A Gross Error Elimination Method for Point Cloud Data Based on Kd-tree

Qi Kang 1, Guoman Huang 1 , Yang Shucheng 1

[1] Key Laboratory of Earth Observation and Geospatial Information Science of NASG, Chinese Academy of Surveying and Mapping, Beijing, 100830, China

KEY WORDS: Data point cloud, Gross error elimination, Kd-tree

ABSTRACT:

Point cloud data has been one type of widely used data sources in the field of remote sensing. Key steps of point cloud data’s pro-processing focus on gross error elimination and quality control. Owing to the volume feature of point could data, existed gross error elimination methods need spend massive memory both in space and time. This paper employed a new method which based on Kd-tree algorithm to construct, k-nearest neighbor algorithm to search, settled appropriate threshold to determine with result turns out a judgement that whether target point is or not an outlier. Experimental results show that, our proposed algorithm will help to delete gross error in point cloud data and facilitate to decrease memory consumption, improve efficiency.

1. INSTRUCTION

Point cloud data is one type of the widely used data sources in many communities such as photogrammetry, remote sensing, computer vision, and machine learning. At present, point cloud data has been used in high precision large scale digital elevation model (DEM) production, power line inspection, three-dimensional modeling of buildings, classification of surface cover, change monitoring, forest resources survey, biomass estimation, mine measurement, city planning and other fields.

During the acquisition process of point cloud data, some elevation anomalies are measured due to the obstruction and concealing of the environmental factors in the scanning process, which are called low gross error and high gross error. The low gross error refers to the obstruction of moving vehicles, pedestrians, and trees during the scanning process of the building, and the non-uniform reflection characteristics of the entities themself, which may cause some abnormal elevation points measured by the “multipath” effect. The high gross error is formed because the reflected signals of birds and low-flying objects under the airborne measurement platform are recorded as the reflected signals of the measured targets, which will result in the formation of error points that are significantly higher than the ground and objects.

With the wide application of point cloud data, researchers realized t that filtering and its associated quality control are the most critical and time-consuming steps in the pro-processing of point cloud data. To achieve a highly efficient, accurate, and adaptable filtering effect, gross error elimination is a necessary and critical step. Compared with image data, point cloud data has its own characteristics in terms of content, form and others, including features such as massiveness, high redundancy, missing local data, uneven density, and unstructured.

Therefore, we need to use more efficient methods to organize and manage point cloud data. This paper employed a new method which based on Kd-tree algorithm to eliminate the gross error in the point cloud data.

2. TITLE AND ABSTRACT BLOCK

In order to cope with the problem of organizing point cloud data, Lu Yueming proposed the method that the point cloud data was arranged and sorted according to a specified rule, and then organized by a compound structure of spatial octree and balanced binary tree(2008). Luo Dean proposed a new ground-based LIDAR data simplification algorithm based on quad-tree partition (2005). The Kd-tree technology, first proposed by Bentley in 1975, is a spatial indexing technique for spatial expansion of BSP trees. It is used to create a spatial index of spatial massive point targets. Afterwards, some researchers have studied the Kd-tree algorithm. Jiang Jingyu et al. thought that Kd-tree was an effective way to organize LiDAR point cloud data (2007).

As for the outlier eliminating, Liu Jingnan et al. (2008) used elevation histogram to eliminate significant high and low gross errors. However, this method cannot eliminate gross errors that are close to the surface. Silv án-C árdenas and Wang et al. (2009) first used the elevation histogram to remove gross errors, then organized the data based on the irregular triangulation to eliminate outliers, but the distance threshold of this method has no obvious distribution law and it is time- consuming.

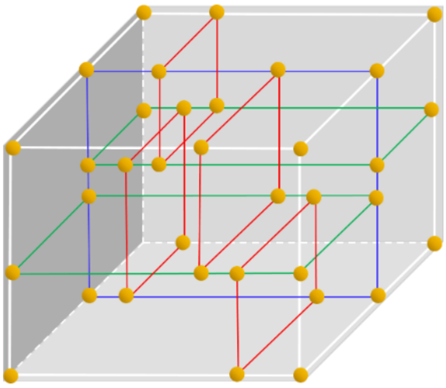


Figure 1. The division of Kd-tree in three-dimensional space
The core technology of Kd-tree algorithm is divided into construction and query of the tree.

3. METHODOLOGY

Kd-tree (K-Dimensional) is a data structure that divides data points in a k-dimensional space. It is a special case of binary space partitioning tree which is mainly used to retrieve multi-attribute data or multi-dimensional point data. Kd-tree is actually

This contribution has been peer-reviewed.

a kind of balanced binary tree. In addition to leaf nodes, each node represents a data point in this k-dimensional space, and searching is performed by continuously recursively dividing a space into two subspaces. In the two-dimensional case, the whole space is a rectangle; in the three-dimensional case, the whole space is a cube (see Figure 1), this paper uses a two-dimensional Kd-tree built on the X, Y axis.

3.1 The construction of Kd-tree

The establishment of Kd-tree is a top-down recursive process. For a k-dimensional data, select the dimension K with a larger variance, and then select the median m in this dimension as the hyperplane Pivot to divide the data set into two sub sets, at the same time create tree nodes to store (K, m). Each time the above division is performed, the data set is divided into two parts in the dimension K. The value of the left part is smaller than the value of the right part in the K-dimension. Repeat the steps of selecting dimensions and medians to division until all subsets cannot be divided then points in the data set are saved to leaf nodes. Figure 2(a) is a space partition of two-dimensional data sets corresponding to Kd-tree. Figure 2(b) is a Kd-tree constructed.

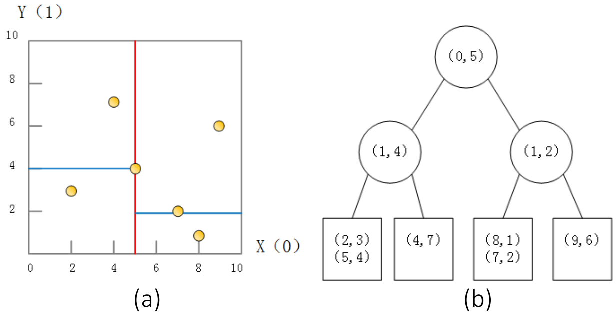


Figure 2(a) The division of Kd-tree in two-dimensional space, (b) Data are stored in leaf nodes of the Kd-tree

3.2 The search process of K- nearest neighbor

After the Kd-tree is constructed, the spatial data can be retrieved quickly through the tree structure. This paper uses the k-nearest neighbor query method. First, find the nearest neighbor point which is the leaf node in the same subspace as the point to be queried along the search path through the binary tree search (Compare the split dimension value of the node to be queried and the split node, and enter the left subtree branch if it is less than or equal to, enter the right subtree branch if it is greater than, until it reaches the leaf node). Then backtrack the search path and determine if the nodes on the search path may have data points of other leaf node space that are closer to the query point. If possible, jump to other leaf nodes space to search (add other leaves to the search path) and repeat the process until the search path is empty.

3.3 Key technical process

According to the above principles, the following technical process of gross error elimination is determined in this paper.

(1) Eliminate the obvious high gross errors and low gross errors by elevation histogram. If there is no obvious high and low gross errors through visual analysis, go directly to step (2).

(2) Construct Kd-tree, build Kd-tree for point cloud data preserved after histogram analysis.

(3) Set the value of k and use the constructed Kd-tree to quickly search and calculate the difference M between the k nearest neighbors' mean elevation and this point's elevation.

(4) Calculate the mean $\bar{\delta}$ and standard deviation σ of all points' M.

(5) Traverse all points, if the difference of this point's M and the mean $\bar{\delta}$ of all points' M is larger than 3σ $(\mathrm{M} - \bar{\delta} > 3\sigma)$, then this point is defined as a gross point and eliminate it.

(6) Evaluate the effect of gross error elimination through visual observation.
Figure 3 illustrates the above process in a simplified sequence flow diagram.

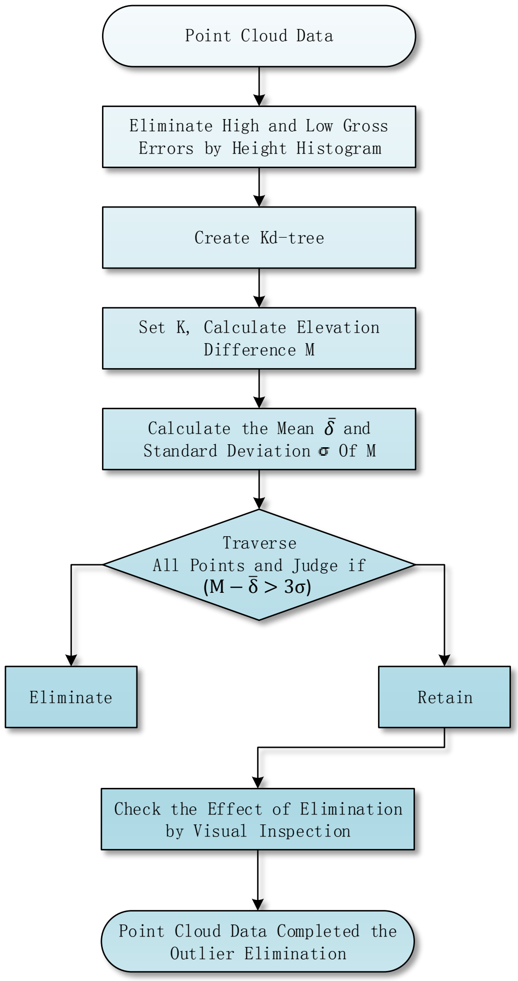

Point Cloud Data
Eliminate High and Low Gross Errors by Height Histogram
Create Kd-tree
Set K, Calculate Elevation Difference M
Calculate the Mean $\bar{\delta}$ and Standard Deviation σ Of M
Traverse All Points and Judge if $(\mathrm{M} - \bar{\delta} > 3\sigma)$
Eliminate
Retain
Check the Effect of Elimination by Visual Inspection
Point Cloud Data Completed the Outlier Elimination

Figure 3 The flowchart of key technology

4. TEST AND RESULT ANALYSIS

Based on the above principles, the experiment program of gross error elimination was developed using the Visual Studio 2008 platform. There are two sets of data for the experiment, one set is urban area data, and the other is mountain area data.

Through observation, it is found that both the urban area data and the mountain area data have gross errors. First, eliminate the obvious gross errors and by elevation histogram. In the course of the experiment, the values of k were set as 30 and 60. The statistical data of the experimental results are shown in Table 1.

		Urban area	Mountain area
Elimination proportion	K=30	0.047%	0.022%
	K=60	0.051%	0.024%
Time	K=30	88s	176s
	K=60	191s	340s

Table 1. Statistics of the results using the proposed gross error elimination method

Figure 4 is the comparison of the urban and mountain areas before and after the elimination of the gross errors. After the proposed method elimination in this paper, the effect of gross error elimination in urban area is more obvious, although the gross errors in mountain area was also eliminated, because of the complex terrain of the mountain, the inspection of the effect of the gross error elimination was not so obvious in the visual analysis. In both scenarios, the range of elevation values has changed, and the number of point cloud data has also changed, indicating that the gross error elimination method in this paper can effectively eliminate gross errors and retain the data characteristics of the correct points.

In order to evaluate the performance of the algorithm, the following verification is also carried out in this paper. Change the value of k, check the influence of the value of k, compare with outlier elimination method based on entity.

According to the data obtained analysis, when the k value varies from 30 to 60, the ratio of the gross error elimination is relatively stable, and the elevation value change is also relatively constant. The visual analysis also shows that in this range, the effect of the gross error elimination tends to be consistent.

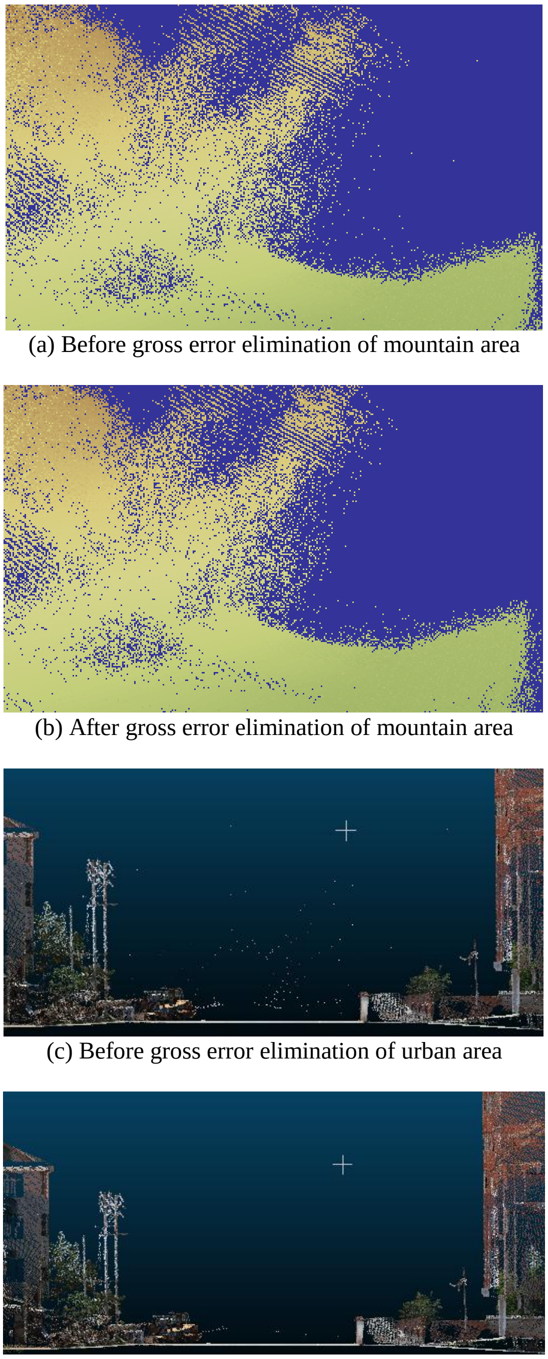
At the same time, it is found that the method proposed in this paper can save more time on the basis of effective gross error elimination. Because of the entity-based approach, the surface of the entity needs to be fitted first, which needs more time than the Kd-tree.

The above experiments show that the method proposed in this paper has a good effect of gross elimination, strong adaptability and high efficiency.

5. CONCLUSION

The quality control of point cloud data is the key to the application of point cloud data, and gross error elimination is a necessary step in point cloud data processing. The gross error elimination method for point cloud data based on Kd-tree in this paper, can improve the efficiency of the point cloud data organization and decrease memory consumption. Then calculate the difference of the point's elevation and k nearest neighbors' mean elevation, and calculate the mean and standard deviation of the above difference using the K-nearest neighbor algorithm. Calculate each point and then compare it to the empirical threshold, decide if the point is a gross point. Through the experiment, the applicability and effecency of the method this paper proposed which use Kd-tree to organize point cloud data and K- nearest neighbor algorithm to eliminate outliers are verified.

However, the method presented in this paper also has its disadvantages: the setting of k value is still an empirical threshold method; and there is no mass data for testing. The next step in the research around this paper focuses on: further improving the rationality of the k-value setting, improving the accuracy of the results and improving the algorithm to deal with massive data increase efficiency.


(a) Before gross error elimination of mountain area

(b) After gross error elimination of mountain area

(c) Before gross error elimination of urban area

(d) After gross error elimination of urban area

Figure 4. Comparison of the side views between the before-after gross elimination of urban and mountain areas

ACKNOWLEDGEMENTS

This work was supported by The Basic Plan of Science and Technology of Surveying and Mapping (NO.2018KJ0103).

REFERENCES

BENTLEY J L., 1975. Multidimensional Binary Search Trees Used for Associative Searching. Communications of the ACM, 18(9), pp. 509-517.

Chen M. L., Wang Y. C., Tian S. Y. et al., 2016. A point cloud data organization method based on linear KD tree. Bulletin of Surveying and mapping, 1, pp. 23-27.

Yang J. S., 2014. A Massive Airborne LiDAR Data Organization and Management Method Combining a Quadtree and a KD Tree. Journal of Wuhan University (Information Science Edition), 39(8), pp. 918-922.

Zhang X. H., 2007. Theory and Method of Airborne Laser Radar Measurement Technology. Wuhan University Press, Wuhan, China.

SITHOLE G., VOSSELMAN G., 2004. Experimental Comparison of Filtering Algorithms for Bare-earth Extraction from Airborne Laser Scanning Point Clouds. ISPRS Journal of Photogrammetry and Remote Sensing, 59(1-2), pp.85-101.

NENE S. A., NAYAR S. K., 1997. A simple Algorithm for Nearest Neighbor Search in High Dimensions. IEEE Pattern Analysis and Machine Intelligence, 19(9), pp. 989-1003.

GREENSPAN M., YURICK M., 2003. Approximate KD Tree Search for Efficient ICP. Proceedings of Fourth International Conference on 3-D Digital Imaging and Modeling: IEEE, pp.442-448.

Zhang Y., Liu X. M., Sui Y. et al., 2009. Research and improvement of K-nearest neighbor point cloud denoising algorithm. Journal of Computer Applications, 29(4), pp. 1011-1014.

Zhang J. X., Lin X. G., Liang X. L., 2017. Advances and Prospects of Information Extraction from Point Clouds. Acta Geodaetica et Cartographica Sinica, 46(10), pp. 1460-1469.

MENG X., WANG L., SILV áN-C áRDENAS J. L., et al, 2009. A Multi-directional Ground Filtering Algorithm for Airborne LIDAR. ISPRS Journal of Photogrammetry and Remote Sensing, 64, pp.117-124.

Liu J., 2008. Adaptive Hierarchical and Weighted Iterative Filtering of Airborne LIDAR Data and Its Quality Assessment. Geomatics and Information Science of Wuhan University, 33(6), pp. 551-555.

Lu Y. M., He Y. J., 2008. Organization and Indexing Method for 3D Points Cloud Data. Geo-NFormation Science, 10(2), pp. 190-194.

Luo D. A., Liao L. Q., 2005. Ground-based LIDAR data simplification based on quad-tree partition. Computer Applications, 25(2), pp. 420-425.

Jiang J. Y., Zhang Z. X., 2008. Filtering of Airborne Lidar Point Clouds for Complex Cityscapes. Geo-spatial Information Science, 11(1), pp. 21-25.

https://doi.org/10.5194/isprs-archives-XLII-3-719-2018