PAPER • **OPEN ACCESS**

A Research on MBES data classification denoising algorithm based on Octree index

To cite this article: Xiaokai Li *et al* 2019 *IOP Conf. Ser.: Earth Environ. Sci.* **237** 022011

View the article online for updates and enhancements.



IOP ebooks™

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the collection - download the first chapter of every title for free.

This content was downloaded from IP address 119.146.131.186 on 20/09/2019 at 22:57

A Research on MBES data classification denoising algorithm based on Octree index

Xiaokai, Li[1*], Fan Zhang[1], Yue Wang[1] and Huanjian, Chen[1]

[1]College of Geomatics, Shandong University of Science and Technology, Qingdao, Shandong 266590, China

[*]Corresponding author's e-mail: 734166372@qq.com

Abstract. For the Multi-Beam Echo Sounder (MBES) data denoising problem, the noise is divided into near-ground noise and outlier noise away from the seabed. The algorithm in this paper considers the spatial distribution characteristics of noise and removes two kinds of noise by initial denoising and precise denoising. The algorithm first uses the octree index to organize the point cloud and removes the outlier noise. Then combined with Bayesian estimation theory and statistical methods to remove near-ground noise. This paper designs experiments to process MBES data and compare it with the statistical filter of PCL point cloud library. The experimental results show that compared with the PCL statistical filter, the algorithm used in this paper has good denoising effect and can better preserve the information such as regional boundaries.

1.Introduction

As an important submarine detection equipment, MBES is widely used in modern marine monitoring and hydrographic surveys [1]. However, because of its self-noise and actual sea conditions, MBES data has a lot of noise. Some of the noise is close to the surface of the seabed, which is difficult to distinguish from each other [2]. The traditional artificial interaction denoising method is inefficient, so a filtering algorithm with reliable denoising quality and high degree of automation becomes a great need in post-processing work. Many foreign scholars in the early days proposed using statistical theory to detect abnormal values of multi-beam sounding, including Ware method [3] and Du method [4]. These methods are less automated and the terrain reliability can not be guaranteed. In response to these problems, many experts use a combination of mathematical statistics and spatial location to deal with this problem. Among them, Canepa et al proposed to use the generated triangulation map to automatically identify the multi-beam sounding anomaly value [5]. Debese et al proposed the use of statistical principles and spatiotemporal properties for dynamic estimation of surfaces [6]. JT Bjørke et al used the spatial correlation between data points and proposed the average difference subdivision method to divide the sounding data into grids, and quickly identify and eliminate multi-beam sounding outliers [7]. B.R.Calder and L.A.Mayer proposed the CUBE algorithm to process MBES data [8]. The core of the algorithm is to establish the total propagation uncertainty of each sounding point [9]. The median filtering and Kalman filtering are used to estimate the depth and error of the sounding node. The above method needs to configure the correction accuracy and repeatedly adjust the threshold, and the denoising efficiency is not very high. In response to this problem, Lu Dan et al proposed a method for detecting and culling abnormal depth values based on truncated least squares estimation [10]. The method uses the high collapse point characteristic of the truncated least squares estimation to fit the local seabed trend surface, and detects and eliminates the anomaly outlier by comparing the depth data

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

with the residual of the seabed trend surface. However, the sensitivity of the method to noise and the efficiency of data processing are related to the size of the detection window, and the window size needs to be manually and repeatedly adjusted. Zhang Zhiwei et al proposed an algorithm based on the estimation of the weighted iteration to fit the trend surface [11]. The algorithm is highly sensitive to noise, but it is difficult to guarantee the computational efficiency when iteratively processing the massive point cloud. Zhang Zhiheng introduced the natural point influence domain and improved the trend surface filtering method [12]. The method is sensitive to noise, but this method needs to search for the natural neighbors of each sounding point. The algorithm is complex and it is not efficient to process massive data.

Through the analysis of a large number of MBES data noise, this paper divides the MBES data noise into two categories: 1. Near-ground noise. This kind of noise is close to the surface of the seabed topography; 2. Obvious outlier noise. This noise is far from the terrain and is sparse.

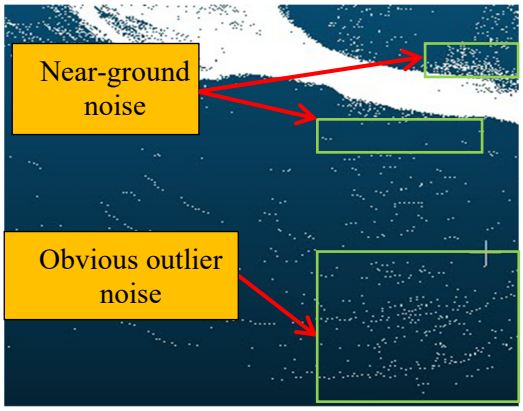


Figure 1. Schematic diagram of MBES noise.

Based on the above analysis, this paper designs algorithms to remove two types of noise according to the characteristics of them. The algorithm is divided into two steps: Initial denoising and precise denoising, respectively removing obvious outlier noise and near-ground noise. Firstly, the octree index is built to construct the topological relationship of the scattered point cloud, and use the number of points in the voxel as a standard to receive or reject the point to remove obvious outlier noise. Then bayesian dynamic estimation theory and statistical methods are used to remove near-surface noise close to the terrain. Design experiments to analyze the execution efficiency and denoising effect of the algorithm.

2. Algorithm principle

The basic idea of the algorithm: First, the octree is used to establish the topological relationship between scattered point clouds, and the number of points in the voxels of all water depth points is counted until the point cloud traversal ends, then remove significant outlier noise outside of given threshold range. Then create a spatial grid, and the depth value of the grid node is estimated based on Bayesian prediction, and the depth difference between each water depth point and the grid node in the node voxel is calculated. According to the statistical characteristics of the point cloud, noise outside the given error range is removed. When the point cloud traverses, the hash table is used to improve the execution efficiency.

The algorithm has the following characteristics: (1) The applicability of the algorithm is wide, and the effect of removing sparse noise is good; (2) The water depth prediction algorithm based on Bayesian estimation is robust and can distinguish between terrain and near-surface noise; (3) The octree tree is simple to build and has a high level of automation, which is suitable for the processing of massive point cloud data. The algorithm principle is described in detail below.

2.1Initial denoising

The basis for removing outlier noise is to establish an octree index. An octree is a tree-like data structure used to describe three-dimensional space. Each node of the octree represents a volume element of a cube, and each node has eight child nodes. The volume elements represented by the eight child nodes are added together to be equal to the volume of the parent node. The topological relationship of point cloud data can be constructed by establishing an octree. By obtaining the neighborhood information of the voxels at each water depth point and setting an appropriate threshold, the apparent outlier noise can be easier removed. The effect of MBES data to establish octree index is shown in Figure 2.

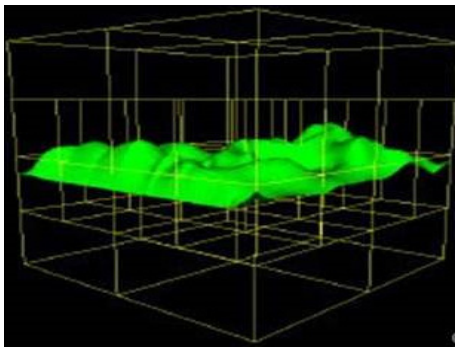


Figure 2. octree indexed terrain.

2.2 Precise denoising

2.2.1 Node depth estimation. Take the octree tree voxel center point as a node of the grid to establish a spatial grid. The Bayesian dynamic model theory is used to estimate the depth value of the grid nodes, and the dynamic linear model is established to dynamically predict and update grid nodes. Let $E_j = \{e_j(s_i): s_i \in S_j\} = \{e_j[0], \dots, e_j[N_j - 1]\}$ be the predicted sequence of N_j water depth points for the grid nodes, $e_j(n) = [d_j[n], \sigma_j^2[n]]^T \ (0 < n \le N_j)$ is the prediction information of the nth water depth point to the grid node. The current mesh node's depth estimate is $\varepsilon_j(n) = [\hat{z}_j[n|n], \hat{\sigma}_j^2[n|n]]^T$, and the node update is performed by the following steps:

$$\hat{\sigma}_j^2[n|n-1] = \hat{\sigma}_j^2[n-1|n-1] \quad (1)$$

$$\hat{z}_j[n|n-1] = \hat{z}_j[n-1|n-1] \quad (2)$$

$$K_j[n] = \frac{\hat{\sigma}_j^2[n|n-1]}{\hat{\sigma}_j^2[n|n-1] + \sigma_j^2[n]} \quad (3)$$

$$v_j[n] = d_j[n] - \hat{z}_j[n|n-1] \quad (4)$$

$$\hat{z}_j[n|n] = \hat{z}_j[n|n-1] + K_j[n]v_j \quad (5)$$

$$\hat{\sigma}_j^2[n|n] = K_j[n]\sigma_j^2[n] \quad (6)$$

According to the above method, the calculation is completed when all neighborhood information participates in the water depth estimation. Finally, get the estimated results of the grid nodes that contain all the neighborhood information.

2.2.2 Statistical denoising. After all grid nodes have been estimated, the depth difference between the grid nodes and the water depth points in their voxels needs to be calculated and counted. Since the

near-ground noise is distributed on both sides of the terrain, the depth difference distribution is symmetrical. The statistical analysis of depth difference is performed on the measured data of 7986215 data points. The resolution of the octree is 0.5m, and the depth difference distribution is shown in Figure 3. The abscissa indicates the depth difference and the ordinate indicates the probability density. It can be found that the depth difference approximates the normal distribution.

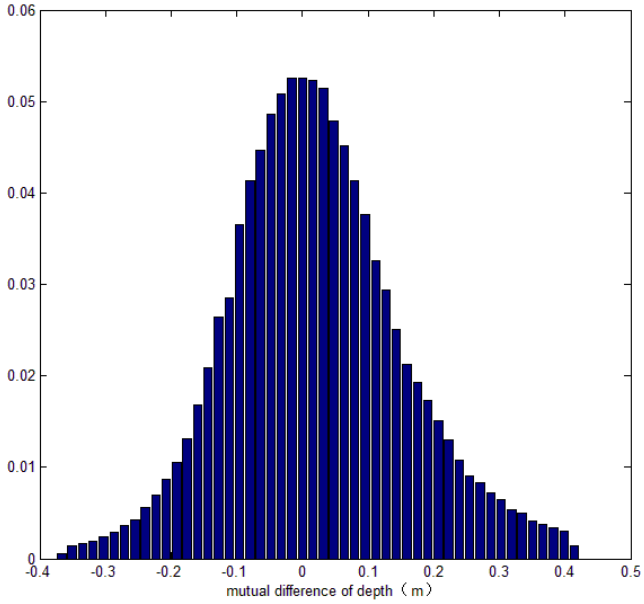


Figure 3. depth difference statistics.

According to the above characteristics, the point cloud outside the given error range is removed based on the statistical method.

3. Algorithm implementation

3.1 Algorithm implementation steps

1) Input the $cloudPoint$, build the index tree $Octree$, and turn 2);

2) Traverse every point in the $cloudPoint$. Perform steps 3) - 4), if the traversal ends turns 5);

3) Check whether $searchPoint$ has been searched; check whether $searchPoint$ is an invalid point. If the above two meet one of them, jump out of this cycle, turn 2), otherwise, turn 4);

4) Count the number of voxel neighborhood points in $searchPoint$, and denoted it as K_Num. Set the threshold to λ_1, if $K_Num > \lambda_1$, then assume that $searchPoint$ is not a distinct outlier noise point. Store the $searchPoint$ index number and its corresponding neighborhood numbers in the hash table 1, turns 2) ;

5) Enter the point cloud set corresponding to the hash table 1, establish an index tree $Octree$ and establish rule grid, and turn 6);

6) Traverse each point $voxelPoint$ in the set of mesh nodes. Perform steps 7) - 10), traverse ends and turn 11);

7) Check whether $voxelPoint$ has been searched; check whether $voxelPoint$ is an invalid point. If the above two meet one of them, jump out of this cycle, turn 6), otherwise, turn 8);

8) Perform a voxel neighborhood search on $voxelPoint$, save the index number of the searched water depth point and the corresponding sounding value to the vector container $pointVector$, and turn 9);

9) Traverse the point in the vector container $pointVector$, the depth value of $voxelPoint$ estimated according to the method in 1.2.1, turn 10);

10) The points in the container $pointVector$ are traversed and the depth difference is calculated,

and store the index number and its corresponding depth difference in hash table 2, turns 6);

11) Calculate the mean and medium error of the distance difference in hash table 2, denoted as u and σ respectively;

12) Set threshold λ_2 and traverse hash table 2. If the distance corresponding to the index number in the hash table 2 is between $[u-\lambda_2\cdot\sigma,\ \mu+\lambda_2\cdot\sigma]$, the point corresponding to the index number is marked as a topographic point, otherwise it is marked as near-ground noise;

13) Save the topographic and noise point data separately to the PCD format file.

3.2Algorithm flow diagram

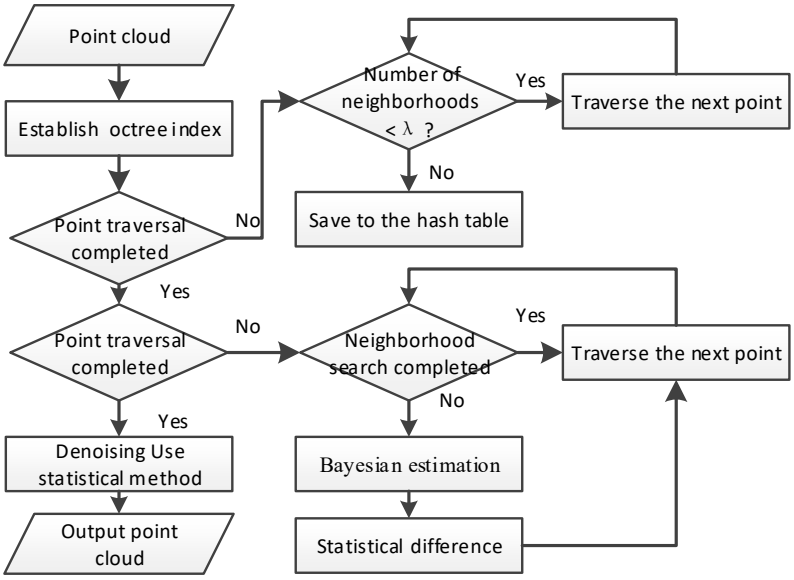


Figure 4. algorithm flow.

3.3 Program optimization

The thresholds involved in the algorithm are determined as follows:

1) The neighborhood number threshold λ_1 is to remove the low-density point cloud while avoiding accidental deletion of the surface and edge of the seabed topography, and in the case of the resolution of the octree is 0.5m, λ_1 takes 2;

2) The denoising threshold λ_2 determines is typically taken as 2σ.

4. Experimental Analysis

4.1Experimental data and environment

The experimental data is the measured point cloud collected by the R2Sonic2024 multi-beam system, and the measurement area is about 8 square kilometers. The basic configuration of the experimental computer is CPU: 1.8GHz, Intel(R) Core(TM) i7-4500U; RAM: 8GB. The experimental software environment is based on Qt4.8.7+VS2013 compiler +PCL (Ver1.8.0).

4.2 Experimental design and analysis.

4.2.1 Algorithm execution efficiency analysis. In this experiment, the same experimental data is processed by the algorithm and PCL statistical filter, and the execution time of the two methods is compared. The PCL filter parameters are $k=100$ and double medium error. The octree resolution of this algorithm is 5m. The experimental results are shown in Table 1 and Figure 5.

Table 1. Experimental result.

Points	**464015**	**1003030**	**1855303**	**4044695**	**6211414**	**7986215**	**10902918**	**12408427**	**14216843**
Algorithm	200.3	430.7	790.5	1801.8	2705.2	3539.3	4562.1	5288.5	5900.9
PCL	192.6	406.3	756.5	1638.0	2415.4	3116.1	4224.2	4887.5	5599.8

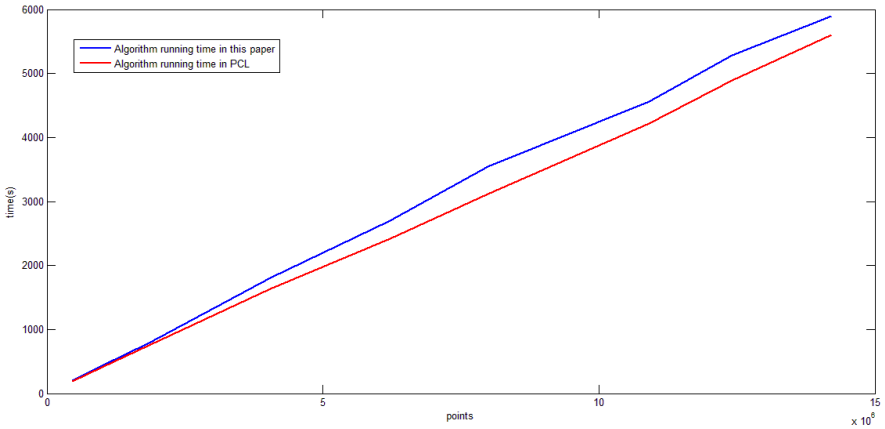
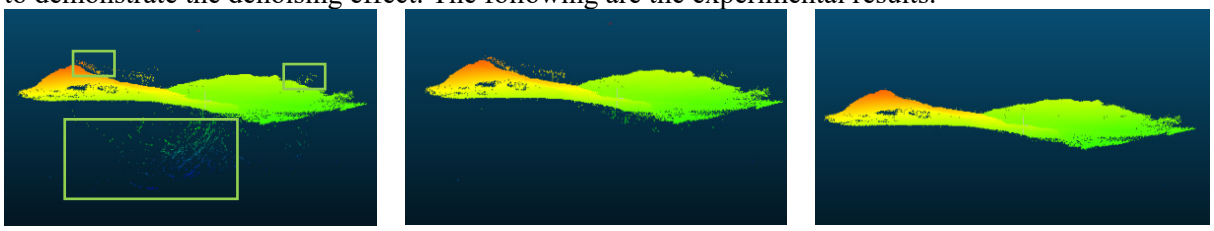


Figure 5. Experimental result diagram.

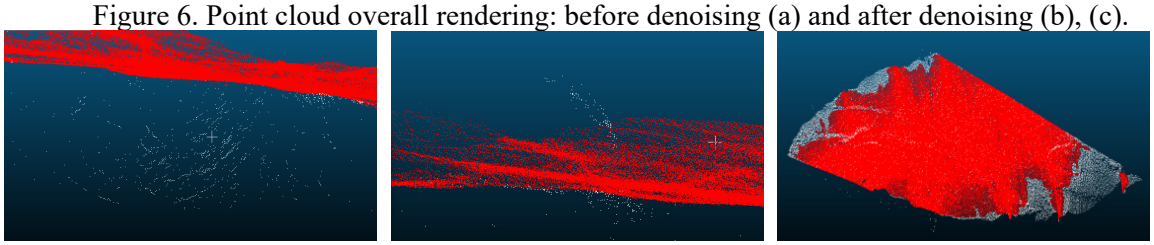
It can be obtained from the experimental results that as the number of points clouds increases, the time consumption of the PCL algorithm and the algorithm increase linearly. However, the execution time of the algorithm in this paper grows faster than the PCL algorithm, but eventually the difference between the two tends to be stable. This is because the efficiency of the algorithm in this paper is affected by the number of water depth points involved in the estimation. As the number of point clouds grows, the number of voxels increases, but the number of water depth points in voxels does not change significantly. Therefore, the execution efficiency of this algorithm increases linearly with the increase of point cloud number. The difference between algorithm efficiency and PCL algorithm is controlled at around 8%.

4.2.2 Algorithm denoising effect analysis. The purpose of this experiment was to analyze the denoising quality of the algorithm used in this paper and compare it with the PCL algorithm. In this experiment, a multi-beam point cloud containing 302062 data points with dense noise in the survey area is selected to demonstrate the denoising effect. The following are the experimental results.



(a) Pre-treatment effect (b) Initial denoising effect (c) Precise denoising effect

Figure 6. Point cloud overall rendering: before denoising (a) and after denoising (b), (c).



(d) Remove outlier noise (e) Remove near-ground noise (f) PCL denoising effect

Figure 7. The algorithm denoising details (d), (e) and PCL denoising algorithm (f).

Figure 6 shows the overall effect before and after denoising. Figure6 (a) shows the point cloud effect before denoising, Figure6 (b) shows the point cloud effect after initial denoising, and Figure6 (c) shows the point cloud effect after precise denoising. It can be found that after initial denoising, obvious outlier noise can be removed, and topographic and near-ground noise are preserved. After precise denoising, near-ground noise is removed and topographic features are preserved. Figure 7(d) and Figure 7 (e) shows the denoising details of the algorithm. The algorithm can distinguish terrain and noise better. Figure 7(f) is the denoising effect diagram of PCL algorithm. Denoising using PCL algorithm is difficult to preserve regional boundary information.

In this paper, 1422 noises are removed by initial denoising, 962 noises are removed by precise denoising, 299678 inner points are confirmed, and 298827 inner points are confirmed by PCL algorithm. Calculate the proportion of terrain that is considered to be a noise point. The algorithm error rate is 0.09% and the PCL algorithm error rate is 0.48%. In summary, it can be seen that the algorithm is superior to the PCL algorithm in terms of boundary reservation.

5. Conclusion

Aiming at the problem of multi-beam point cloud denoising, this paper divides the noise into near-ground noise and outlier noise according to the spatial distribution characteristics of noise. After initial denoising and precise denoising, two types of noise are removed respectively. The experimental results show that the algorithm has good denoising effect. Compared with the PCL statistical filter, the algorithm can better preserve the region boundary information. However, it is found that the denoising effect of the algorithm is not good when dealing with strip edges and data staggered layers. These situations will be the focus of subsequent research.

References

[1] Lu, X.P. (2016) Research progress and prospects of key technologies for multi-beam sounding data processing. Marine mapping., 36(4):1-6.
[2] Feng, D.H. (2018) Multi-beam point cloud denoising algorithm considering underwater terrain features. Journal of Surveying and Mapping Science and Technology.. 34(4):364-369.
[3] Ware, C. (1992) A system for cleaning high volume bathymetry .
[4] Du, Z., Wells, D., Mayer, L. (1996) An approach to automatic detection of outliers in multibeam echo sounding data. Hydrographic Journal.
[5] Canepa, G., Bergem, O. (2003) A new algorithm for automatic processing of bathymetric data. IEEE Journal of Oceanic Engineering., 2003, 28(1):62-77.
[6] Debese, N. (2007) Multibeam Echosounder Data Cleaning Through an Adaptive Surface-based Approach. Us Hydro:
[7] Bjørke J, T., Nilsen, S. (2009) Fast trend extraction and identification of spikes in bathymetric data. Computers & Geosciences., 35(6):1061-1071.
[8] Calder.B.R,, Mayer, L.A. (2013) Automatic processing of high-rate, high-density multibeam echosounder data. Geochemistry Geophysics Geosystems., 4(6):
[9] Ji.X. (2018) A Review of Research on Detection and Elimination Methods of Multi-beam Sounding Anomaly Data. Surveying science., 2018(1):38-44.
[10] Lu, D. (2012) Multi-beam anomaly sounding value rejection method based on truncated least squares estimation. Geodesy and geodynamics., 32(1):89-93.
[11] Zhang, Z. (2016) Application of Robust Estimation in Detection of Outliers of Multibeam Sounding. Journal of Liaoning Technical University., 2016(7):755-758.
[12] Zhang, Z., Peng, R., Huang, W. (2018) An Improved Algorithm of Tendency Surface Filtering in Multi-beam Bathymetric Data Considering the Natural Neighboring Points Influence Field. Acta Geodaetica Et Cartographica Sinica.,