J Inf Process Syst, Vol.16, No.3, pp.688~698, June 2020
https://doi.org/10.3745/JIPS.01.0057

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

A Point Clouds Fast Thinning Algorithm Based on Sample Point Spatial Neighborhood

Jiaxing Wei*, Maolin Xu*, and Hongling Xiu*

Abstract

Point clouds have ability to express the spatial entities, however, the point clouds redundancy always involves some uncertainties in computer recognition and model construction. Therefore, point clouds thinning is an indispensable step in point clouds model reconstruction and other applications. To overcome the shortcomings of complex classification index and long time consuming in existing point clouds thinning algorithms, this paper proposes a point clouds fast thinning algorithm. Specifically, the two-dimensional index is established in plane linear array (x, y) for the scanned point clouds, and the thresholds of adjacent point distance difference and height difference are employed to further delete or retain the selected sample point. Sequentially, the index of sample point is traversed forwardly and backwardly until the process of point clouds thinning is completed. The results suggest that the proposed new algorithm can be applied to different targets when the thresholds are built in advance. Besides, the new method also performs superiority in time consuming, modelling accuracy and feature retention by comparing with octree thinning algorithm.

Keywords

Fast Thinning Algorithm, Model Deviation, Point Clouds Thinning, Octree Thinning Algorithm, Thinning Rate, Visualization

1. Introduction

Point clouds can be scanned directly from light detection and ranging (LIDAR) technique or obtained indirectly through multiple view registration of images. At most previous studies, point clouds based on LIDAR and images have been widely used in digital elevation model (DEM) generation, antiquities protection, industrial collision detection and related domains [1-3]. In general, point clouds have ability to express the spatial entities. Meanwhile, the acquisition of point clouds has advantages of real-time and extensive. In terms of complex target modelling, the number of point clouds may reach ten-million level. It not only has no substantial benefits for the expression of the target, but also brings some difficulties to the computer recognition and storage. Accordingly, on the premise of ensuring the integrity and accuracy of target features, it's important to remove redundant points for three-dimensional (3D) reconstruction and explicit expression. The purpose of point clouds thinning is to remove the large amount of redundant points and retain the characteristics of points. Currently, there has been proposed many point clouds thinning algorithms [4-6]. The core of point clouds thinning is how to build the spatial index efficiently

Manuscript received February 19, 2019; first revision May 9, 2019; accepted July 8, 2019.

Corresponding Author: Maolin Xu (1574991189@qq.com)

* School of Civil Engineering, University of Science and Technology Liaoning, Anshan, China (18741297225@163.com, 1574991189@qq.com, 18341269177@163.com)

based on the adjacent point information. In theory, the model with low degree of complexity has the disadvantage of low thinning accuracy and the model with high degree of complexity usually involves more computing times. As a result, a fast and accurate thinning algorithm is essential for small and medium-sized users.

In order to solve above problems, this paper proposes a point clouds fast thinning algorithm. The two-dimensional (2D) index is established for the plane linear array (x, y). Firstly, the distance difference between adjacent sample points is taken as the validate condition. The distance difference threshold of point clouds thinning is set to mark whether the selected sample point is deleted or retained. Sequentially, the height difference of marked points is employed to further analysis. The height difference threshold is set to investigate whether the height difference of marked points is larger than the height difference threshold. After that, retaining the points less than the threshold and deleting the points greater than the threshold. Finally, traversing all the index of sample points forwardly and backwardly, and completing the point clouds thinning. Differ from the data density reduction (DDR) algorithm proposed by [7], our algorithm can effectively detect the changes of sample points in horizontal and vertical directions. Overall, our contributions can be summarized as follows:

- We propose a point clouds fast thinning algorithm, aiming to overcome the shortcomings of low thinning accuracy and more computing time by previous algorithms, and it's suitable for the small and medium-sized users.
- Differ from previous algorithms, the proposed method has the advantage of simple 2D index, and it bypasses complex process of index establishment and longer thinning time.
- In the validation of point clouds modelling, the proposed new algorithm has reached an accuracy comparable level by comparing with octree-based region growing algorithm.

This paper is structured as follows. Section 2 discusses the related work. Section 3 presents the methodology and data. The following Section 4 gives the results. In Section 5, the results are discussed. And the last section draws the conclusion of our research.

2. Related Work

For the research of point clouds thinning, there are several proposed algorithms. The enveloping box method builds enveloping boxes by obtaining the spatial range of point clouds, aiming to subdivide small enveloping boxes. After that replacing all points in the enveloping boxes with the barycenter points or the points closest to the barycenter [8]. The spatial index of enveloping box method is so simple that the feature points of targets are removed [9]. The random sampling method is to develop a multivariate random function according to the features of point clouds, and generate a set of random numbers. The core of mentioned method is to delete the points corresponding to these random numbers until the number of point clouds meets the users' requirements. However, the random sampling method often involves randomness, which will greatly reduce the accuracy of point clouds representation [10]. The curvature sampling method relies on the curvature threshold of point clouds, and it can retain the target local feature points. However, the curvature sampling method usually spends long computing time in index establishment, which weakens the efficiency of point clouds processing [11,12]. The algorithm named octree for point clouds thinning is to build root node and child nodes, aiming to thin the points in child nodes that closest to the mean of point normal vectors [13,14]. The accuracy of point clouds thinning by this method is higher than other methods, but the main drawback of this method is involved more threshold inputs. In addition, the computing times of octree method is large, which causing a high memory consumption. Therefore, for small and medium-sized users, it's necessary to choose a point clouds thinning algorithm with small computing footprint, simple spatial index and accurate feature retention.

According to the above point clouds thinning algorithms, the optimal methods with different degree of complexity are developed. In the research of [15], a global clustering point clouds thinning algorithm is defined by gathering the subset of original points according to the specified thinning rate. Although the number of points has achieved the specific level, the low accuracy of approximated point to model distances is involved and long time to build the spatial index of point is inevitable. A new point clouds thinning method is proposed by [16], the thinning process is completed by maintaining the position of feature points and removing the selected points based on the range of target points. The new method can obtain the points with high thinning rate, but it only works well for individual forms of point clouds. Following the research of [17], a point clouds thinning algorithm based on the normal vectors is conducted to retain the point clouds boundaries. The mentioned method adopts octree index to build the spatial topology relationship, and the boundary points are retained using a simple and effective method. The time complexity is involved in their study, although it performs a better thinning rate and accuracy compared with previous work. The planar simplification and triangulation algorithm is developed by Whelan et al. [18], which is capable of running in real-time while producing a segmentation faithful to what would be achieved using a batch segmentation method. However, the simplicity of spatial index is a blemish that they still need to further consider.

In general, the above statements of previous studies have pointed out three issues in the current point clouds thinning algorithms. Firstly, the time complexity is usually involved in the complex structure of point clouds, which involves more computing times than simple spatial index. Secondly, the feature of point clouds boundaries are usually removed in the thinning process, it's crucial to retain the feature points for well expression. The last and most important is that a simple and accurate point clouds thinning algorithm is always needed in different application fields for small and medium-sized users.

3. Methodology and Data

3.1 Fast Thinning Algorithm

In this study, considering the computing times and spatial complexity of point clouds thinning method, a fast thinning algorithm with simple index and acceptable accuracy is proposed. The point clouds fast thinning algorithm adopts the method of spatial index judgment. Since the initial point clouds spatial index is well structured, the 2D index needs to be established for scanned point clouds plane (x, y). The purpose of establishing 2D index is mainly to record the position of sample points in the 2D plane and increase the position attributes of sample points. The point clouds are scanned in small horizontal (X) steps and large vertical (Y) steps, which means the index spread over the sample points follows a serpentine pattern.

Besides, taking adjacent point clouds distance difference threshold (d_s) and height difference threshold (d_h) as the conditions of the determination of valid point. Actually, d_s and d_h are joint conditions, only if these conditions are satisfied can we make a further judgment. After the judgment conditions are executed, the operation of deleting and retaining sample points can be carried out. Finally, traversing all the sample point index and completing the point clouds thinning. The flow of point clouds fast thinning algorithm is shown in Fig. 1, where *n* is the number of point clouds. In theory, a forward traverse is required for all sample points in the process of point clouds thinning, and the valid and invalid points can be directly interpreted. However, a single traverse cannot guarantee all the remaining valid points exceed the distance difference threshold and height difference threshold. Therefore, the backward traverse is adopted after the forward traverse, which means the index traverses from *n* to 1.

Specially, the proposed point clouds fast thinning algorithm can be summarized as follows:

Step 1. Build the initial 2D plane index (x, y) for the scanned point clouds.

Step 2. Calculate the plane distances of neighbor points, and record corresponding height information.

Step 3. Set the plane distance difference threshold, and traverse forwardly to select the points under the threshold.

Step 4. Set the height difference threshold, to further judge the selected points.

Step 5. Backward traverse, to find the valid points based on the forward traverse results.

Step 6. Update the point index, to retain the valid points.

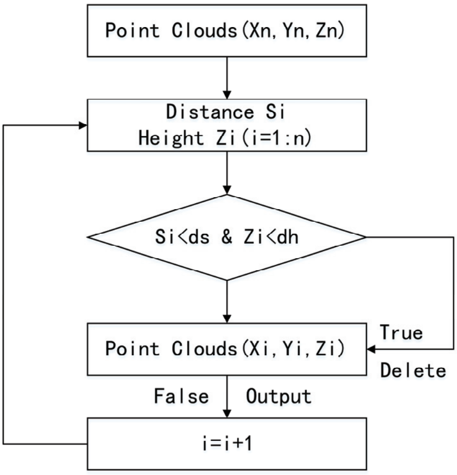
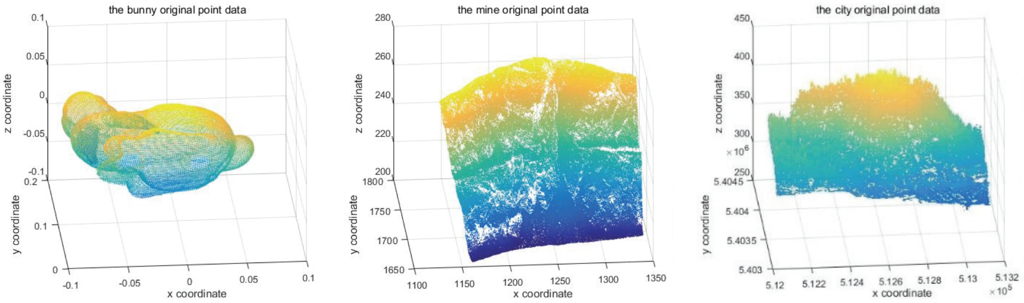


Fig. 1. The flowchart of point clouds fast thinning algorithm.



(a) (b) (c)

Fig. 2. The 3D view of point clouds for (a) Stanford rabbit, (b) open-pit mine, and (c) residential area.

3.2 Data

In order to validate the stability and efficiency of proposed algorithm, the point clouds datasets with different feature and number are used in this paper. The Stanford rabbit has a small number of point clouds and obvious curvature changes. The slope surface point clouds are located in an open-pit mine, the number of which is huge and intervals are small. The residential area, including a large number of point clouds and large intervals. The 3D view of these datasets are presented in Fig. 2. In this paper, the platform of thinning is MATLAB and the computer is configured with Core i5 4G, which conforms to the basic configuration requirements of small and medium-sized users. Besides, it also highlights the superiority of our proposed fast thinning algorithm in low configuration.

4. Results

Due to the initial point clouds intervals of three datasets are different, different distance and height thresholds need to be applied to different targets. For the Stanford rabbit, the curvature changes obviously. Meanwhile, the size of Stanford rabbit is small and point clouds are dense, so the distance difference threshold is set at centimeter level and the magnitude of height difference threshold is less than distance difference threshold. The curvature of open-pit mine point clouds are approximately 90° and the overlap of elevation is high, so the height difference threshold need to consider this factor. The residential point clouds are scanned from airborne LIDAR, the plane point interval is large and the distance difference threshold should be increased. In addition, the quality of point clouds thinning is evaluated by the visualization of point clouds, the retention of point clouds details and the thinning rate. The detail information before and after point clouds thinning is shown in Table 1.

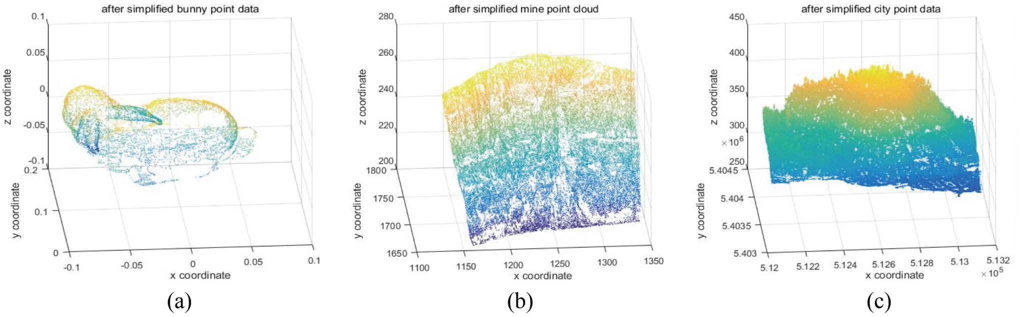
It can be learned from Table 1 that the number of point clouds changes significantly before and after thinning, and the thinning rate of rabbit has reached 82%, while the residential area is only 55%. The reason lies in the establishment of thinning threshold, that is, the setting of distance and height difference thresholds will greatly affect the efficiency of point clouds thinning. Certainly, the setting of thresholds is a reasonable judgment made by users based on their own point clouds information. In addition, it's easy to find the time of fast thinning algorithm proposed in this paper is short. For small and medium-size users, the time of point clouds thinning within 1 second is completely acceptable.

Table 1. The detail information before and after point clouds thinning

Data	Numbers of original point clouds	Threshold (m)	Numbers of thinned point clouds	Rate (%)	Time (s)
Rabbit	35,947	$d_s = 0.05$, $d_h = 0.1$	6,557	82	0.173
Mine	92,583	$d_s = 0.5$, $d_h = 0.8$	28,199	70	0.265
Residential area	683,204	$d_s = 1.5$, $d_h = 0.5$	308,865	55	0.448

In general, the thinning rate can judge the efficiency of point clouds thinning, and the thinning time has ability to describe the complexity of the algorithm. The most important thing is whether the thinned point clouds have an influence on the later application. To further illustrated, the 3D view of point clouds after thinning is shown in Fig. 3.

From Fig. 3 we can find that the original contours of three datasets are retained, and the curvature characteristics of original point clouds are well preserved. Due to the large number of point clouds in residential areas, the point clouds after thinning is not obvious on the display of Stanford rabbits. It’s more representative for open-pit mine, because the original point clouds are dense. The number of point clouds in the part of few point clouds remains unchanged, which makes the characteristics of point clouds be well preserved. Since the open-pit mine point clouds perform the best visualization and with large enough area, we only adopt the open-pit mine point clouds as the test object for the comparison with other algorithms.



(a) (b) (c)

Fig. 3. The 3D view of thinned point clouds of (a) Stanford rabbit, (b) open-pit mine and (c) residential area.

5. Discussion

5.1 Point Clouds Visualization

In order to test the stability of our proposed algorithm, the octree-based region growing algorithm is adopted to compare with the fast thinning algorithm [6]. The octree-based region growing algorithm is demonstrated to be a robust method with accurate thinning results, and this method is cited by many other studies. It should be also emphasized that consider the scale and characteristics of original point clouds, the open-pit mine is involved for the further analysis. Although the threshold setting of our proposed algorithm is different from the contrast one, the height difference threshold of fast thinning method can be approximately regarded as the height of closed minimum cube in octree-based region growing algorithm. Thereafter, the size threshold of the minimum voxel used in octree method can be regarded as the distance difference threshold of the fast thinning method. Therefore, following the above rules, we have enough reasons to compare two different algorithms. Specifically, the octree-based region growing algorithm taking the segmented nodes as index, but the principle of the segmented node is similar to that of the sample points in the fast thinning algorithm. By means of the above threshold setting, the point clouds of open-pit mine are thinned by our proposed method and octree-based region growing algorithm, respectively. The comparison of point clouds thinning by above two algorithms are shown in Table 2.

Table 2. Comparisons of two thinning algorithms

Algorithm	Original point clouds numbers	Thinned point clouds numbers	Rate (%)	Time (s)	Operatio n times	Memory profiler (%)
Octree-based region growing	92,583	34,083	63	17.3	t ≤ 2*n	33
Fast thinning	92,583	37,923	61	0.3	t ≤ 8*n	26

From Table 2 we can learn that the fast thinning algorithm proposed in this paper is superior to the octree-based region growing algorithm in thinning time. Meanwhile, the thinning rate of fast thinning method is similar to octree-based region growing algorithm. The advantage in thinning time is mainly reflected in the simple index and less computing times. In addition, the memory profiler of octree-based region growing occupies 33%, and our proposed algorithm is 26%. This lies in the method of spatial index established and the times of index traversed, this also demonstrates the simplicity of our fast thinning algorithm. It can be seen that the advantage of octree-based region growing algorithm lies in the construction of spatial index (see Fig. 4). In terms of visualization, octree-based region growing algorithm has a more compact organization, while fast thinning algorithm behaves more chaotic than former. Point clouds are ultimately used in model reconstruction and other applications, so we will further validate the visualization of point clouds model.

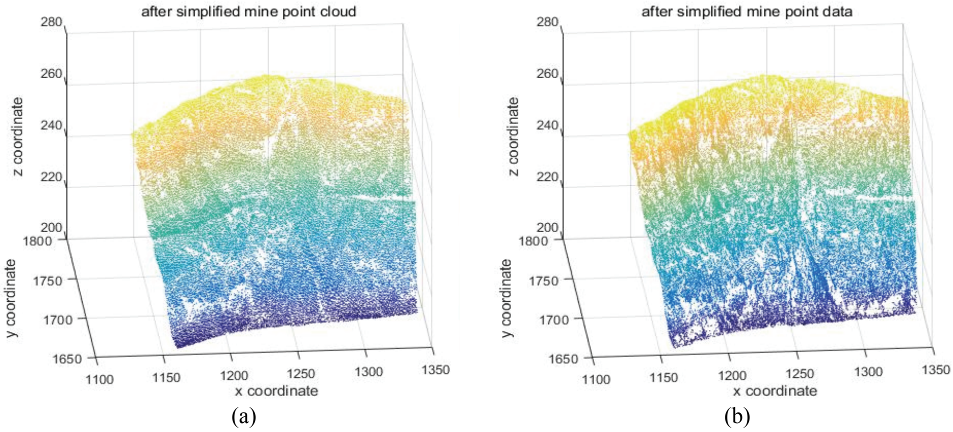


Fig. 4. Point clouds visualization by (a) octree-based region growing algorithm and (b) our proposed fast thinning algorithm.

5.2 Model Visualization

In order to avoid the subjectivities in point clouds visualization, we model the two kinds of thinned point clouds, respectively. The cubic method with interpolation function is adopted for point clouds modelling [19,20]. The visualization of modelled point clouds by above two algorithms are presented in Fig. 5. There is no significant difference in the point clouds model and the contour of entire target. Although the point clouds visualization in Fig. 4 is different, the model reconstruction result is basically similar.

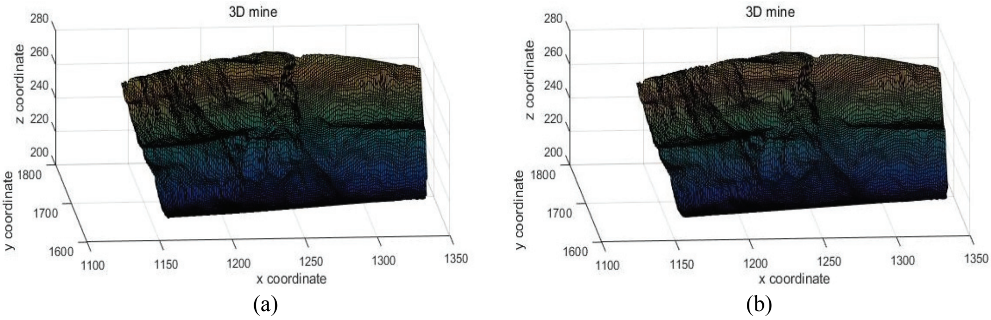
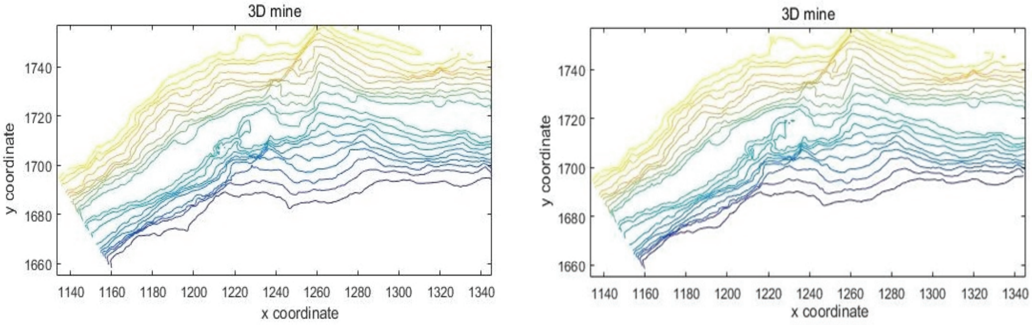


Fig. 5. The point clouds model construction using thinned point clouds by (a) octree algorithm (b) our proposed algorithm.

The above comparison is presented in a 2D view, and it's difficult to observe the detail changes in local contours. Consequently, we extract the contours of thinned point clouds, which is shown in Fig. 6. The contours are defined by x and y coordinates, and points with the same elevation are connected into a curve. That means contours have advantage in the expression of variations in different height layers. The contours variation in high region is basically identical, and there is a small deviation in the central region. The bottom contours produced by fast thinning algorithm is smoother than octree algorithm, and octree algorithm has obvious turbulence and low divergence. The main difference lies in the existence of holes at the bottom of open-pit mine in fast thinning algorithm, which causes the area become smoothed when contours are established. The reason can be interpreted by the difference of initial setting in distance and height difference thresholds.



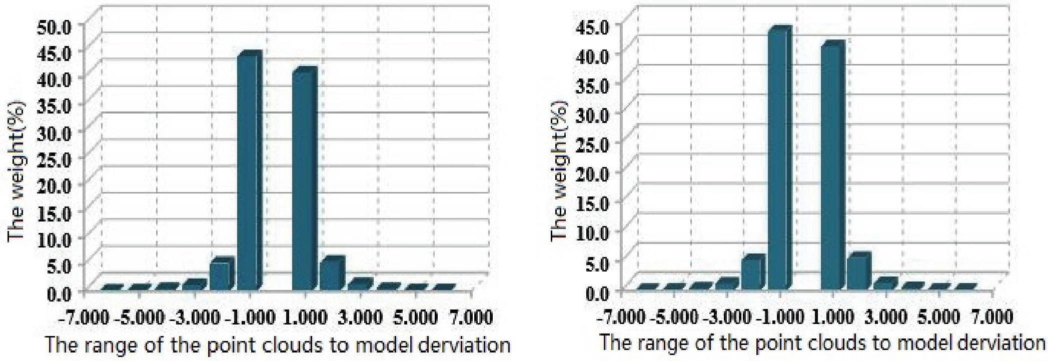
(a) (b)

Fig. 6. The point clouds contours interpolation of 3D open-pit mine using thinned point clouds by (a) octree algorithm and (b) our proposed algorithm.

5.3 Points to Model Deviation

The above comparisons of point clouds and corresponding model belong to qualitative analysis, and the deviation of point clouds to model is used as the quantitative analysis method for the comparison of two algorithms. The initial model adopted to overlay analysis is built by the original point clouds, which is not thinned. At the same time, the cubic interpolation is still adopted to model construction. The thinned point clouds are overlaid to the model, the interval and weight of deviation distance are taken as the indicators for quantitative comparison. This method can directly determine the influence of point clouds thinning algorithm on model construction, because the cubic interpolation applied to two thinned point clouds has never changed. Therefore, if a local point is lost, the interpolation algorithm will automatically match another point to draw a triangle. This may indirectly affect the shape of the entire model [21].

The deviation interval and weight of point clouds to model superimposed by two algorithms are presented in Fig. 7. It is not difficult to find that the deviation interval in point clouds to model is mainly distributed at -1.5 mm to 1.5 mm. When the deviation is 1.5 mm, two kinds of weights are located at 3% to 5%. When the deviation is 1.0 mm, these two weights are occupied at 40% to 42%. And when the deviation is 1.0 mm, the weights are located at 38% to 40%. When the deviation is 1.5 mm, both two weights are located at 3% to 5%. Through the superposition analysis of point clouds to model, we conclude that although the fast thinning algorithm is slightly different from the octree algorithm in point clouds and model visualization, these differences are not significant to affect the quality of model construction. As a result, the above validation further illustrates the robust of our proposed algorithm.



(a) (b)

Fig. 7. The point clouds to model deviation of 3D open-pit mine using thinned point clouds by (a) octree algorithm and (b) our proposed algorithm.

6. Conclusions

This paper proposes a point clouds fast thinning algorithm, to overcome the shortcomings of complex classification index and long-time consuming in existing point clouds thinning algorithms. Compared with the octree-based region growing algorithm, our proposed algorithm has disadvantages in the visualization of point clouds and model. However, we obtain certain advantages in the thinning and modeling time, and the thinning rate has reached a level of octree-based region growing algorithm. The algorithm used in this paper illustrates the rapidity and stability of point clouds thinning, which would be useful for small and medium-sized users.

The algorithm proposed in this paper is similar to most existing algorithms, which always needs to set the thresholds in advance. In theory, the threshold problem is unavoidable in current point clouds thinning domain, because the users need to thin their point clouds in their different applications. Therefore, our future work will focus the threshold reduction. Besides, we will consider the application of our proposed algorithm, it will be useful for the fast feature extraction, fast model construction and related aspects.

Acknowledgement

This paper is supported by the National Key Research and Development Project of China (No. 2016YFC0801602). And the Innovation Training Program of the University of Science and Technology Liaoning (No. LKDYC201822).

References

[1] X. Yang, L. W. Clements, M. Luo, S. Narasimhan, R. C. Thompson, B. M. Dawant, and M. I. Miga, “Stereovision-based integrated system for point cloud reconstruction and simulated brain shift validation,” *Journal of Medical Imaging*, vol. 4, no. 3, article no. 035002, 2017.

[2] P. Grussenmeyer and O. Al Khalil, "From metric image archives to point cloud reconstruction: case study of the Great Mosque of Aleppo in Syria," in *Proceedings of International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, Ottawa, Canada, 2017, pp. 295-301.

[3] A. K. Patil, P. Holi, S. K. Lee, and Y. H. Chai, "An adaptive approach for the reconstruction and modeling of as-built 3D pipelines from point clouds," *Automation in Construction*, vol. 75, pp. 65-78, 2017.

[4] E. El-Sayed, R. F. Abdel-Kader, H. Nashaat, and M. Marei, "Plane detection in 3D point cloud using octree-balanced density down-sampling and iterative adaptive plane extraction," *IET Image Processing*, vol. 12, no. 9, pp. 1595-1605, 2018.

[5] M. Pauly, M. Gross, and L. P. Kobbelt, "Efficient simplification of point-sampled surfaces," in *Proceedings of IEEE Visualization*, Boston, MA, 2002, pp. 163-170.

[6] A. V. Vo, L. Truong-Hong, D. F. Laefer, and M. Bertolotto, "Octree-based region growing for point cloud segmentation," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 104, pp. 88-100, 2015.

[7] S. Pamela, "A data density reduction algorithm for Post-processed airborne lidar bathymetric survey data," Ph.D. dissertation, University of Florida, FL, 2000.

[8] H. Woo, E. Kang, S. Wang, and K. H. Lee, "A new segmentation method for point cloud data," *International Journal of Machine Tools and Manufacture*, vol. 42, no. 2, pp. 167-178, 2002.

[9] T. Liu, Z. Xu, C. M. Sha, and J. T. Zhao, "Curvature estimation of scattered-point cloud data based on bounding box method," *Science Technology and Engineering*, vol. 9, no. 12, pp. 3333-3336, 2009.

[10] P. Burger and H. J. Wuensche, "Fast multi-pass 3D point segmentation based on a structured mesh graph for ground vehicles," in *Proceedings of 2018 IEEE Intelligent Vehicles Symposium (IV)*, Changshu, China, 2018, pp. 2150-2156.

[11] L. Pagani and P. J. Scott, "Curvature based sampling of curves and surfaces," *Computer Aided Geometric Design*, viol. 59, pp. 32-48, 2018.

[12] Y. Wang, Y. Ma, A. X. Zhu, H. Zhao, and L. Liao, "Accurate facade feature extraction method for buildings from three-dimensional point cloud data considering structural information," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 139, pp. 146-153, 2018.

[13] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for point-cloud shape detection," *Computer Graphics Forum*, vol. 26, no. 2, pp. 214-226, 2007.

[14] R. Schnabel and R. Klein, "Octree-based point-cloud compression," in *Proceedings of 3rd Eurographics/ IEEE VGTC Conference on Point-Based Graphics*, Boston, MA, 2009, pp. 111-120.

[15] H. Song and H. Y. Feng, "A global clustering approach to point cloud simplification with a specified data reduction ratio," *Computer-Aided Design*, vol. 40, no. 3, pp. 281-292, 2008.

[16] H. Han, X. Han, F. Sun, and C. Huang, "Point cloud simplification with preserved edge based on normal vector," *Optik-International Journal for Light and Electron Optics*, vol. 126, no. 19, pp. 2157-2162, 2015.

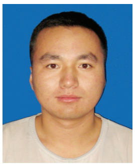
[17] X. Yang, K. Matsuyama, K. Konno, and Y. Tokuyama, "Feature-preserving simplification of point cloud by using clustering approach based on mean curvature," *The Society for Art and Science*, vol. 14, no. 4, pp. 117-128, 2015.

[18] T. Whelan, L. Ma, E. Bondarev, P. H. N. de With, and J. McDonald, "Incremental and batch planar simplification of dense point cloud maps," *Robotics and Autonomous Systems*, vol. 69, pp. 3-14, 2015.

[19] F. N. Fritsch and R. E. Carlson, "Monotone piecewise cubic interpolation," *SIAM Journal on Numerical Analysis*, vol. 17, no. 2, pp. 238-246, 1980.

[20] H. T. Huynh, "Accurate monotone cubic interpolation," *SIAM Journal on Numerical Analysis*, vol. 30, no. 1, pp. 57-100, 1993.

[21] K. W. Brodlie and S. Butt, "Preserving convexity using piecewise cubic interpolation," *Computers & Graphics*, vol. 15, no. 1, pp. 15-23, 1991.



Jiaxing Wei **https://orcid.org/0000-0002-6818-9271**

He is currently a master graduate student of University of Science and Technology Liaoning, Anshan, China. His research interest includes data processing and application of point clouds.

Maolin Xu **https://orcid.org/0000-0002-9469-9021**

He is currently a professor of University of Science and Technology Liaoning, Anshan, China. His research interest includes mine monitoring and measurement data processing.

Hongling Xiu **https://orcid.org/0000-0002-2081-8307**

She is currently a master graduate student of University of Science and Technology Liaoning, Anshan, China. Her research interest includes remote sensing image processing and application.