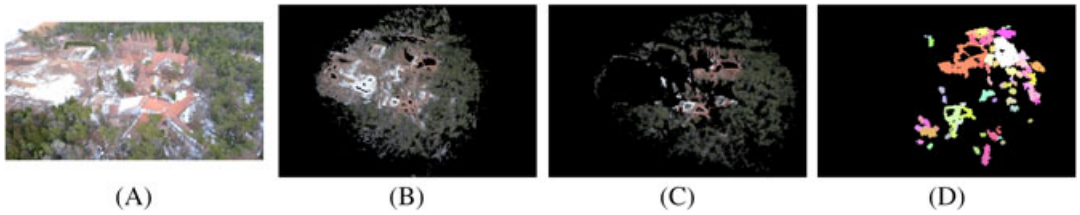

FIGURE 14 The forest and buildings. A, A forest in Nose Town, Osaka Pref; B, Input point cloud; C, After removing the ground surface; D, After applying our method [Color figure can be viewed at wileyonlinelibrary.com]

the change rate in the height of No.4, which suffered great damage, reaches −58.7%. The rate is much higher than in the other clusters. The comparison shows the method can contribute to the detection of the houses damaged to a certain extent.

4.4 | Buildings in the forest

Finally, we examined the results of applying our method to a point cloud capturing forests and buildings on the undulating ground surface. We used our drone and took photos of a forest in Nose Town, Osaka Prefecture. Figure 14A is one of the 49 images used for generating a point cloud. Figure 14B shows the point cloud from these images.

Figure 14C shows the result of the ground surface removal in Section 3.1. The place is down from the left side to the right side. Even in such a situation, the ground surface is correctly removed. The points at the left end of Figure 14C are not residues of the ground surface but the trees surrounding the athletic field. Figure 14D shows the result of applying a series of processing. In the central part of the point cloud, the points form the clusters according to the trees, whereas most of the points outside the image disappear. Generally, more images capture a certain area, more points distribute in the area. The center part is captured more times than the outside, and accordingly, the points in the center part are denser than the outside. Therefore, when we set the threshold suitable for the center part, clusters are formed poorly on the outside due to low density, and they are excluded in the method. The result shows that point clouds capturing a wide area cannot be used equally in a whole area. Figure 15 shows the result of segmentation of a building shown in the back of Figure 14A. The cluster has a shape that can be distinguished as a building, but the trees adjacent to the building are erroneously extracted in the part surrounded by the yellow line. The method merges multiple clusters based on the degree of overlap of convex hulls formed from distribution areas (Section 3.5). The target building has a complicated shape, and the range of the convex hull (surrounded by the blue line) becomes larger than the actual distribution area, and the trees in the convex hull are merged. If the shape of object is apart from the convex hull, assimilation of several objects occurs frequently.

Both the division of trees and that of buildings is lower in accuracy than the verification in the Sections 4.1-4.3. This is because vegetation in the forest is too dense to clarify the spatial separation between objects, which is the premise of the method. However, the clusters can be distinguished between buildings and trees by visual observation of the cluster shape. The method contributes to the improvement of efficiency in grasping the distribution situation of trees, the arrangement and size of buildings, and so on. In particular, the method is useful for such usage where a precise 3D shape is not required. An example of such an application is geometry creation for wireless network simulation in real field.

(A) (B)

FIGURE 15 Comparison of a real building and a cluster. A, Aerial Image; B, Cluster [Color figure can be viewed at wileyonlinelibrary.com]

5 | LIMITATION

In this section, we discuss the limitations of our method.

5.1 | The terrain is significantly different from the elevation data

In our method, the ground removal uses elevation data as a true value. When the data and the real topography are significantly different, the method extracts the points different from the ground surface as the ground surface. For example, the error may occur in the case that the ground surface is changed after elevation data measured (eg, landslides, snow cover, etc.) or the elevation data has a too large error.

A simple survey of damage caused by a disaster is an application example of our method. If the large landslide is caused by a severe disaster and the shape of the ground surface may change, the error from the elevation data may increase. Accordingly, some ground surface may not be removed correctly.

Our method outputs those points that are not removed as objects although these points are from the ground surface. This means that by checking each object, we can detect the change of the ground surface and the occurrence of the landslide, etc, as these changes appear as objects in the point cloud.

5.2 | Part of the objects is missing in point cloud

The reintegration in our method is based on the premise that an entire object is in point clouds. If a part of the object is not reproduced as a point cloud due to a limitation of views from drones, the space with objects exists may not match that of the clusters. This naturally causes the errors in reintegration.

5.3 | The same colored objects are adjacent

Our design is based on the hypothesis that most of the objects are spatially separated or different in color. Therefore, the condition of clustering depends only on interpoint distance and the color

difference. Straightforwardly, our method is not able to clearly separate adjacent objects with the same color, eg, trees in the forest.

6 | CONCLUSION

This paper has presented a ground object recognition method for 3D point clouds. The research is aimed at contributing to the utilization of (noise prone) 3D point clouds generated from aerial images captured by a drone. Low-cost feature is significant for many applications and services that wish to make use of such information, compared with LiDAR-based high-cost accurate point cloud generation. The method consists of following processes: the removal of the ground surface, the interpoint distance-based clustering, filtering for noise removal, color-based clustering, integrating clusters representing the same object, and filtering according to the purpose of the output.

We applied the method to the four situations and verified the effectiveness of the method from several viewpoints. Firstly, we have examined the ground surface removal accuracy on the plane ground with a single object using images taken by a drone from different angles, and the error of the estimated object height after ground surface removal was less than 20 cm. Secondly, we have verified the building recognition capability, and points that correspond to the building was correctly identified even with some plants such as trees around the building. In addition, we have confirmed that the clustering and reintegration process contribute to improving accuracy. Thirdly, assuming a disaster scenario, we have recognized the height difference before and after the building collapse. Changes in the height of a house appeared as changes in the z-width of the cluster, which indicated the possibility of detecting the collapse of a house. Finally, we have applied our method to recognize a single-story building as well as a number of trees in a mountainous area, which was the more challenging case among the four. We have confirmed that the ground surface was mostly removed even under undulating situations, and most parts of the building were accurately identified.

ACKNOWLEDGMENTS

This work was supported by the Empirical Research and Development for Solving Regional Issues by Data Utilization (No. 200) and the Commissioned Research of the National Institute of Information and Communications Technology (NICT), Japan.

ORCID

Katsuya Ogura https://orcid.org/0000-0002-1575-1216

REFERENCES

1. Agarwal S, Furukawa Y, Snavely N, et al. Building Rome in a day. *Commun ACM*. 2011;54(10):105-112.
2. Ogura K, Yamada Y, Kajita S, Yamaguchi H, Higashino T, Takai M. Ground object recognition from aerial image-based 3D point cloud. In: Proceedings of 2018 11th International Conference on Mobile Computing and Ubiquitous Network (ICMU); 2018; Auckland, New Zealand.
3. Arifin F, Daniel RA, Widiyanto D. Autonomous detection and tracking of an object autonomously using ar.drone quadcopter. *J Ilmu Komputer dan Informasi*. 2014;7(1):11-17.
4. Chen P, Dang Y, Liang R, Zhu W, He X. Real-time object tracking on a drone with multi-inertial sensing data. *IEEE Trans Intell Transport Syst*. 2018;19(1):131-139.

5. Yasuoka R, Sonogashira M, Iiyama M. An object recognition with photographs that was taken by a drone' camera. *ELCAS Journal*. 2018;3:85-87.
6. van Gemert JC, Verschoor CR, Mettes P, Epema K, Koh LP, Wich S. Nature conservation drones for automatic localization and counting of animals. In: Proceedings of European Conference on Computer Vision (ECCV) Workshops; 2014; Zurich, Switzerland.
7. Maria G, Baccaglini E, Brevi D, Gavelli M, Scopigno R. A drone-based image processing system for car detection in a smart transport infrastructure. In: Proceedings of 2016 18th Mediterranean Electrotechnical Conference (MELECON); 2016; Lemesos, Cyprus.
8. Fischler MA, Bolles RC. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun ACM*. 1981;24(6):381-395.
9. Schnabel R, Wahl R, Klein R. Efficient RANSAC for point-cloud shape detection. *Comput Graph Forum*. 2007;26(2):214-226.
10. Zhang K, Chen S-C, Whitman D, Shyu M-L, Yan J, Zhang C. A progressive morphological filter for removing nonground measurements from airborne LIDAR data. *IEEE Trans Geosci Remote Sens*. 2003;41(4):872-882.
11. Golovinskiy A, Funkhouser T. Min-cut based segmentation of point clouds. In: Proceedings of 2009 IEEE 12th International Conference on Computer Vision (ICCV) Workshops; 2009; Kyoto, Japan.
12. Ramiya AM, Nidamanuri RR, Krishnan R. Object-oriented semantic labelling of spectral-spatial LiDAR point cloud for urban land cover classification and buildings detection. *Geocarto International*. 2015;31(2):121-139.
13. OpenDroneMap. Drone mapping software. OpenDroneMap. http://opendronemap.org/
14. Japan Geospatial Information Authority. Fundamental geospatial data download service. https://fgd.gsi.go.jp/download/menu.php
15. Rusu RB. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments* [PhD Thesis]. Munich, Germany: Computer Science Department, Technische Universität; 2009.
16. Kirkpatrick DG, Seidel R. The ultimate planar convex hull algorithm? *SIAM J Comput*. 1986;15(1):287-299.
17. Arthur D, Vassilvitskii S. k-means++: the advantages of careful seeding. In: Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA); 2007; New Orleans, LA.
18. Tibshirani R, Walther G, Hastie T. Estimating the number of clusters in a data set via the gap statistic. *J R Stat Soc Ser B Stat Methodol*. 2001;63(2):411-423.
19. Rusu RB, Cousins S. 3D is here: Point Cloud Library (PCL). In: Proceedings of 2011 IEEE International Conference on Robotics and Automation (ICRA); 2011; Shanghai, China.
20. Esri. Sample data - Drone2Map for ArcGIS help. http://doc.arcgis.com/en/drone2map/get-started/sample-data.htm
21. Federal Agency for Cartography and Geodesy. Open data - free data and services of BKG. http://www.geodatenzentrum.de/geodaten/gdz?l=down_opendata

How to cite this article: Ogura K, Yamada Y, Kajita S, Yamaguchi H, Higashino T, Takai M. Ground object recognition and segmentation from aerial image-based 3D point cloud. *Computational Intelligence*. 2019;35:625–642. https://doi.org/10.1111/coin.12232