A New Method of Achieving Single Three-Dimensional Building Model Automatically Based on Oblique Photography Data

Defu Che,1,2 Zonghui Li,1,2 Yining Liu,1,2 Renqing Zhong,1,2 and Baodong Ma1,2

1Key Laboratory of Ministry of Education on Safe Mining of Deep Metal Mines, Northeastern University, Shenyang 110819, China
2Institute for Geoinformatics & Digital Mine Research, Northeastern University, Shenyang 110819, China

Correspondence should be addressed to Defu Che; chedefuneu@163.com

Received 25 January 2021; Accepted 4 September 2021; Published 17 September 2021

Academic Editor: Florin Stoican

Copyright © 2021 Defu Che et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Operating and managing single three-dimensional building model individually are critical in the application of oblique photography models. However, these models are usually complete and continuous, and the single three-dimensional building model in these models cannot be managed individually. Generally, achieving a single three-dimensional building model requires a human-computer interaction to determine the cutting range, but this process is time-consuming and inefficient. To overcome this problem, this study proposed a new method for automatically achieving single three-dimensional building model without the need for human-computer interaction. First, the point clouds of an oblique photography model are divided into virtual grids, and the point clouds in each virtual grid are seen as a whole. In this way, the number of point clouds involved in the calculation is reduced, thereby improving computing efficiency. Second, the point clouds of a building facade are extracted by setting the height difference. By comparing the height difference between the highest point and the lowest point of the grid with height difference threshold, all point clouds in the grid that do not meet the requirements are eliminated. Third, the point clouds of the building facade are classified, and the contour line is extracted by the classified point clouds. Finally, the single three-dimensional building model is achieved by reconstructing the triangles that intersect with the extracted contour line. Experimental results show that the proposed method can effectively achieve single bodies automatically from an oblique photography building model. This method is then useful for achieving single three-dimensional building model from massive oblique photography data.

1. Introduction

With the increasing application of 3D Geographic Information System (GIS) [1, 2], dealing with problems in the 3D space is considered more intuitive and efficient than those in the 2D space. 3D building models are important parts of 3D GIS, which are widely used in city map of smart city and Virtual campus [3]. There are some ways to obtain 3D building models. One common method is to use LIDAR to collect point cloud data and then generate a 3D model. Although the method is effective, the cost is high because the equipment is expensive and the data processing is complicated. The development of oblique photography technology has enabled a low-cost and high-efficiency 3D reconstruction of real scenes through multiview aerial images [4]. Oblique photography is a new technology in the field of surveying and mapping where cameras are installed obliquely on a platform to obtain information about the ground features, especially the facade of buildings. Oblique photography has been applied in various fields, such as location services [5], disaster assessment [6], and archaeological research [7]. However, the original model from oblique photography is a continuous triangular net, and buildings and other features cannot be managed separately [8]. Therefore, achieving single 3D building model is critical in the application of the oblique photography model in 3D GIS.

Single 3D building model can be achieved in three ways [9]. First, an ID is added to the vertices of triangles, and the single bodies can then be identified by different IDs. Second,
dynamic achieving superimposes the corresponding vector on a single body, and the single body model in 3D rendering is identified by the vector face. These two methods do not need to obtain the contour line information of the building model but cannot achieve an actual segmentation of continuous triangles. Third, single bodies are achieved by cutting the complete and continuous one-scene model [10, 11]. This method needs to obtain the contour information of the building model and then reconstructs the triangulation network according to the intersection relationship between the contour information and the triangle [12]. After cutting, the objects in the oblique photography are separated.

To cut the building model, the cutting range or contour line of the building needs to be obtained. Currently, the contour line is acquired via human–computer interaction. However, such method is time-consuming and may lead to certain errors that affect its application. Building contour information serves as the basis for building extraction and 3D model reconstruction, and it has been widely used in various applications, such as urban basic information database update [13], target recognition [14], disaster prediction [15], and other fields [16, 17]. In addition to manual drawing, the contour line of the building can also be automatically extracted. Building contour information is mainly extracted in two ways. On the one hand, image-based building contour extraction [18] applies geometric boundary corner feature extraction and reconstruction methods to extract building outlines from high-resolution remote sensing images. However, it is difficult to extract accurate boundary from only aerial image partially due to shadows, occlusions, and poor contrast [19]. The process is complex and time-consuming. On the other hand, point-cloud-data-based methods [20, 21] use clustering to extract the point cloud of a building and then construct the building contour line using Alpha Shapes [22, 23]. In addition to building point cloud, the collected point cloud data usually includes various types of ground objects, such as vegetation, ground, etc. In the process of extracting contour lines, other point clouds such as vegetation will greatly interfere with the results. So, first of all, the point cloud data of tree and ground will be removed [24]. Afterward, the triangle inside the obtained contour line is separated from the original model, and a new model is constructed. The triangles intersecting with the contour line are cut and subsequently classified. The triangles outside the contour line are added to the original model, whereas those inside the contour line are added to the newly generated single model. In this way, a single body can be achieved by cutting.

As shown in Figure 1, the oblique photography model is a continuous triangular mesh. It is impossible to detect the triangular network of the building model separately. There are some studies that use vector files imported from outside or polygon by human–computer interaction to obtain single 3D building model of the oblique photography model [25]. Although its cutting effect is acceptable, this method relies on the baseline files imported from outside and does not realize the achieving single three-dimensional building model of the oblique photography model automatically.

There are also some researches that used LIDAR point cloud data to extract the outline information automatically of buildings to represent the building model [26]. However, the final result is only to obtain the 2D information of buildings, not to obtain the 3D model of a single building [27]. Thus, 3D building models cannot be managed separately. Furthermore, the data source used in these studies is LIDAR data, which has high cost in actual production [28]. Given the large number of vertices in the oblique photography model, vertex data can be extracted as point cloud data. A large amount of discrete point cloud data can be extracted in this process. These point cloud data are used to construct a virtual grid to reduce the amount and to achieve an even distribution of these data, thus improving computing efficiency. The difference in elevation between buildings can also be used to filter the point cloud of buildings, remove most point clouds that are not intended to improve accuracy, and use the clustering algorithm to achieve point cloud clustering [29]. The building contour can then be extracted through point cloud data [30]. Therefore, based on the characteristics of oblique photography model data, a method for automatically achieving single bodies from a 3D building model is proposed in this study. This method avoids the complicated process of manually drawing the range, and the accuracy of the extracted contour line is high. Furthermore, the final result is to obtain the 3D information of buildings, and 3D building models can be managed separately.

2. Data and Method

2.1. Data Acquisition and Presentation. The oblique photography model data of Northeastern University, which were collected by an unmanned aerial vehicle, are used in this study. Consistent with the content of this study, the area covered by these data includes multiple building models. In the data acquisition process, the UAV carries five lenses, and the sensor size is 23.2 mm × 15.4 mm. The 3D model format is generated using the ContextCapture Center software, and the final model is proven to have good quality. The data in the 3D model are formatted in OSGB, which is a common 3D model data format. This model has its own multilayer LOD and demonstrates natural advantages in browsing and loading large scenes. The OSGB model file stores all types of...
information, such as vertex coordinates, texture coordinates, and texture pictures of the model. The vertex coordinates and triangulation data of the model are easy to extract, thereby facilitating subsequent model processing.

2.2. Flowchart. The proposed method is divided into two parts, namely, extracting the building contour and achieving single bodies by cutting a triangular net. Figure 2 illustrates the entire process. To extract the building contour, the point cloud data extracted from the oblique photography model are divided into virtual grids with a fixed size. Afterward, the height difference of \( t \) points in each grid is calculated, and both ground and vegetation are removed by setting a height difference. The point cloud data are then clustered, and the contour line is drawn by the cluster point cloud data. To achieve a single body from an oblique photography model, the cutting polygon is determined by the constructed building contour, the bounding box of the cutting polygon is intersected with bounding box of the first-level LOD [31] of the oblique photography model, and the intersecting models are extracted. The triangles of the intersecting model are divided into three parts, namely, inside, outside, and intersecting with the cutting polygon. The intersecting triangle is then cut, the resulting triangle is divided into inside and outside, the inner triangle is combined into a single model, and the outer model is modified to the original model file. All LODs of the oblique photography model are traversed to achieve a single body.

2.3. Extracting Building Contours Based on an Oblique Photography Model. The vertex information extracted from the original data includes information related to the ground, trees, and buildings, among others. The distribution of the extracted point cloud data differs from that of the point cloud data scanned by a laser radar. To consider the model features and reduce the amount of data in the production process of the oblique photography model, the point cloud is dense in areas where the features are obvious yet sparse in those areas where the features are not obvious. Therefore, the uneven distribution and density of point cloud data make the laser-point-cloud-based building contour extraction method very unsuitable for this work.

2.3.1. Virtual Grid Generation. The point cloud is denser where the features are obvious, and the point cloud is sparse where the features are not obvious. Unevenly distributed point clouds will affect the extraction of contour lines. To improve the calculation efficiency and accuracy of the drawing contour, the neighborhood relationship of the point cloud should be constructed. In this study, a virtual grid [32] is constructed from point clouds to facilitate the subsequent analysis and processing. A virtual grid is then constructed as follows.

Calculate the bounding box of the input point cloud to obtain the minimum \((x_{\text{min}}, y_{\text{min}}, z_{\text{min}})\) and maximum point coordinates \((x_{\text{max}}, y_{\text{max}}, z_{\text{max}})\) of the point cloud area. The number of rows \(r\) and columns \(c\) of the entire grid is calculated as

\[
\begin{align*}
  r &= \frac{x_{\text{max}} - x_{\text{min}}}{w}, \\
  c &= \frac{y_{\text{max}} - y_{\text{min}}}{w},
\end{align*}
\]

where \(w\) is the size of the virtual grid that needs to take into account the point cloud density. The row and column number of the virtual grid at any point \((x_i, y_i)\) can be calculated as

\[
\begin{align*}
  r_i &= \frac{x_i - x_{\text{min}}}{w}, \\
  c_i &= \frac{y_i - y_{\text{min}}}{w},
\end{align*}
\]

where \((x_i, y_i)\) is the plane coordinate of point \(i\).

All point cloud data are eventually divided into several grids of the same size. The point cloud data contained in each grid are represented by the central point cloud of the grid. In this way, the problems related to the large amount of point cloud data and their uneven distribution can be solved.

2.3.2. Point Cloud Extraction of Building Façade Based on Height Difference. The largest difference between buildings and the ground is reflected in elevation. In the oblique photography model, the elevation change between adjacent points is not obvious. However, the building has walls, and the elevation obviously changes at the edge of the building. Therefore, most of the plane point clouds can be eliminated by setting a height difference threshold. These point clouds include ground points and the top surface of buildings and the most of the vegetation points. The extraction process can be roughly divided into several steps. First, the height difference threshold is set, all grids are traversed, and the point clouds whose height difference is below this threshold are removed from the grid. Second, the retained point clouds are mostly the façade points of buildings, and few of these points are residual point clouds from tall trees.

2.3.3. Point Cloud Classification and Contour Generation. Many algorithms, such as DBSCAN [33, 34] and K-means [35, 36], are used for point cloud classification. This study adopted the density-based DBSCAN clustering method, which divides regions with sufficiently high density into several clusters. Unlike K-means, this method does not need to know the number of clusters in advance and can identify noise points [37]. Afterward, the Alpha Shapes [38] algorithm is used to construct the contours of each building from the classified point cloud.

2.4. Achieving a Single Body by Cutting Triangles

2.4.1. Cross Detection and Classification of Triangular Face. An oblique photography 3D model has a large number of triangles. If each of these triangles intersects with the cut polygon formed by the building contour, then the processing
efficiency will be greatly affected [39]. The following processing flow is therefore adopted. First, the intersection relationship between the bounding box of the polygon and the bounding box of the model is determined. Second, the model whose bounding box intersects with that of the cut polygon is identified. Third, AABB is applied to detect the intersection of triangular surfaces. Fourth, these triangular surfaces are divided into three categories, namely, inside the area, outside the area, and intersecting. Finally, the intersecting triangular facets are cut, and the results are divided into two categories, namely, inside and outside the ranges. The triangles outside the area are removed, and those inside the area are preserved as a single body model.

2.4.2. Cutting Triangulation Net. The relationship between the cutting line and the triangulation net can be divided into three categories, namely, the triangle is tangent to the cutting line, the triangle intersects with a cutting line, and the triangle intersects with two cutting lines. Figure 3 presents four cases of intersections and tangencies between a triangle and a cutting line. First, a cutting line intersects with two sides of the triangle. Second, a cutting line intersects with one vertex and one side of the triangle. Third, a cutting line is tangent to one side of the triangle. Fourth, a cutting line is tangent to a point of the triangle. Meanwhile, Figure 4 presents four cases of intersections between a triangle and two cutting lines. First, a vertex of the cut polygon is located inside the triangle, and two sides intersect with two points of the triangle. Second, a vertex of the cut polygon is located inside the triangle, and two sides intersect with one side of the triangle. Third, a vertex of the cut polygon is located inside the triangle, a vertex of the triangle is located inside the cut polygon, and two sides intersect with two sides of the triangle. Fourth, a vertex of the cut polygon is located inside the triangle, and two sides intersect with one point and one side of the triangle.

The cutting process cuts the minimum geometric element and ensures that the geometric characteristics are still met afterward. Given that the oblique photography model used in this study comprised triangular slices, after cutting the minimum unit of the model, the topological relationship between triangular slices should still be maintained.

In Figures 3(c) and 3(d), given that the cutting line is tangent to the triangle, cutting triangles is not necessary. When a triangle is related to two lines, and one line is
tangent to the triangle and the other line interests with the triangle, this case can be seen that the triangle is cutting by one line. Different cutting methods are used for each intersecting mode. For the case in Figure 3(a) where one side of the cut polygon intersects with two sides of a triangle, the intersecting part is clipped as follows. Figure 5 shows that a cutting line cuts the triangle into two triangles and that a topological relation among the triangles is established.

For the case in Figure 3(b), one side of the cut polygon passes through the vertex of a triangle and intersects with one side of the triangle, and the intersecting part is clipped as follows. Figure 6 shows that a cutting line cuts the triangle into two triangles and that a topological relation is established among the triangles.

In Figure 4(a), one vertex of the cut polygon is located in a triangle, two cutting lines pass through two vertices of a triangle, and the intersecting part is clipped as follows. Figure 7 shows that two cutting lines cut the triangle into three triangles while maintaining their topological relation.

In Figure 4(b), a vertex of the cut polygon is located in a triangle, two cutting lines intersect one side of a triangle, and the intersecting part is clipped as follows. Figure 8 shows that two cutting lines cut the triangle into five triangles, and a topological relation is established among the triangles.

In Figure 4(c), a vertex of the cut polygon is located inside a triangle, one of two cutting lines crosses the vertex of a triangle, whereas the other intersects with one side of this triangle, and the intersecting part is clipped as follows. Figure 9 shows that two cutting lines cut the triangle into four while establishing a topological relation among these triangles.

In Figure 4(d), a vertex of the cut polygon is located in a triangle, a vertex of cutting polygon is located in the triangle, two cut lines intersect with two sides, and the intersecting part is clipped as follows. Figure 10 shows that two cutting lines cut the triangle into five and that a topological relation among these triangles is established.

2.4.3. Reconstructing the 3D Model. The reason for the reconstruction of the 3D model is that the single body by cutting triangular elements is the physical single body; this method will increase the number of vertices and triangular elements. Therefore, the data in the 3D model should be reorganized.

The intersecting triangular slices are cut according to the method described in Section 2.2.2. The intersecting points generated in this process will become new vertices of the 3D model, and a single triangular slice will be divided into multiple slices. To achieve the effect of 3D model display, the texture coordinates of the newly added vertices should be calculated. Afterward, the intersecting triangular slices are removed from the model, and the newly generated triangular slices are divided into two types, namely, outside and inside the area. All triangles inside the cut polygon are fused.
to form a single model, and fusing the external triangles into the original scene completes the removal of the single model from the original scene. The oblique photography model then achieves a single body without influencing the overall display effect.

3. Experimental Results and Discussion

To verify the effectiveness of the proposed method, this study adopted C/C++, OpenSceneGraph (OSG), and point cloud library (PCL) to carry out experiments. The original oblique
photographic data have a large number of building models, and a certain building model is selected in this study for the experiment.

3.1. Result of Building Contour Extraction. Vertex information is extracted from the original data as the initial point cloud, and then, ground and vegetation points are

Figure 8: Schematic diagram of a triangle cut by two straight lines passing through a vertex. (a) The triangle before cutting and (b) the triangle after cutting.

Figure 9: Schematic diagram of a triangle cut by one line passing through a vertex and one line without passing through a vertex. (a) The triangle before cutting and (b) the triangle after cutting.

Figure 10: Schematic diagram of a triangle cut by two lines without passing through a vertex. (a) The triangle before cutting and (b) the triangle after cutting.
eliminated by setting the virtual grid and the height difference threshold. Different grid sizes and height difference thresholds will greatly influence the extracted point cloud effect.

When the parameters are inaccurate, many wrong clustering results will be generated mainly due to the tall trees and the characteristic information of the building (Figure 11). As a rule of thumb, no intersection and inclusion are observed between the contours of separate buildings. Therefore, the clustering results affected by the characteristic information inside the building can be eliminated. The other interference clusters can be eliminated based on the shape and area of the polygon of the contour line. Figure 11(a) shows that many wrong clustering results are observed. The clustering results inside the building are eliminated by the intersection of polygons in Figure 11(b). The clustering results of tall trees can then be removed by the area and shape of the polygon in Figure 11(c).

The height difference threshold is set to 18 m, the grid size is set to 0.5, 1, and 2 m, and the extracted point cloud results are shown in Figure 12.

The effects of the point clouds extracted by 0.5 m and 1 m grids do not greatly differ. Meanwhile, the point cloud extracted by the 2 m grid is sparse, and the contour line from the sparse point cloud shows a poor effect. The extraction efficiency of the 1 m grid is higher than that of the 0.5 m grid. Therefore, the 1 m grid is adopted in this study.

The grid size is set to 1 m, the height difference threshold is set to 14, 16, and 18 m, and the extracted point cloud results are shown in Figure 13.

The above results show that the point cloud of the building has been extracted completely, but many noise points are observed when the height difference threshold is set to 14 m. These noise points will affect the extraction of contour lines. The point cloud of the building is not completely extracted when the height difference threshold is set to 18 m, which will make the extracted contour line ineffective. Meanwhile, the point cloud of the building is completely extracted, and only few noise points are observed when the height difference threshold is set to 16 m. Therefore, the height difference threshold is set to 16 m in this study. Different point cloud extraction effects are obtained by setting grid size and elevation difference as shown in Table 1.

The DBSCAN algorithm is adopted to segment the above point cloud, and Alpha Shapes is used to construct the contour lines for each point cloud cluster.

The point cloud data are clustered by setting the search range to 4 m and the minimum number of clusters to 100, 150, and 200. The effect diagrams are shown in Figure 14.

The characteristic information inside the building is also clustered when the minimum number of clusters is set to 100. Meanwhile, when the number of clusters is 200, smaller buildings are omitted, and when the minimum number of clusters is 150, the building outline is extracted correctly and an improved effect is recorded.

By combining the building contour information with the elevation difference information of each virtual grid, the building bounding box is drawn and superposed on the oblique photography 3D model. The effect is shown in Figure 15, which reveals that the bounding box can accurately contain the building model.

3.2. Effect of Achieving a Single Body Based on an Oblique Photography Model. The building contour obtained by the above method is used as a polygon to cut the oblique photography model and achieve a single body. In this study, a single building model is separated from the original scene and transformed into a child node of the original scene. Figure 16(a) shows that the building model can be managed individually, Figure 16(b) shows that this building model is independent, and Figure 16(c) shows that the edge of this model is a straight line.
4. Discussion

The measurement accuracy of oblique photography is relatively high, and the measurement result has an elevation error between 2 and 10 cm [40, 41]. Compared with the elevation of buildings, the impact of the measurement error is slight. The point cloud data of the building is distributed in multiple grids, which are connected. When the grids of the building are detected, the height difference of each grid is calculated. Finally, the average of these height differences is calculated.
calculated as the final result. In the study, measurement errors are reduced by calculating the average of multiple results, and the final result is also better.

Elevation information of the building is extracted from the original point cloud data, and the minimum number of clusters is determined by the point cloud density of the grid.
For different data, the final parameters are not same, but the contour lines of the building can be accurately extracted in the end. Therefore, the algorithm is suitable for different examples. Figure 17 is the processing result of another example.

5. Conclusion
Achieving single bodies is one of the major problems that need to be solved in the application of oblique photography technology. To address this problem, this study proposed a method of automatically achieving single bodies. At present, a single body is achieved by drawing a contour line via human-computer interaction or importing vector files. However, this method is time-consuming and has poor accuracy. This study used the point cloud data extracted from an oblique photography model to realize an automatic extraction of contour lines. In the contour line extraction, the initial point cloud data are screened by setting the size of the virtual grid and the height difference threshold to improve efficiency and accuracy. Afterward, the point cloud is clustered using the DBSCAN clustering algorithm, and the building outline is constructed by the algorithm. The extracted contour line is then used to cut the original model, and the boundary triangulation is reconstructed to generate a new single model. This new model can be managed separately without affecting the overall display effect. Experimental results show that the proposed method is easy to implement and has high accuracy. This method is of great significance in achieving a single body and can further promote the application of oblique photography technology.

Data Availability
The data used in this article is oblique photography data obtained by drones, which is relatively easy to obtain.

Conflicts of Interest
The authors declare that they have no conflicts of interest.

Acknowledgments
This research was jointly supported by the Fundamental Research Funds for the Central Universities (N2124005 and N2001020) and the National Natural Science Foundation of China (41871310).

References
[1] Y.-S. Song, “Cultural assets reconstruction using efficient 3D-positioning method for tourism geographic information system,” vol. 22, no. 1, pp. 97–111, 2010.
[2] L. Xiaopeng, L. Dunlong, L. Junsong, and Z. Mei, “Research on a 3D geological disaster monitoring platform based on REST service,” International Journal of Geo-Information, vol. 7, no. 6, p. 226, 2018.
[3] J. G. Singla and K. Padia, “A novel approach for generation and visualization of virtual 3D city model using open source libraries,” Journal of the Indian Society of Remote Sensing, vol. 49, 2020.
[4] I. Mademlis, V. Mygdalis, N. Nikolaidis et al., “High-level multiple-UAV cinematography tools for covering outdoor events,” IEEE Transactions on Broadcasting, vol. 65, no. 3, pp. 627–635, 2019.
[5] M. Lin, F. Y. Li, and H. J. S. Zhou, “A research on the combination of oblique photography and mobile applications based on the sustainable development of tourism,” Sustainability, vol. 12, 2020.
[6] R. Zhang, H. Li, K. Duan et al., “Automatic detection of earthquake-damaged buildings by integrating UAV oblique photography and infrared thermal imaging,” Remote Sensing, vol. 12, no. 16, p. 2621, 2020.
[7] G. Verhoeven, “Taking computer vision aloft-archaeological three-dimensional reconstructions from aerial photographs with photoscan,” Archaeological Prospection, vol. 18, no. 1, pp. 67–73, 2011.
[8] W. Weimin, “Research on the automatic ID method of oblique photography model,” Geomatics & Spatial Information Technology, vol. 41, no. 11, p. 3, 2018, in Chinese.
[9] C. Liangchao, Z. Yong, and W. Junyong, “A method of achieving single body for real 3D model generated by oblique photography,” Bulletin of Surveying and Mapping, vol. 10, no. 6, p. 5, 2018, in Chinese.
[10] R. S. Jia and H. M. Sun, Research of 3D Stratum Model and Cutting Algorithm Based on DEMs-TEN, Springer, Heidelberg, Germany, 2011.

[11] N. Zan, "Research on the single modeling of oblique photogra-phy," Geospatial Information, vol. 17, no. 3, p. 4, 2019, in Chinese.

[12] C. H. Lindenbeck, H. D. Ebert, H. Ulmer, L. Pallozzi Lav-orate, and R. Pfug, "TRICUT: a program to clip triangle meshes using the rapid and triangle libraries and the visual-ization toolkit," Computers & Geosciences, vol. 28, no. 7, pp. 841–850, 2002.

[13] T. Czerniawski and F. Leite, "Automated digital modeling of existing buildings: a review of visual object recognition methods," Automation in Construction, vol. 113, p. 103131, 2020.

[14] X. Yang, T. Zhang, and Y. Lu, "Method for building recogni-tion from FLIR images," IEEE Aerospace and Electronic Systems Magazine, vol. 26, no. 5, pp. 28–33, 2011.

[15] A. W. Coburn, R. J. Spence, and A. Pomonis, "Factors de-termining human casualty levels in earthquakes: mortality prediction in building collapse," in Proceedings of the Tenth World Conference on Earthquake Engineering, vol. 10, pp. 5989–5994, Madrid, Spain, July 1992.

[16] M. Bassier, R. Klein, B. V. Genechten, and M. Vergauwen, "IFCwall reconstruction from unstructured point clouds," ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. IV-2, 2018.

[17] T. H. Kolbe, G. Gröger, and L. Plümer, CityGML–3D City Models and their Potential for Emergency Response, CRC Press, Boca Raton, FL, USA, 2008.

[18] D. J. Elshehaby and L. G. El-deen Taha, "A new expert system module for building detection in Urban areas using spectral information and LiDAR data," Applied Geomatics, vol. 1, 2009.

[19] Y. Chen, W. Zhang, G. Zhou, and G. Yan, "A novel building boundary reconstruction method based on lidar data and images," Laser Sensing and Imaging and Applications, International Society for Optics and Photonics, vol. 8905, , p. 890522, 2013.

[20] A. Mohammad and S. F. Clive, "Automatic segmentation of raw LiDAR data for extraction of building roofs," Remote Sensing, vol. 6, no. 5, pp. 3716–3751, 2014.

[21] R. Richter, M. Behrens, and J. Döllner, "Object class seg-mentation of massive 3D point clouds of urban areas using point cloud topology," International Journal of Remote Sensing, vol. 34, no. 23, pp. 8408–8424, 2013.

[22] M. Awrangjeb, G. Lu, and C. Fraser, "Automatic building extraction from LiDAR data covering complex urban scenes," The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XL-3, no. 3, pp. 25–32, 2014.

[23] N. Akkiraju, H. Edelsbrunner, M. Facello, P. Fu, E. Mucke, and C. Varela, "Alpha shapes: definition and software," Proceedings of the 1st International Computational Geometry Software Workshop, vol. 63, p. 66, 1995.

[24] M. Awrangjeb, C. S. Zhang, and C. S. Fraser, "Building de-tection in complex scenes thorough effective separation of buildings from trees," Photogrammetric Engineering and Remote Sensing, vol. 78, no. 11, p. 1162, 2012 in English.

[25] Y. Wang, X. Hao, and L. Ying, "Study of method for achieving 3D model to be single based on oblique photogrammetry image," CEA, vol. 54, 2018.

[26] M. Awrangjeb, M. Ravanbakhash, and C. S. Fraser, "Automatic detection of residential buildings using LiDAR data and multispectral imagery," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 65, no. 5, pp. 457–467, 2010, in English.

[27] Y. Dai, J. Gong, Y. Li, and Q. Feng, "Building segmentation and outline extraction from UAV image-derived point clouds by a line growing algorithm," International Journal of Digital Earth, vol. 10, no. 11, pp. 1077–1097, 2017.

[28] F. Rottensteiner, J. Trinder, S. Clode, and K. Kubik, "Using the Dempster-Shafer method for the fusion of LiDAR data and multi-spectral images for building detection," Information Fusion, vol. 6, no. 4, pp. 283–300, 2005.

[29] F. Bergamasco, M. Pistellato, A. Albarelli, and A. Torsello, "Cylinders extraction in non-oriented point clouds as a clustering problem," Pattern Recognition, vol. 107, p. 107443, 2020.

[30] S.-M. Hu, J.-X. Cai, and Y.-K. Lai, "Semantic labeling and instance segmentation of 3D point clouds using patch context analysis and multiscale processing," IEEE Transactions on Visualization and Computer Graphics, vol. 26, no. 7, 2018.

[31] S. Y. Li, W. J. Li, Z. Y. Lin, and S. J. Yi, "Method for 3D city building continuous transformation based on an improved LOD topological data structure," ISPRS International Journal of Geo-Information, vol. 8, no. 11, p. 14, 2019 in English.

[32] W. Sun, J. Wang, and F. Jin, "An automatic coordinate unification method of multitemporal point clouds based on virtual reference datum detection," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 13, pp. 3942–3950, 2020, in English.

[33] Y. Jung, S. W. Seo, and S. W. Kim, "Curb detection and tracking in low-resolution 3D point clouds based on opti-mization framework," IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 9, pp. 1–16, 2019.

[34] H. T. Nguyen, E.-H. Lee, C. H. Bae, and S. Lee, "Multiple object detection based on clustering and deep learning methods," Sensors, vol. 20, no. 16, p. 4424, 2020.

[35] F. Xue, W. Lu, Z. Chen, C. J. Webster, and R. Sensing, "From LiDAR point cloud towards digital twin city: clustering city objects based on Gestalt principles," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 167, pp. 418–431, 2020.

[36] S. Zhou, X. Liu, C. Wang, and B. Yong, "Non-iterative denoising algorithm based on a dual threshold for a 3D point cloud," Optics and Lasers in Engineering, vol. 126, no. 45, 2020.

[37] A. M. Abadi, C. M. Rowe, and M. Andrade, "Climate re-gionalization in Bolivia: a combination of non-hierarchical and consensus clustering analyses based on precipitation and temperature," International Journal of Climatology, vol. 40, no. 17, 2020.

[38] Y. Ge, Z. Lin, H. Tang, P. Zhong, and B. Cao, "Measurement of particle size of loose accumulation based on Alpha shapes (AS) and hill climbing-region growing (HC-rg) algorithms," Sensors (Basel, Switzerland), vol. 20, no. 3, p. 17, 2020 in English.

[39] Y. Liu and G. Yin, "The Delaunay triangulation learner and its ensembles," Computational Statistics & Data Analysis, vol. 152, Article ID 107030, 2020.

[40] Y. Y. Li Biao, H. Yang, and D. Hefang, "An empirical study on the accuracy of UAV low-altitude tilt photogrammetry," Mine Survey, vol. 5, no. 48, pp. 123–126, 2020, in Chinese.

[41] L. K. Wu Jiajie, L. Sun, J. Deng, H. Xia, and Q. Zheng, "Research on weighted least squares method for optimizing the accuracy of vectorization of oblique photogrammetry model," Bulletin of Surveying and Mapping, vol. 4, pp. 116–119+135, 2021, in Chinese.