GOSPODARKA SUROWCAMI MINERALNYMI – MINERAL RESOURCES MANAGEMENT

2025 Volume 41 Issue 2 Pages 197–211

DOI: 10.24425/gsm.2025.154547

LIRONG YANG[1], YANG LIU[2], CHONG CAO[3]

Adaptive ore point cloud filtering algorithm based on the K-nearest neighbor

Introduction

In recent years, ore crushing technology has advanced toward intelligence (Cai et al. 2021), employing technologies such as LiDAR, stereo vision, and ultrasound to sense the complex ore production environment and facilitate automated operations (Li et al. 2019, 2023). In the roughing process of ore, in order to prevent the oversized raw ore from entering the rough crushing equipment, which leads to the jamming and damage of the equipment, it is necessary to set up a grizzly screen on the raw ore silo to filter the oversized ore.

✉ Corresponding Author: Lirong Yang; e-mail: candy_yang_jx@163.com

[1] Jiangxi Mining and Metallurgical Engineering Research Center; School of Mechanical and Electrical Engineering, jiangxi University of Science and TechnologyGanzhou, jiangxi Province, China; ORCID iD: 0000-0003-0259-8459; e-mail: candy_yang_jx@163.com

[2] School of Mechanical and Electrical Engineering, jiangxi University of Science and TechnologyGanzhou, jiangxi Province, China; ORCID iD: 0009-0000-7732-6760; e-mail: 1556408795@qq.com

[3] School of Mechanical and Electrical Engineering, jiangxi University of Science and TechnologyGanzhou, jiangxi Province, China; e-mail: 172035074@qq.com

For large-size ore on the grizzly screen, manual crushing or robotic arm manipulation is commonly applied, posing high labor intensity and safety risks. This highlights the need for an intelligent robotic arm system that utilizes stereoscopic vision to autonomously locate and crush large ore on the grizzly screen (Weales 2018). In the research of stereo vision technology, due to the complexity of the factory environment, the actual acquisition of the original point cloud of the grizzly screen with ore generally has problems such as noise points and missing features. These problems will seriously affect the point cloud recognition and reconstruction accuracy, so it is necessary to study a filtering algorithm suitable for the noise characteristics of the ore point cloud data.

Some of the more commonly used filtering methods for complex surfaces like ores are mathematical form filtering (Pingel et al. 2013) and triangular mesh filtering (Axelsson 2000). However, the literature (Chen et al. 2021) has tested and analyzed the above two filtering methods. The results show that the above methods are less effective for filtering in complex scenarios. The various parameter settings in the filter require the user to have a lot of on-site experience and technical reserves, so these two methods are not applicable to the crushing of ore on the grizzly screen. While in other filtering methods. Moreno and Li (2016) compared point cloud filtering methods in real-time video streaming and proved the effectiveness of different algorithms in processing Kinect sensor data. Chen et al. (2023) proposed a filtering method considering uniform point distribution, which can enhance the distribution of the point cloud while preserving the feature information. Han et al. (2017, 2018) derived a linear model for the guided and filtered point cloud inspired by the guided image filtering method while analyzing the performance of various 3D point cloud filtering algorithms in detail in a review article. Wang and Jiang (Wang et al. 2022). The nonlocal position-based method proposed performs well in preserving geometric features. In addition, Lu et al. (2020) proposed a new deep-learning method for preserving geometric features that improve the automatic prediction of regular lines and can automatically estimate the point cloud normals. Zeybek and Şanlıoğlu (2019) investigated the filtering techniques for UAV point clouds. They demonstrated the effectiveness of the different methods in practical applications, but the evaluation criteria of the filtering results are relatively simple. Jia (Jia et al. 2019) provided a new idea for point cloud filtering with a classification method based on surface change factors. Zhang et al. (2016) proposed using a Cloth Simulation Filter (CSF) to filter the ground point cloud, but the effect is slightly insufficient on more complex surfaces.

Most of the above filtering methods are based on LiDAR. However, some mines are unable to configure high-precision LiDAR based on cost, and based on the improvement of optical sensor technology, the point cloud stereo reconstruction technology of binocular vision has become a new choice in recent years (Yang et al. 2024). Therefore, this paper analyzes the characteristics of the ore at the ore-crushing site and proposes an improved K-nearest-neighbor density filtering algorithm based on the point cloud captured by its binocular camera. Different from traditional ore point cloud processing algorithms, the algorithm proposed in this paper argues that regions of ore point clouds with significant density variations contain more important information than regions of uniform density, and larger elimination weights should be given to noise points that are slightly farther away from the central ore point cloud. Therefore, in this paper, according to the average size of the domain density variation of the target points and the clustering coefficient, the optimal neighboring point cloud weight coefficients, filtering parameters, and cost functions are obtained through experiments. Finally, the domain density can be adjusted adaptively to output the optimal ore point cloud model.

1. Algorithm fundamentals

1.1. Description of the problem

As a key screening step before the realization of ore refining, the use of a grizzly screen intercepts and crushes the oversized ore, which can protect the jaw crusher under the grizzly screen from jamming and causing material accumulation. In contrast, human crushing and worker-operated robotic arm crushing exist in the problem of low crushing efficiency, high labor intensity, and safety hazards, based on which there is a need to design an unmanned intelligent ore position measurement device to identify the grizzly screen and to identify the ore position on the screen. In this paper, a binocular camera is used to collect ore images, and according to the left and right views to take stereo vision algorithm 3D reconstruction to generate the point cloud, the process schematic shown in Figure 1, after downsampling

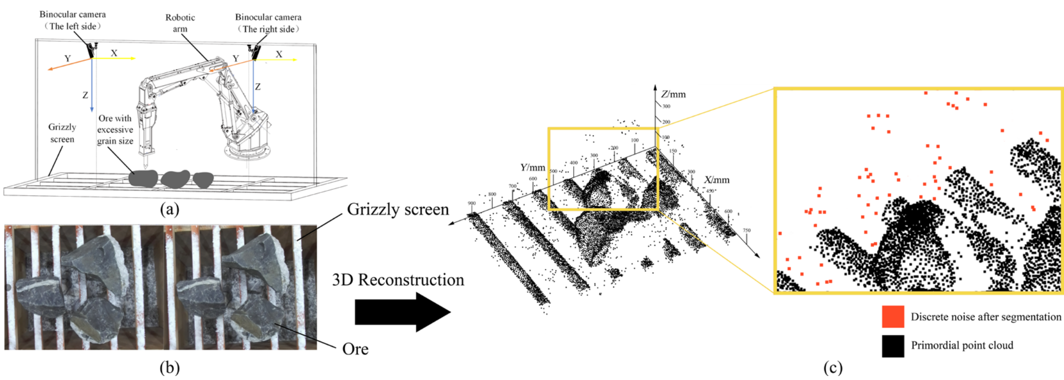

Fig. 1. 3D reconstruction of the ore on the grizzly screen
(a) Binocular camera installation diagram, (b) Grizzly screen and ore captured by binocular camera,
(c) Discrete noise in grizzly screen and ore point clouds

Rys. 1. Trójwymiarowa rekonstrukcja rudy na ekranie grizzly
(a) Schemat instalacji kamery lornetkowej, (b) Ekran grizzly i ruda uchwycone przez kamerę lornetkową,
(c) Dyskretny szum w chmurze punktów ekranu grizzly i rudy

the original point cloud generated, the point cloud data of the grid sieve and the ore position and center of gravity calculation of the processing difficulties are mainly for the acquisition of point cloud data, which is usually accompanied by a large amount of noise and usually accompanied by a large amount of noise. These noises mainly come from irregular reflections on the ore surface, interference from ambient light, and equipment errors. These noise points will lead to blurring of the point cloud data, which seriously affects the subsequent point cloud identification and reconstruction. To address this difficulty, this paper proposes a K-nearest neighbor density filtering algorithm to filter and smooth the noisy points in the point cloud.

1.2. K-neighborhood density filtering algorithm

The K-nearest neighbor density filtering algorithm consists of the following steps: first, input and read the point cloud data, then construct a KD-tree index structure to support neighbor searches. The K-nearest neighbor density algorithm calculates the local density and density variance of each point within the structure to retrieve local density information. Next, the K-means clustering algorithm is applied to categorize points by density values,

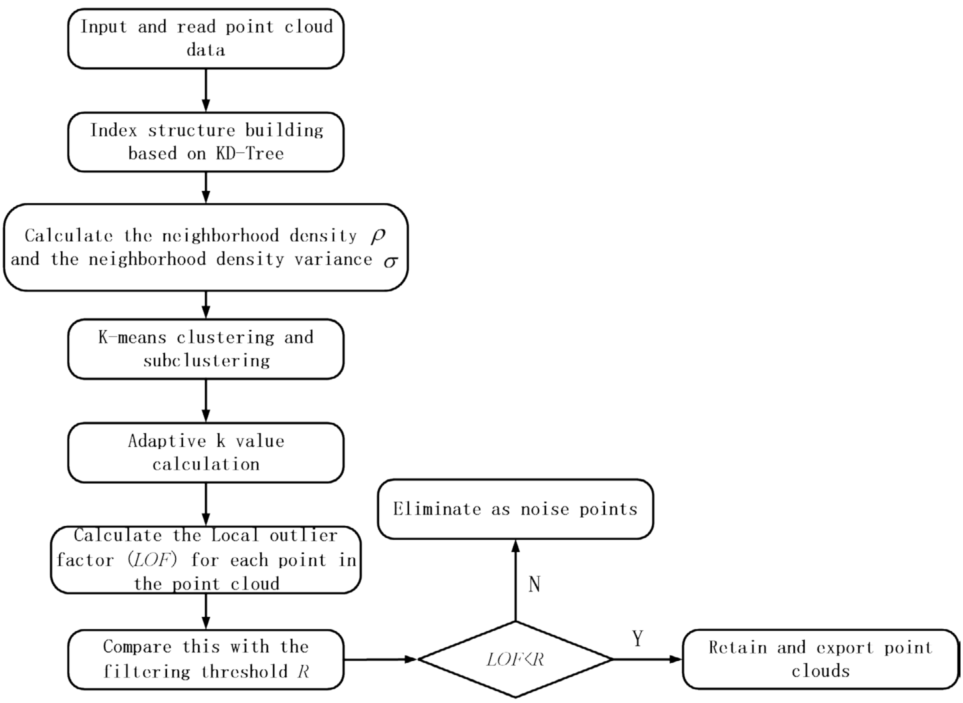

Fig. 2. K-neighborhood density filtering flowchart

Rys. 2. Schemat blokowy filtrowania gęstości sąsiedztwa K highlighting regions with varying densities within the point cloud. Based on these density clustering results and density variance information, the optimal K-value is obtained, the Outlier Factor is calculated, and points are removed to complete the point cloud filtering process. The process flow is illustrated in Figure 2.

1.3. Topological relationship construction and neighborhood density calculation

The original point cloud of ore is a disordered point cloud, including the corresponding structure of ore and grid sieve, and contains a large amount of outlier noise without any topological information between the points. Therefore, it is necessary to establish the topological relationship. In this study, a K-dimensional tree (K-D tree) is used to construct the topological relationship between the points in the point cloud. The main steps are as follows: Firstly, the root node is established, and the feature with the most significant variance value is selected as the segmentation feature according to the input point cloud data. Secondly, the corresponding median for the selected segmentation feature is calculated as the segmentation point to traverse all the data. Thirdly, during the traversal, points with features less than the median are divided into the left sub-node under the root node, while points with features greater than the median are divided into the right sub-node. Finally, this process is repeated for each child node until all the data are built on the leaf nodes, thus completing the establishment of the KD tree structure.

After the topological relationship is constructed, K neighborhood search method is used to construct point cloud information (Chae et al. 2017). Statistical point aggregation $P = \{p_1, p_2, ..., p_n\}$ consists of n points, and the collection is stored in KD tree for subsequent calls (Zeng et al. 2023). Traverse each sampling point p_i in P, and then calculate the Euclidean distance d_{ij} from other points p_i in P to p_j. According to the size of d_{ij}, Select the k points closest to p_i as the K-neighborhood point set $N_k(p_i) = \{q_1, q_2, ..., q_k\}$ of the sampling point. q_j represents the j TH point in the neighborhood of point p_i, and setting $d_j(p_i)$ represents the Euclidean distance between point p_i and the j TH point in the neighborhood, as shown in Equation (1).

$$d_j\left(p_i\right) = d\left(p_i, q_j\right) = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 + \left(z_i - z_j\right)^2} \tag{1}$$

Wherein, the three-dimensional coordinates of p_i are x_i, y_i, z_i, and the three-dimensional coordinates of q_j are x_j, y_j, z_j. The K-neighborhood density $\rho(p_i)$ of point p_i is introduced. The density of the neighborhood of this point is represented by the inverse of the average distance of the K-neighborhood. The larger the density, the denser the point cloud of the neighborhood of this point. On the contrary, the sparser the point cloud. The schematic diagram of neighborhood density is shown in Figure 3.

The K-neighborhood density $\rho(P_i)$ of the point p_i is introduced, the density (local point concentration) of the neighborhood of the point is expressed by the inverse of the average distance of the K-neighborhood, the larger the density is, the denser the neighborhood point cloud of the point is, and vice versa, the sparser the point cloud is, the algorithm is shown in the schematic in Figure 3, and the formula for the neighborhood density is shown in Equation (2).

$$(P_i) = -\sum \left| p_j - p_i \right|, \quad p_j \in N(p_i) \tag{2}$$

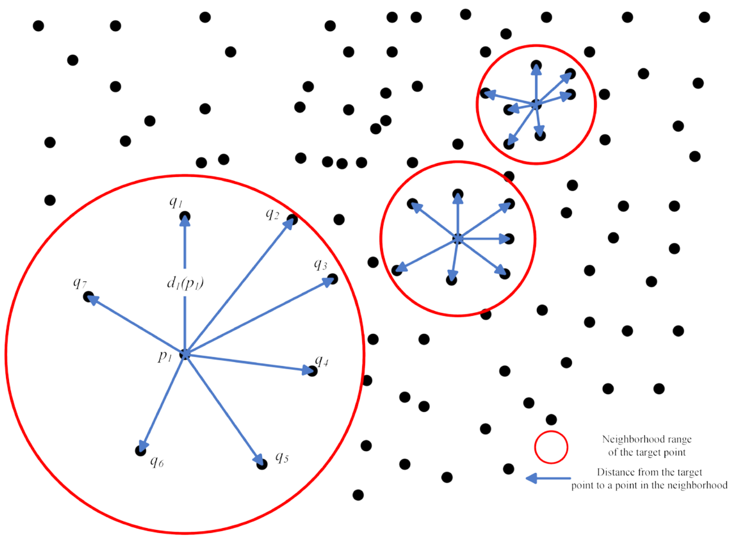

Fig. 3. Schematic of K-neighborhood density

Rys. 3. Schemat gęstości dzielnicy K

1.4. K-value adaptive selection

One of the difficulties of the K-neighborhood filtering method is determining the optimal value of K. A smaller K value will be too sensitive to noise, resulting in a decrease in the original point retention rate. In comparison, a larger K value will blur the local structure. Here, a method based on the neighborhood point cloud density variance is introduced to select a suitable K value. First, the point cloud is divided into two clusters, namely the noise cluster and the retention cluster, by manually setting the point cloud density center point according to the point cloud density through the K-means clustering algorithm.

The appropriate K value should contain a small number of points from another cluster within the neighborhood range of the critical point of a certain cluster so as to separate the critical point from the non-critical point. Under this condition, the K value should be as small as possible to reduce the calculation pressure. In order to determine whether the K neighborhood of the target point contains points from other clusters, the K neighborhood point density variance $\sigma(p_i)$ is introduced, and the calculation formula is shown in Equation (3).

$$\sigma(p_i) = \sqrt{\frac{1}{k}\sum_{j=1}^{k}\left(\rho(q_j) - \rho(p_i)\right)^2} \tag{3}$$

The larger the variance of the K neighborhood density, the more uneven the density distribution in the neighborhood of the point is, indicating that the probability of noise in the neighborhood of the point is greater. For the target point cloud computing K neighborhood density variance distribution map, the maximum value of the neighborhood density variance in the map is set as the reference value K_σ, and then the average density μ_ρ of each cluster is calculated according to Equation (4).

$$\mu_\rho = \frac{1}{|C|}\sum_{i=1}^{C}\rho(p_i) \tag{4}$$

Where C is the density cluster point set after clustering. Finally, the adaptive K value K_i corresponding to each point is calculated according to the above parameters. The calculation formula is shown in Equation (5).

$$K_i = \alpha \cdot \left(\frac{\mu_\rho}{\rho(p_i)}\right) + \beta K_\sigma \tag{5}$$

In the Equation, α and β are linear weights, and finally K_i is rounded down to get the optimal K value for each local point.

After selecting a suitable K value, a method is needed to detect outliers, i.e. noise, and remove them from the point set to complete the filtering operation. Here, the local outlier factor $LOF_k(p_i)$ needs to be extracted by calculating the difference in density between the target point and its neighborhood points. The calculation method expresses the local outlier factor as the ratio of the average density of the neighborhood of the target point to the density of the point. The calculation formula is shown in Equation (6).

$$LOF_k(p_i) = \frac{\sum_{j=1}^{K_i}\rho(q_j)}{K_i} \Big/ \rho(p_i) \tag{6}$$

Set the filtering threshold according to the closeness of the actual point cloud R. When the local outlier factor $LOF_k(p_i)$ is greater than R, it means that the density of the target point is less than the average density of the neighboring points in its neighborhood, and the point is likely to be in the noise cluster. On the contrary, when the local outlier factor is less than R, it means that the density of the target point is greater than the average density of the neighboring points in its neighborhood, and the point is likely to be in the retained cluster. The outlier factor is removed to complete the filtering of the point cloud.

2. Experiments and results analysis

Experiments were conducted on a Windows 10 64-bit system with an Intel Core i5-12400 processor at 2.4 GHz and 16 GB of memory. The algorithm will be implemented by Visual Studio 2019 with C++ programming language and PyCharm 3.2 with Python programming language. The point cloud model is generated from the irregular ore and the point cloud model generated by the scanning of the grid sieve at the mining site captured by the ZED2 camera.

2.1. Outlier visualization experiments

Local outlier visualization experiments are carried out on the noisy point cloud data to support the advantages of a local outlier in representing the degree of outlier and obtaining the adaptive optimal K-value. The number of points in the ore point cloud is 18,809, and the three-dimensional dimensions are: X = 628.83 mm, Y = 901.48 mm, and Z = 196.65 mm. The maximum offset distance between the noisy point cloud and the standard point cloud is 48.12 mm. The average offset distance is 16.039 mm, and the standard deviation of the average offset distance is 8.019 mm. Calculate the value of K according to the algorithm. Then, the outlier factor of each point is identified and its value is visualized using the local outlier factor algorithm. The result is shown in Figure 4. In the figure, the size of the red circle indicates the size of the outlier, and the larger the red circle, the more likely the point is to be an outlier.

The visualized image reveals a significant difference in outlier factors between sparsely and densely distributed point sets. The algorithm proposed in this chapter will adaptively adjust the parameters and accurately identify the outlier points to be rejected to achieve the filtering result through this difference.

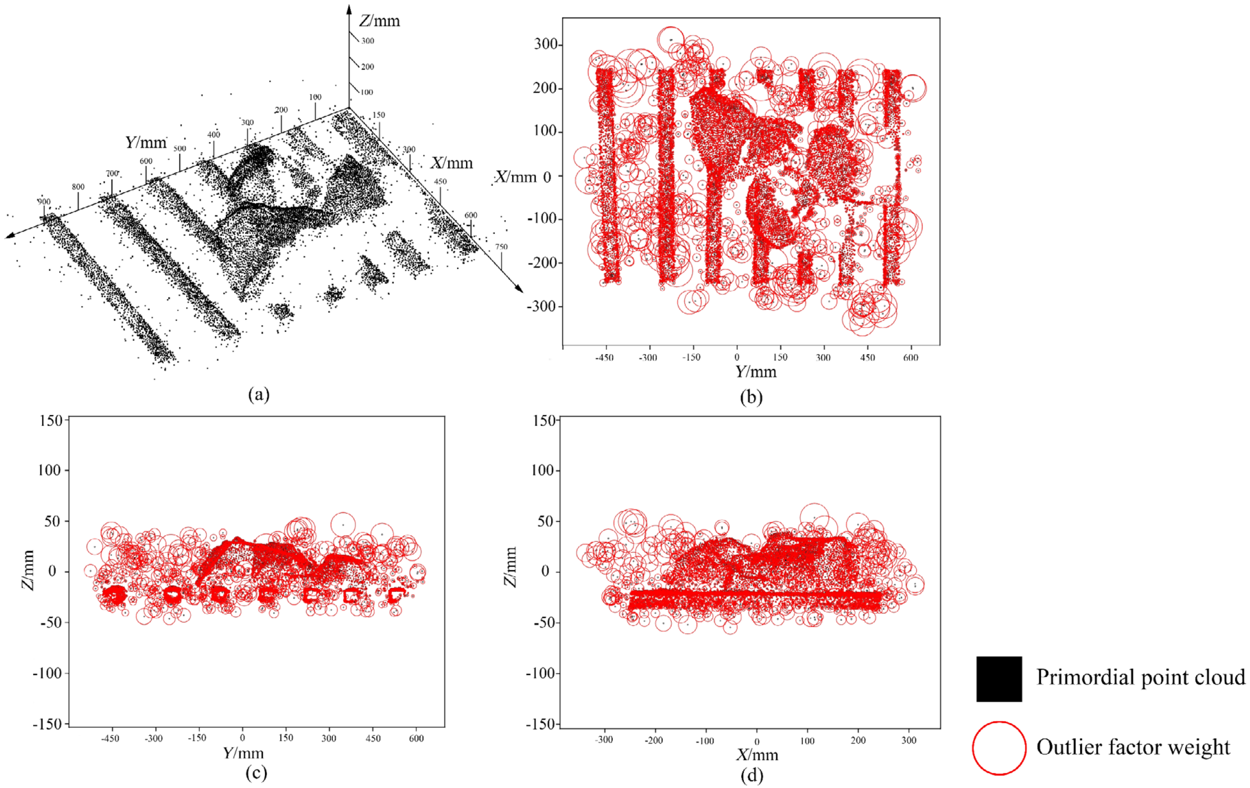

Fig. 4. Ore point cloud and visualization of outlier factors
(a) ore points cloud in the presence with noise, (b) top view of visualized outlier factors,
(c) main view of visualized outlier factors, (d) side view of visualized outlier factors

Rys. 4. Chmura punktów rudy i wizualizacja czynników odstających
(a) chmura punktów rudy w obecności szumu, (b) widok z góry wizualizowanych czynników odstających,
(c) widok główny wizualizowanych czynników odstających, (d) widok z boku wizualizowanych czynników odstających

2.2. Comparison experiment on denoising effect

Set the filter threshold R to 1 and filter the point cloud. In order to verify the effectiveness of the denoising algorithm, the adaptive K-value density filtering algorithm proposed in this paper is compared with the radius filtering algorithm and the statistical filtering algorithm. The same noisy point cloud is filtered by all three denoising algorithms; the experimental results are shown in Figure 5.

Figures 5(b) through (d) display the denoising results for the noisy point cloud using radius filtering, statistical filtering, and K-neighborhood density filtering, respectively. The results indicate that the edges of the point cloud denoised by Radius filtering and Statistical filtering are still not clear enough, and there are scattered outliers around the main body, leading to structural disorder. At the same time the edges of the point cloud denoised by K-neighborhood density filtering are more transparent and smoother, with fewer outliers. In order to quantitatively and objectively assess the denoising effect, three indicators are used as evaluation criteria for denoising performance, namely denoising precision P_r, noise recall rate RE*call*, and origin retention rate RE*tain*. The formulas of the three indicators are shown in Equation (7).

$$\left\{ \begin{array}{l} P_r = \dfrac{TP}{TP + FP} \\ \mathrm{Re}\,call = \dfrac{TP}{TP + FN} \\ \mathrm{Re}\,tain = \dfrac{TN}{TN + FP} \end{array} \right\} \tag{7}$$

TP – represents the number of points that are correctly detected as noise points,
FP – represents the number of points that are actually non-noise points but are incorrectly detected as noise points,
FN – represents the number of points that are actually noise points but are not detected,
TN – represents the number of points that are actually non-noise points and are correctly detected as non-noise points.

As the important indexes of denoising effect, P_r and RE*call* indicate the effect on detection and elimination of outlying noisy points by the algorithm. The higher the value of P_r and RE*call*, the better the denoising effect. The other indicator RE*tain* indicate the effect on retaining the non-noise points. The higher the value of RE*tain*, the more structural features can be preserved without being destroyed. Comparison of the performance of three algorithms is shown in Table 1 Comprehensive performance comparison of different denoising algorithms Table 1.

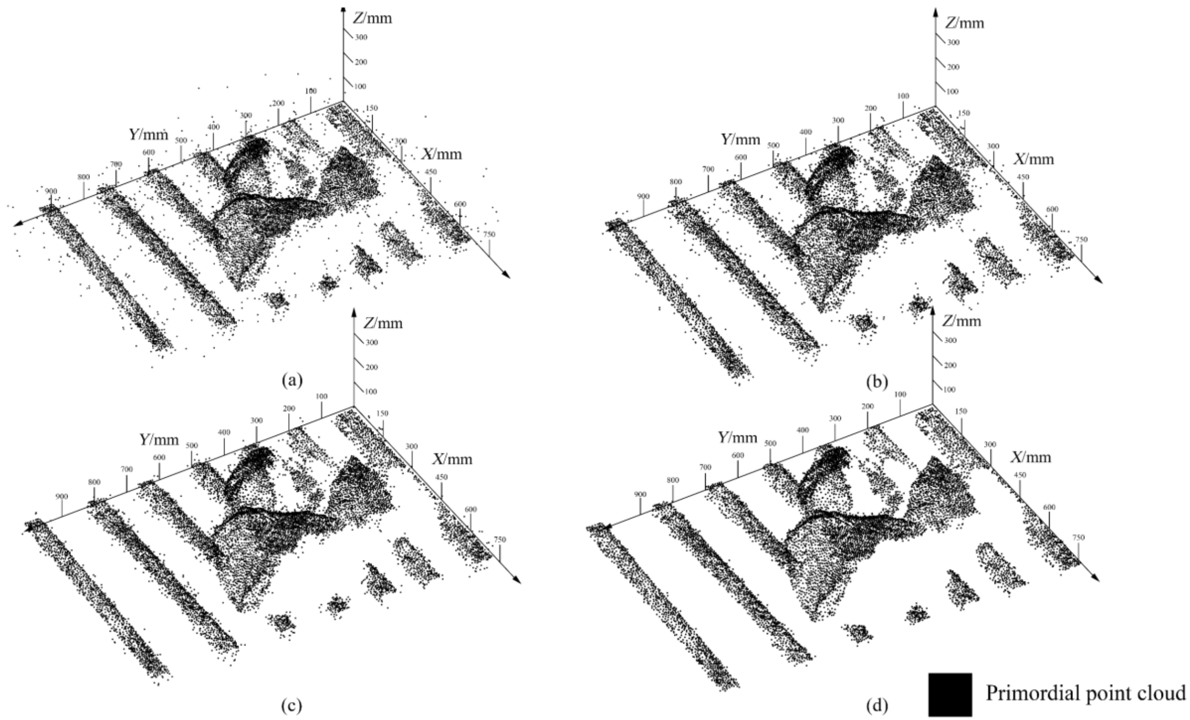

Fig. 5. Comparison of denoising effect by different algorithms
(a) primordial point cloud, (b) radius filtering, (c) statistical filtering, (d) K-neighborhood density filtering

Rys. 5. Porównanie efektu odszumiania za pomocą różnych algorytmów
(a) pierwotna chmura punktów, (b) filtrowanie promieniowe, (c) filtrowanie statystyczne, (d) filtrowanie gęstości K-sąsiedztwa

Table 1. Comprehensive performance comparison of different denoising algorithms

Tabela 1. Kompleksowe porównanie wydajności różnych algorytmów odszumiania

Algorithm	Points numbers	P_r (%)	RE*call* (%)	RE*tain* (%)
Radius filter	18,809	40.62	47.36	92,37
Statistical filter	18,809	68.18	99.91	91.89
K-neighborhood density filtering	18,809	95.68	99.92	94.17

As indicated in Table 1, the original retention rates of the radius filtering and statistical filtering algorithms are relatively high, yet their denoising accuracies are poor. Additionally, the noise recall rate of the radius filtering algorithm merely amounts to 47.36%, signifying that a considerable number of actual noise points remain unremoved. Despite the fact that the recall rate of the statistical filtering algorithm reaches as high as 99.91%, its denoising accuracy is relatively low, standing at 68.18%, suggesting that there exists a significant error during the denoising process. Both these algorithms are influenced by their fixed neighborhood radius and reliance on global statistics, along with the presence of numerous outliers in the ore point cloud, which affects the overall mean and standard deviation, thereby resulting in the deletion of a large number of dense points. In contrast, the K-neighborhood density filtering algorithm proposed in this paper exhibits outstanding performance in all three indicators, with a denoising accuracy of 95.68%, a noise recall rate of 99.92%, and an original retention rate of 94.17%, indicating that this algorithm possesses high precision and reliability in denoising ore point clouds.

Conclusions

To address structural noise removal and smoothing in ore point clouds collected via stereo vision, perceptual weights were designed based on neighborhood density and density variance. The traditional point cloud neighborhood filtering algorithm is improved, and the point cloud adaptive K-neighborhood density filtering algorithm is proposed.

Comparative experiments show that the K-neighborhood density filtering algorithm provides significant improvement in terms of precision and noise recall compared with traditional statistical and radius filtering. The filtering performance of the proposed algorithm better matches the ore point cloud model and supports subsequent tasks such as ore crushing and point cloud processing, demonstrating the practical applicability in the field of ore crushing.

This work was supported by the General Projects of R&D Program of Ganzhou (No. 202101124911).

The Authors have no conflict of interest to declare.

REFERENCES

Axelsson, P. 2000. DEM generation from laser scanner data using adaptive TIN models. *International Archives of Photogrammetry and Remote Sensing* 33(4), pp. 110–117.

Cai et al. 2021 – Cai, M., Li, P., Tan, W. and Ren, F. 2021. Key engineering technologies to achieve green, intelligent, and sustainable development of deep metal mines in China. *Engineering* 7(11), pp. 1513–1517, DOI: 10.1016/j.eng.2021.07.010.

Chae et al. 2017 – Chae, D.K., Lee, S.C., Lee, S.Y. and Kim, S.W. 2017. On identifying k-nearest neighbors in neighborhood models for efficient and effective collaborative filtering. Neurocomputing, 278, pp. 134–143, DOI: 10.1016/j.neucom.2017.06.081.

Chen et al. 2021 – Chen, C., Guo, J., Wu, H., Li, Y. and Shi, B. 2021. Performance comparison of filtering algorithms for high-density airborne LiDAR point clouds over complex LandScapes. *Remote Sensing* 13(14), DOI: 10.3390/rs13142663.

Chen et al. 2023 – Chen, S., Wang, J., Pan, W., Gao, S., Wang, M. and Lu, X. 2023. Towards uniform point distribution in feature-preserving point cloud filtering. *Computational Visual Media* 9(2), pp. 249–263, DOI: 10.1007/s41095-022-0278-4.

Han et al. 2017 – Han, X.F., Jin, J.S., Wang, M.J., Jiang, W., Gao, L. and Xiao, L. 2017. A review of algorithms for filtering the 3D point cloud. *Signal Processing Image Communication* 57, pp. 103–112, DOI: 10.1016/j.image.2017.05.009.

Han et al. 2018 – Han, X.F., Jin, J.S., Wang, M.J. and Jiang, W. 2018. Guided 3D point cloud filtering. *Multimedia Tools and Applications* 77, pp. 17397–17411, DOI: 10.1007/s11042-017-5310-9.

Jia et al. 2019 – Jia, C.C., Wang, C.J., Yang, T., Fan, B.H. and He, F.G. 2019. A 3D point cloud filtering algorithm based on surface variation factor classification. *Procedia Computer Science* 154, pp. 54–61, DOI: 10.1016/j.procs.2019.06.010.

Li et al. 2019 – Li, X., Chen, Z., Chen, J. and Zhu, H. 2019. Automatic characterization of rock mass discontinuities using 3D point clouds. *Engineering Geology* 259, DOI: 10.1016/j.enggeo.2019.05.008.

Li et al. 2023 – Li, Y., Peng, P., Li, H., Xie, J., Liu, L. and Xiao, J. 2023. Drilling Path Planning of Rock-Drilling Jumbo Using a Vehicle-Mounted 3D Scanner. *Sustainability* 15(12), DOI: 10.3390/su15129737.

Lu et al. 2020 – Lu, D., Lu, X., Sun, Y. and Wang, J. 2020. Deep feature-preserving normal estimation for point cloud filtering. *Computer-Aided Design* 125, DOI: 10.1016/j.cad.2020.102860.

Moreno, C. and Li, M. 2016. A comparative study of filtering methods for point clouds in real-time video streaming. *Proceedings of the World Congress on Engineering and Computer Science* 1, pp. 388–393).

Pingel et al. 2013 – Pingel, T.J., Clarke, K.C. and McBride, W.A. 2013. An improved simple morphological filter for the terrain classification of airborne LIDAR data. *ISPRS Journal of Photogrammetry and Remote Sensing* 77, pp. 21–30, DOI: 10.1016/j.isprsjprs.2012.12.002.

Wang et al. 2022 – Wang, J., Jiang, J., Lu, X. and Wang, M. 2022. Rethinking point cloud filtering: A non-local position based approach. Computer-Aided Design, 144, DOI: 10.1016/j.cad.2021.103162.

Weales, D. 2018. *Design of a Bin-picking System for Large Objects with High Clutter and Occlusion* (Doctoral dissertation, University of Guelph).

Yang et al. 2024 – Yang, L., Yang, H., Liu, Y. and Cao, C. 2024. Stereo matching algorithm for mineral images based on improved BT-Census. *Minerals Engineering* 216, DOI: 10.1016/j.mineng.2024.108905.

Zeng et al. 2023 – Zeng, N., Li, J., Zhang, Y., Gao, X. and Luo, L. 2023. Scattered Train Bolt Point Cloud Segmentation Based on Hierarchical Multi-Scale Feature Learning. *Sensors* 23(4), DOI: 10.3390/s23042019.

Zeybek, M. and Şanlıoğlu, İ. 2019. Point cloud filtering on UAV based point cloud. *Measurement* 133, pp. 99–111, DOI: 10.1016/j.measurement.2018.10.013.

Zhang et al. 2016 – Zhang, W., Qi, J., Wan, P., Wang, H., Xie, D., Wang, X. and Yan, G. 2016. An easy-to-use airborne LiDAR data filtering method based on cloth simulation. *Remote sensing* 8(6), DOI: 10.3390/rs8060501.

ADAPTIVE ORE POINT CLOUD FILTERING ALGORITHM BASED ON THE K-NEAREST NEIGHBOR

Keywords

stereo vision, ore point cloud, filtering, K nearest neighbors

Abstract

A robotic arm can determine the crushing center of ore using point clouds reconstructed by a binocular camera. However, noise in the original point cloud creates ambiguity, complicating the determination process. To address this, an efficient noise filtering and smoothing algorithm for point clouds is proposed. First, the topological relationships among the point clouds are established using a K-D tree, enabling neighborhood selection and query for each point. The density and density variance for each neighborhood are then calculated via the K-nearest neighbor density filtering method. Clustering is applied to determine the average density, and the optimal K value is adaptively obtained based on both the density variance and cluster densities with assigned weights. The local outlier factor is subsequently calculated using this K value, and noise points are filtered out by setting an outlier factor threshold. Based on the enhanced K-nearest neighbor density filtering algorithm, the experimental results demonstrate that this method achieves a denoising precision of 95.68%, representing an improvement of 55.06% over the traditional Radius filtering method and 27.5% over the statistical filtering method. Additionally, the noise recall rate reaches 99.92%, and the original retention rate is 94.17%, showcasing superior filtering performance while preserving data integrity. These advancements provide a reliable technical foundation for subsequent ore crushing and point cloud data processing tasks.

ADAPTACYJNY ALGORYTM FILTROWANIA CHMURY PUNKTÓW RUDY OPARTY NA NAJBLIŻSZYCH SĄSIADACH K

Słowa kluczowe

widzenie stereoskopowe, chmura punktów rudy, filtrowanie, K najbliższych sąsiadów

Streszczenie

Manipulator może określić środek kruszenia rudy za pomocą chmur punktów rekonstruowanych przez kamerę binokularną. Jednak szumy w oryginalnej chmurze punktów powodują niejednoznaczność, co komplikuje proces określania środka kruszenia. Aby temu zaradzić, zaproponowano wydajny algorytm filtrowania i wygładzania szumów w chmurach punktów. Najpierw za pomocą drzewa K-D ustalane są relacje topologiczne między punktami chmury, co umożliwia wybór i zapytania dotyczące sąsiedztwa każdego punktu. Następnie dla każdego sąsiedztwa obliczana jest gęstość i wariancja gęstości za pomocą metody filtrowania gęstości najbliższych sąsiadów K. Zastosowano klasteryzację w celu określenia średniej gęstości, a optymalna wartość K jest adaptacyjnie uzyskiwana na podsta-

wie zarówno wariancji gęstości, jak i gęstości klastrów z przypisanymi wagami. Następnie obliczany jest lokalny współczynnik odstępstwa przy użyciu tej wartości K, a punkty szumów są odfiltrowywane poprzez ustawienie progu współczynnika odstępstwa. Na podstawie udoskonalonego algorytmu filtrowania gęstości najbliższych sąsiadów K wyniki eksperymentów wykazują, że metoda ta osiąga precyzję odszumiania na poziomie 95,68%, co stanowi poprawę o 55,06% w porównaniu z tradycyjną metodą filtrowania promienia oraz o 27,5% w porównaniu z metodą filtrowania statystycznego. Ponadto wskaźnik odzyskiwania szumów osiąga 99,92%, a wskaźnik zachowania oryginalnych danych wynosi 94,17%, co pokazuje doskonałą wydajność filtrowania przy jednoczesnym zachowaniu integralności danych. Te osiągnięcia stanowią niezawodną podstawę techniczną do dalszych zadań związanych z kruszeniem rudy oraz przetwarzaniem danych chmury punktów.