GOSPODARKA SUROWCAMI MINERALNYMI – MINERAL RESOURCES MANAGEMENT

2026 Volume 42 Issue 1 Pages 45–68

DOI: 10.24425/gsm.2026.158279

LIRONG YANG[1], XIAOLONG ZHU[2], ZHIWEN HUANG[3], CHONG CAO[3]

Ore crushing position study method based on stacked ore point cloud instance segmentation

Introduction

In the context of mineral processing, the raw ore bin is equipped with a grizzly screen, a device utilized for the purpose of sifting large ores and averting their entry into the rough crushing equipment. This is achieved by mitigating the risk of jamming or damage to the equipment. The extraction of ore from the grizzly screen typically involves the use of artificial hammer crushing or manual manipulation of mechanical arm crushing. These methods, however, are characterized by low crushing efficiency, high labor intensity, and significant safety hazards. Therefore, it is necessary to study a method of segmenting the stacked ores by instances, locating the crushing position of the stacked ores, and thus realizing the autonomous crushing of the ores by the robotic arm.

✉ Corresponding Author: Lirong Yang; e-mail: 2168504069@qq.com

[1] Jiangxi Mining and Metallurgical Engineering Research Center; School of Mechanical and Electrical Engineering, Jiangxi University of Science and Technology, China; e-mail: 2168504069@qq.com

[2] School of Mechanical and Electrical Engineering, Jiangxi University of Science and Technology, China; ORCID iD: 0009-0006-5854-175X; e-mail: 2774159670@qq.com

[3] School of Mechanical and Electrical Engineering, Jiangxi University of Science and Technology, China

At present, the field of point cloud segmentation is predominantly divided into two primary categories: traditional point cloud segmentation algorithms and deep learning point cloud segmentation algorithms. Traditional point cloud segmentation algorithms can be classified into two primary categories: methods based on geometric features and methods based on feature clustering. Common point cloud segmentation methods based on geometric features include region growth, curvature thresholding, and normal vector and other direction-based approaches. In addressing this challenge, Wang Jingxue proposes a voxel region growth-based point cloud building roof surface segmentation method, which has been shown to enhance the accuracy of roof surface segmentation, particularly in complex structures (Wang et al. 2023). Zhi-Bin Zhao employs the LCCP region growing segmentation algorithm to achieve the segmentation of part point clouds based on point cloud features (Zhao 2023). In the following study, Ma Xuelei et al. present an approach to accurately segmenting sheep body point clouds that employs principal component analysis, a random sampling consistency algorithm, and an improved region growing method (Ma et al. 2020). Zhao et al. proposed a point cloud segmentation algorithm based on multi-feature training and weighted random forest (RF) to improve the classification accuracy of point clouds and achieve accurate segmentation of targets or scenes (Zhao et al. 2025). A. Dreier et al. proposed a 3D point cloud-based instance segmentation algorithm with reflection intensity information for identifying individual stones in rubble masonry dams that can be used for deformation monitoring (Dreier et al. 2025). Another type of feature clustering-based approach involves the segmentation of point cloud models as a classification process of data points with geometric feature parameters. This approach involves dividing a large number of point clouds into multiple subclasses so that the data in the classes have the same or similar features, while the data points between the classes are relatively independent. Longqiang Guo has developed a technique that integrates visual projection with point cloud clustering to achieve effective segmentation of target point clouds for measuring object volume (Guo et al. 2024). Junchao Zhu's advancement in point cloud clustering algorithms involves the integration of dual criteria, namely distance and angle, to enhance the recognition and definition of object boundaries in three-dimensional space. This innovation is poised to improve the accuracy and efficacy of point cloud analysis, a crucial step in numerous applications, including but not limited to robotics, automation, and geospatial analysis (Zhu et al. 2023). Huanghe's innovative Kersey's Inverse Vulture Search algorithm is predicated on a "frantic hunting" strategy. This strategy is employed to optimize K-means clustering, thereby enhancing the grouping efficiency and adaptability of point cloud data (Huang et al. 2023). Fang et al. advanced a novel framework, BGPSeg, for point cloud primitive instance segmentation (Fang et al. 2025). This framework employs boundary-guided feature extraction and clustering.

Methods for deep learning point cloud segmentation can be categorized primarily as point convolution or graph convolution. Convolution point: the PointNet network is a pioneering work in directly processing point cloud features. It has provided inspiration for subsequent point cloud research efforts (Qi et al. 2017a). PointNet++ overcomes the limitations of

PointNet by enabling the network to more effectively capture the local and global features of point clouds (Qi et al. 2017b). PointConv is a point cloud segmentation method based on convolutional neural networks. It uses the powerful learning capability of multilayer perceptrons to simulate continuous 3D convolutional kernels and dynamically adjust weight coefficients (Wu et al. 2020). KPConv is a point cloud segmentation method based on deformable convolutions. The network is inspired by 2D image variability convolutions and the design of core point convolutions that skip multilayer perceptrons to learn weight parameters (Hugues et al. 2019). Many scholars have applied point convolution to object segmentation and obtained significant results. Among them is Wenjie Yang, who innovated the PointNet++ algorithm and proposed a point cloud segmentation technique specialized in the fine structure of transmission lines (Yang et al. 2024). In his proposal, Wenguang Wu sets forth a semantic segmentation method for convex features on unpaved roads. This method is designed to effectively classify convex surfaces of varying scales and height differences (Wu et al. 2023). Cheng, Xin, et al., in a seminal study, developed a 3D target detection model, termed FDG-PointNet, which was based on the F-PointNet network. This model was distinguished by its innovative integration of a dense connectivity strategy and Gaussian distance features. The primary objective of this model was to facilitate the detection of pedestrians on roads (Cheng et al. 2024). Li et al. proposed a point cloud segmentation network based on a point convolutional coding and decoding structure. This structure extracts local features of a point cloud through density-position adaptive convolution. This process fuses density information and the positional relationship between points (Li et al. 2025). Du et al. advanced the argument that semantic segmentation can be regarded as a foundational task for instance segmentation. In response, they developed a point cloud instance segmentation network, designated as MTCloud, which is based on a multi-type convolutional linkage algorithm (Du et al. 2025). Daif and Marzouk developed a comprehensive framework for the classification and segmentation of point clouds for structural steel elements (Daif and Marzouk 2025). The objective of this framework is to enhance the efficiency of artificial intelligence (AI) models in processing point cloud datasets. This approach has been shown to improve the accuracy of the PointNet algorithm. Shahraki et al. proposed a hybrid 3D segmentation model, SAMNet++, which combines the generalization of SAM with the local feature extraction of PointNet++ (Shahraki et al. 2025). This combination is intended to optimize PointNet++ for improved accuracy.

Graph convolution: graph encoding is a process that converts the global coordinates of point cloud features to relative coordinates. It employs edge convolution to gather semantic information from edge features. Furthermore, it continuously updates the feature graph, which contains feature information. DGCNN is a graph convolutional neural network (GCNN) that has been developed for the classification and segmentation of point clouds. It is based on PointNet, and it employs edge convolution. In this method, the graph concept is applied to point clouds. This involves replacing point features with edge features and dynamically aggregating local geometric features (Wang et al. 2019). RSNet is a deep learning-based point cloud segmentation method that has been demonstrated to capture

details and global features of point clouds more effectively by introducing residual structures and null convolution (Huang et al. 2018). The PointSIFT network employs the K-nearest neighbor (KNN) algorithm to explore the set of nearest neighbor points in a single direction, thus failing to leverage the spatial advantages of the 3D point cloud. To address this limitation, the SIFT algorithm was utilized to construct a direction encoding unit with eight direction searching capabilities for the point cloud segmentation task (Jang et al. 2018). Morteza's proposal entails the introduction of a novel pattern-based deep neural network, designated as Pattern-Net. This innovative network possesses the capacity to analyze unstructured and complex point clouds, effectively decomposing them into semantically meaningful components. In a notable application, Pattern-Net has been successfully employed to segment point clouds of wheat, showcasing its efficacy in processing diverse data sets (Ghahremani et al. 2021).In their study, Andy Zhu et al. proposed a method for screening and enriching fusion features. This method utilizes the CurveNet network, with the objective of enhancing the recognition and segmentation of point clouds. The efficacy of this approach has been demonstrated, resulting in a significant advancement in the field (Zhu et al. 2024). Fedor Zolotarev has proposed a novel point cloud segmentation technique based on Point Transformer that learns to identify points belonging to a logarithmic surface in an unsupervised manner (Zolotarev et al. 2025). Huang et al. advanced a theory positing that CL2M acquires more precise features of point clouds at varying locations by means of a self-attention mechanism. They further propose the incorporation of a contrast learning method to enhance the precision of point cloud boundaries (Huang et al. 2025).

The aforementioned multiple segmentation architectures of the segmentation network, although certain algorithms can effectively segment the ores from the lattice sieve, the segmented results are unable to distinguish each individual ore entity due to the physical contact and overlap between the ores. To address this challenge, the present study proposes an instance segmentation method based on the concave and convex feature attributes of the point cloud. This approach aims to ensure the accurate segmentation of point cloud data for each ore. In the context of a single ore point cloud, a surface reconstruction of the ore point cloud is subsequently executed. The location of the ore center of mass is then determined based on the geometric information of the reconstructed surface. Finally, a random sampling consistency algorithm is utilized to detect the planar region of the ore point cloud, to screen out the most suitable planes for crushing operations, and to determine the specific crushing position accordingly.

1. Research on segmentation method of stacked ore point cloud instances

1.1. Ore point cloud voxelization and seed voxel selection

The raw point cloud data contains noise that may cause boundary ambiguity and segmentation errors if processed directly. To mitigate these noise effects, voxelization preprocessing is required prior to supervoxel clustering.

The voxelization process comprises three sequential steps:

1. Boundary determination through extraction of the maximum and minimum coordinate values from the point cloud data to construct an axis-aligned bounding box (AABB).
2. Establishment of a voxel coordinate system aligned with the X, Y, and Z axes, using the minimum boundary point as origin and applying a predefined voxel resolution R_{voxel} as the unit length.
3. Construction of a three-dimensional uniform grid within this coordinate system, generating voxel cells whose centroids approximate the contained point clusters.

The voxelization process constructs an adjacency structure among voxels within the discretized 3D space, effectively preventing supervoxels from spanning discontinuous object boundaries. This study employs a 26-connectivity neighborhood structure to enhance segmentation accuracy, where adjacency relationships are efficiently determined through k-d tree accelerated neighbor searches. Figure 1 demonstrates the voxelization results of the ore point cloud dataset.

Following voxelization, seed voxels are selected as the initial step for instance segmentation. To achieve effective hyperbody clustering, our method ensures uniform and comprehensive coverage of the entire point cloud through rasterization with an optimized spatial resolution R_{seed}.

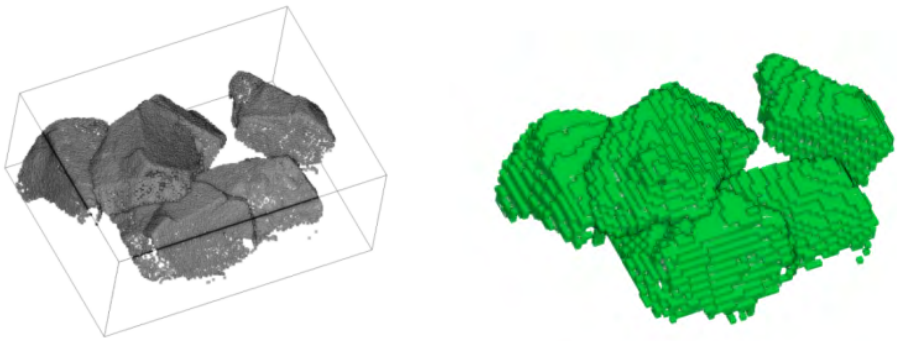


Fig. 1. Ore point cloud mineralization

Rys. 1. Mineralizacja chmury punktów rudy

1.2. Ore point cloud superstitial clustering

The clustering process initiates from seed points and progressively expands to adjacent voxels. Through precise control of clustering conditions and boundary definitions, the entire voxel space becomes partitioned. During expansion, the algorithm continuously incorporates neighboring voxels by evaluating inter-voxel similarity, thereby grouping highly similar voxels together. This similarity-driven clustering approach ensures both semantic consistency and geometric accuracy in the resulting clusters, establishing a robust foundation for subsequent point cloud segmentation. The similarity judgment between voxels is shown in Equation 1.

$$D = \sqrt{w_c D_c^2 + \frac{w_s D_s^2}{2R_{seed}^2} + w_n D_n^2} \tag{1}$$

- D_n – indicates normalized differences,
- D_s – denotes the European distance difference,
- w – denotes the weight parameter, which is used to control clustering.

D_c the formula for this is shown in Equation 2.

$$D_c = \sqrt{(L_2 - L_1)^2 + (a_2 - a_1)^2 + (b_2 - b_1)^2} \tag{2}$$

- D_c – European distance of the CIELAB color space,
- L – the degree of lightness or darkness of a color,
- a – the contrast between the degree of redness and the degree of greenness of a color.
- b – the contrast between the degree of yellow and the degree of blue of a color.

D_n is the normal variance calculation formula shown in Equation 3.

$$D_n = \frac{(nx_1 \cdot nx_2 + ny_1 \cdot ny_2 + nz_1 \cdot nz_2)}{\sqrt{nx_1^2 + ny_1^2 + nz_1^2}\sqrt{nx_2^2 + ny_2^2 + nz_2^2}} \tag{3}$$

where the direction vector normal to the two points is (nx_1,ny_1,nz_1), (nx_2,ny_2,nz_2).

The clustering algorithm employs breadth-first traversal (BFT) with a layer-wise expansion strategy, where all points at the current traversal level are processed before proceeding to the next level. During this process, each voxel may be accessed by multiple seed voxels. In such cases, the seed voxel exhibiting the minimum Euclidean distance to the target voxel is selected as the clustering center for that voxel.

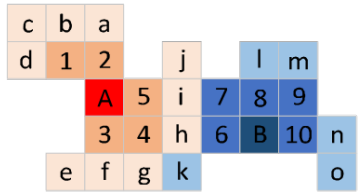


Fig. 2. Schematic diagram of neighboring elements

Rys. 2. Schematyczny rysunek sąsiadujących elementów

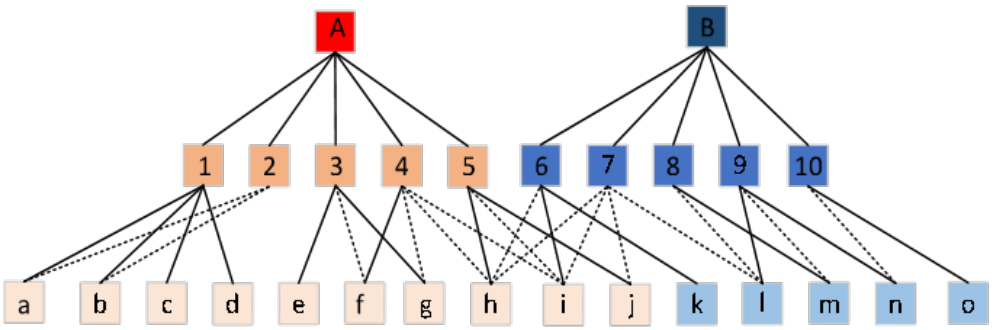


Fig. 3. Schematic diagram of somatostatin clustering

Rys. 3. Schematyczne przedstawienie skupisk somatostatyny

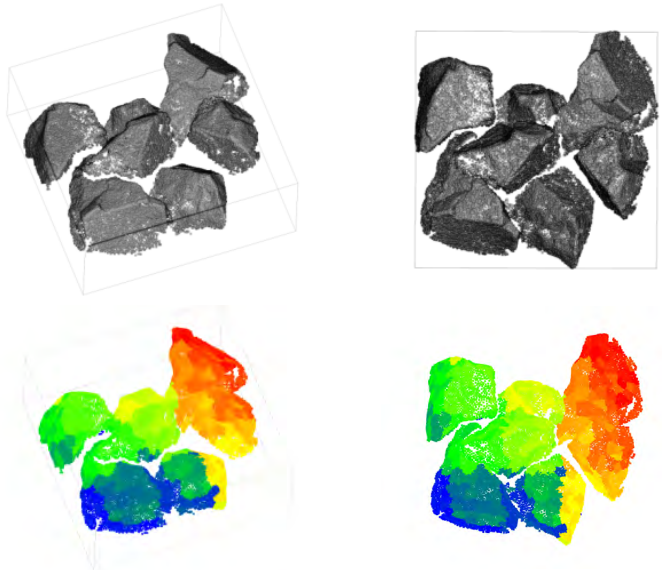


Fig. 4. Stacked ore 3D point cloud superbody clustering results

Rys. 4. Wyniki grupowania superciała chmury punktów 3D ułożonego rudy

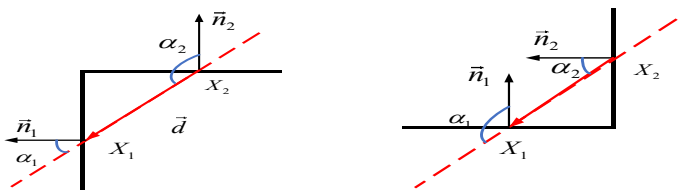
The voxel distribution is illustrated in Figure 2, where voxels *A* and *B* serve as seed points for their respective clusters. When performing neighborhood expansion centered on *A* and *B*, the first-order neighboring voxels of *A* are {1, 2, 3, 4, 5}, while its second-order neighbors comprise {*a*, *b*, *c*, *d*, *e*, *f*, *g*, *i*, *j*, *h*, *k*}. Similarly, voxel *B* exhibits first-order neighbors {6, 7, 8, 9, 10} and second-order neighbors {*k*, *n*, *m*, *l*, *o*, *i*, *j*, *h*}. Notably, voxels {*k*, *h*, *i*, *j*} reside in the junction region between clusters. For these boundary voxels, we compute their Euclidean distances to both seed points using the aforementioned distance metric to determine cluster affiliation. Figure 3 presents the tree-structured representation of these neighborhood relationships, demonstrating the breadth-first search (BFS) traversal process. The final clustering results are visualized in Figure 4.

As illustrated in Figure 4, the point cloud regions exhibiting high similarity have successfully completed clustering and are represented by different colors. This indicates that the voxel has been effectively clustered using the superbody clustering method. The point cloud regions that are illuminated by the same color are characterized by the presence of analogous normal directions, analogous colors, and analogous Euclidean distances.

1.3. Establishment of a criterion for judging inter-voxel concavity

The point cloud data that have been clustered by the supersomes have been shown to possess significant concavity and convexity characteristics. The concave and convex parts can be segmented by judging the relationship between voxels, but a strict similarity judgment criterion should be formulated. The criterion is shown in Figure 5.

The inter-voxel concavity judgment is shown in Figure 5, assuming that the centers of mass of two connected voxel surfaces are X_1 and X_2, the normal directions are n_1 and n_2, and X_1 and X_2 form a vector of $\vec{d} = \dfrac{X_1 - X_2}{\|X_1 - X_2\|}$. The angle between the normal vector and the vector d is α_1 and α_2. The cosine of the angle is found by calculating the cosine of the angle, which is shown in Equation 4.



（a）convex link （b）concave link

Fig. 5. Convexity judgment of voxel blocks

Rys. 5. Ocena wypukłości bloków wokselowych

$$\begin{cases} \cos\alpha_1 = \overline{n}_1 \cdot \overline{d} \\ \cos\alpha_2 = \overline{n}_2 \cdot \overline{d} \end{cases} \tag{4}$$

The formula for determining whether there is a convex or concave link between the hyperbodies is shown in Equation 5.

$$\left(\vec{n}_1 - \vec{n}_2\right) \cdot \vec{d} \tag{5}$$

In the voxel connection relationship, convex connections correspond to positive values and concave connections correspond to negative values. In principle, when a surface is sufficiently smooth, it is possible to determine the convexity or concavity of a surface with a high degree of accuracy based solely on the angular relationship. In practice, however, surfaces that are ideally smooth are seldom encountered, and the point cloud data frequently exhibits clear unevenness. If the judgment is based exclusively on the angular threshold, small depressions on the same plane will be misidentified as independent objects. It is therefore necessary to introduce a segmentation threshold, $\beta_{Threshold}$, in order to avoid the possibility of over-segmentation. In the event of the angular difference between α_1 and α_2 being less than $\beta_{Threshold}$, the region is still considered to be a convex region. The segmentation threshold formula is shown in Equation 6.

$$\beta = \left|\alpha_1 - \alpha = \cos^{-1}\left(\vec{n}_1 \cdot \vec{n}_2\right)\right| < \beta_{Threshold} \tag{6}$$

The resulting formula for the concavity judgment criterion is shown in Equation 7.

$$f\left(\vec{p}_i, \vec{p}_j\right) = \left(\vec{n}_1 - \vec{n}_2\right) \cdot \vec{d} > 0 \cup \beta < \beta_{Threshold} \tag{7}$$

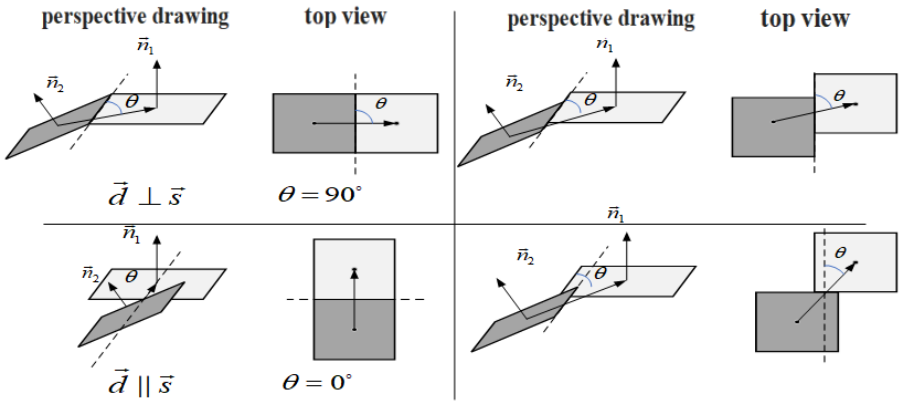


Fig. 6. Schematic diagram of incomplete linkage between voxels

Rys. 6. Schematyczny diagram niekompletnego połączenia między wokselami

In real three-dimensional scenes, a special case of hypervoxel connection exists: two neighboring hypervoxels are not connected by continuous smooth transition surfaces, but form topological associations only through discrete singularities (see Schematic 6). This discontinuous surface property invalidates the traditional concavity criterion based on the curvature continuity assumption. The paper proposes a new kind of criterion based on the topological characteristics of singularities, with s being the cross product of n_1 and n_2, d being the center-of-mass vector, and θ being the angle between the voxel connecting line and the center-of-mass connecting line. The paper hypothesizes that when θ is closer to 90, it is more likely that the two voxels are in the same region. θ is computed as shown in Equation 8.

$$\theta\left(\vec{p}_i,\vec{p}_j\right)=\min\left(\angle\left(\vec{d},\vec{s}\right),\angle\left(\vec{d},-\vec{s}\right)\right) \tag{8}$$

The introduction of a threshold θ_{Thresh} leads to the determination of non-connected voxels when $\theta < \theta_{Thresh}$. In the context of a super voxel, the angle of entrapment is integrated into the threshold setting, and the segmentation threshold is calculated using Equation 9.

$$\theta_{Threshold}(\beta)=\theta_{Threshold}^{\max}\left(1+\exp\left[-a\cdot\left(\beta-\beta_{off}\right)\right]\right)^{-1} \tag{9}$$

β is the normal vector pinch angle, and taking $\theta_{Threshold}^{\max}=60$, $\beta_{off}=0.25$, $a=25$ yields the judgment criterion shown in Equation 10.

$$g\left(\vec{p}_i,\vec{p}_j\right)=\theta\left(\vec{p}_i,\vec{p}_j\right)>\theta_{Threshold}(\beta) \tag{10}$$

Combining the concavity judgment criterion, the final concavity judgment criterion is obtained as shown in Equation 11.

$$conv\left(\vec{p}_i,\vec{p}_j\right)=f\left(\vec{p}_i,\vec{p}_j\right)\cap g\left(\vec{p}_i,\vec{p}_j\right) \tag{11}$$

1.4. Region growing segmentation based on concave-convex judgment criterion

Following the voxel clustering process, which yielded hypervoxel blocks demonstrating high levels of similarity, this study employs the region growing algorithm for the purpose of achieving precise delineation of concave-convex regions in the ore point cloud. This is illustrated in Figure 7. The process is comprised of the following steps:

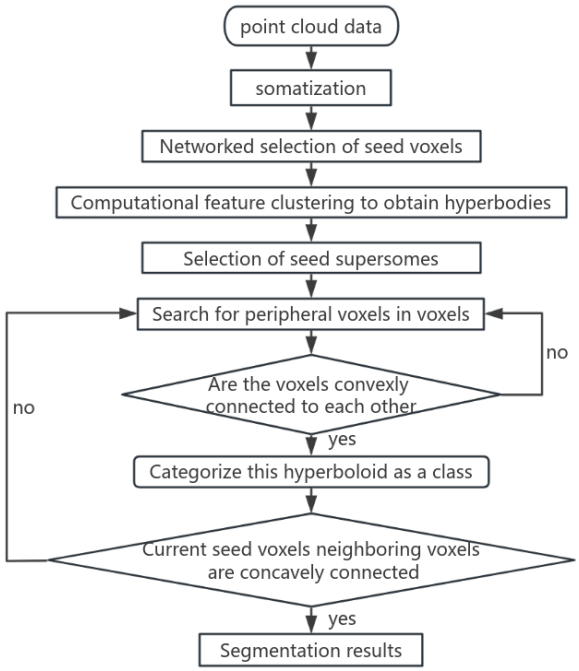


Fig. 7. Flow chart of regional growth segmentation

Rys. 7. Schemat segmentacji wzrostu regionalnego

1. The process of voxelization is defined as follows:

The original ore point cloud data are firstly voxelized and discretized into a regular 3D voxel grid structure. This step establishes a unified spatial reference frame for subsequent processing.

2. The selection of the seed point is a crucial stage in the process.

The selection of seed points is intended to ensure the uniform and comprehensive coverage of the entire point cloud, with the result that the final segmentation effect is directly influenced by the choice of these initial points.

3. Hypervoxel clustering.

The implementation of the similarity measure formula leads to the formation of larger supervoxel blocks through the aggregation of neighboring voxel blocks that exhibit analogous features. The process is carried out in an iterative manner, gradually expanding the size of the hyper voxels.

4. The process of convexity-based region growing is outlined herewith.

In the clustering process, the concavity-convexity property of voxel junctions is utilized as a growth control condition: when neighboring voxels demonstrate a convex connection, the expansion of the current region is continued. In the event of the identification of a concave boundary, the process of growth in the current region should be brought to a conclusion.

Through the above steps, a region growing algorithm is used to segment the ore point cloud. The algorithm first merges the voxels with convex connection characteristics, and automatically stops the growth process when it encounters the concave boundary, and finally realizes the instance segmentation of the ore point cloud. As shown in Figure 8, the segmentation results not only clearly distinguish the concave and convex characteristics of the ore point cloud, but also completely present the whole process of regional growth and its final effect. Experiments show that this method can effectively realize the accurate segmentation of the ore point cloud.

Fig. 8. LCCP region growth segmentation process and results

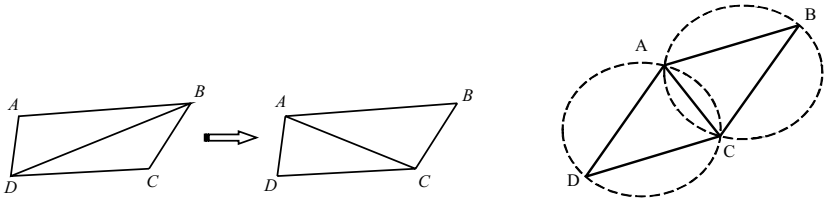
Rys. 8. Proces segmentacji wzrostu regionu LCCP i wyniki

2. Research on the method of calculating the center of mass of ore point cloud

2.1. Ore point cloud triangulation for surface reconstruction

The growth segmentation algorithm extracts individual ore point clouds from the ore point cloud data, and the resulting ore point clouds contain noise and holes due to errors in the segmentation and acquisition process. To improve the accuracy of subsequent steps, such as center of mass calculation, surface reconstruction of the ore point cloud is performed to fill the gaps and remove the noise.

The Delaunay triangular dissection method is utilized to establish a connection between a specified set of points, thereby forming a triangular mesh. The method ensures that the outer circle of the triangle formed by any two points in the triangular dissection does not contain any other points. This leads to the Delaunay optimization criterion, which optimizes the triangular mesh by maximizing the minimum angle. This effectively reduces the number of sharp-angled triangles and thus improves the overall mesh quality. It is guaranteed that the minimum angle of all triangles is greater than or equal to some given value, as demonstrated in Figure 9(a). In order to maintain the empty circle property, it is necessary to demonstrate that no other point exists within the outer circle of any triangle in the Delaunay triangle mesh. In the event of such a point existing, it should be connected to the corresponding vertex. This will ensure the uniqueness of the Delaunay triangles, as illustrated in Figure 9(b).



(a) Minimum angle maximum criterion (b) Null-circle characteristics

Fig. 9. Delaunay triangular optimization criterion

Rys. 9. Kryterium optymalizacji trójkątów Delaunaya

The point cloud data is then reconstructed using Delaunay triangulation, which is triangulated using the Bowyer-Watson algorithm. This exploits the properties of Delaunay to add one point at a time to the Delaunay mesh, using a triangle as an initial, and deleting those triangle cells whose outer sphere contains the new point, forming the Delaunay nulls. The establishment of new points connected to the vertices of the Delaunay void, resulting in the formation of new edges, is contingent upon the satisfaction of the Delaunay triangular splitting rule. This ensures that the current triangular mesh is guaranteed to be optimal within the local range. The following steps are to be taken in the implementation of the algorithm:

1. The initial step in this process is to generate a triangle of sufficient size, which is referred to as a super-triangle. It is imperative to ensure that the input vertices are completely contained within the outer circle of the triangle, thereby guaranteeing that they are all contained within the confines of the triangle.
2. It is imperative that each point in the set is processed in the correct order. For the current insertion point, it is necessary to identify which triangles in the current triangulation contain that point in their outer circles, and to remove those triangles from the triangulation by calling them illegal triangles.

3. The elimination of illegal triangles results in the formation of a polygonal void. The edges that are double-counted (i.e., edges that are shared by two neighboring removed triangles) must be removed for the determination of the void's boundary. The final remaining edge thus constitutes the boundary of the polygonal void.
4. The formation of new triangles is achieved by establishing connections between the current point pair and the boundary point pair of the polygonal void. This process results in the void being filled, while the Delaunay property of the triangular dissection is preserved. Steps 2 through 4 should be repeated until all points have been inserted into the triangle dissection.

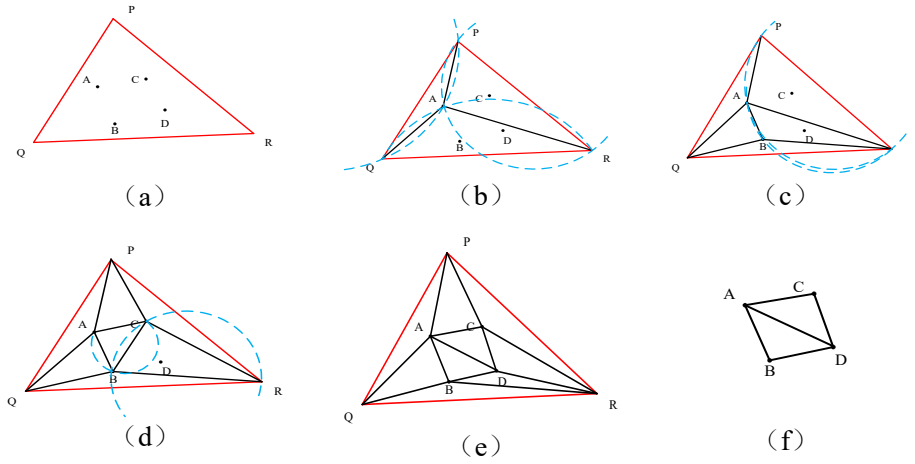


Fig. 10. Triangulated surface reconstruction process

Rys. 10. Proces rekonstrukcji powierzchni trójkątnej

(a) Original ore point cloud

(b) Side view of the ore model after surface reconstruction

(c) Top view of the ore model after surface reconstruction

(d) Front view of the ore model after surface reconstruction

Fig. 11. Triangulated surface reconstruction

Rys. 11. Rekonstrukcja powierzchni trójkątnej

5. Subsequent to the insertion of all points, it is imperative to remove all triangles that contain supertriangle vertices, as these triangles do not belong to the true Delaunay triangular dissection of the original point set.

As demonstrated in Figure 10, the triangulation process is illustrated by a two-dimensional planar schematic.

The triangular mesh reconstruction of the ore point cloud using the aforementioned algorithm ultimately generates a smooth and intricate 3D mesh model composed of dense triangles, as illustrated in Figure 11.

2.2. Center of mass calculation for ore point clouds

In the process of ore crushing on a grizzly screen, the crushing point should be selected with the center of mass of the ore as the core consideration. In the event that the crushing point is situated at a distance from the center of mass, this may result in the hammer slipping, thereby affecting the crushing efficiency and posing a safety hazard.

The center of mass of an ore is traditionally determined by means of the vector method. That is to say, the ore point cloud is represented as a collection of vectors. Each ore point cloud possesses a position vector, which is denoted by three-dimensional coordinates (x, y, z). The position vector of each ore point cloud is considered as a vector from the origin (0,0,0) to that point. The summation of all the position vectors is required to obtain a total vector. The total vector is then divided by the number of ore points to calculate the average vector. It can be demonstrated that the coordinates of the average vector are equivalent to the center of mass coordinates.

However, the application of the point cloud to calculate the vectors is not an accurate process due to the existence of some missing point clouds. In order to address this issue, the present paper employs the triangular mesh reconstructed from the point cloud to obtain the area and center of gravity coordinates of the triangles by traversing each triangle in the triangular mesh. The calculation of the total center of mass of the point cloud is based on the area and center of gravity coordinates of the triangles.

Assuming that the three vertices of each triangle are A, B, and C, the area of the triangle is S, the coordinates of the center of gravity are (x_g, y_g, z_g), and the coordinates of each vertex are (x_i, y_i, z_i), the formula for calculating the coordinates of the center of mass of the triangle is shown in Equations 12 to 14.

$$x_g = \frac{1}{3S}\sum_{i=1}^{3}\left(x_i + x_j\right)\left(x_i y_j + x_j y_i\right) \tag{12}$$

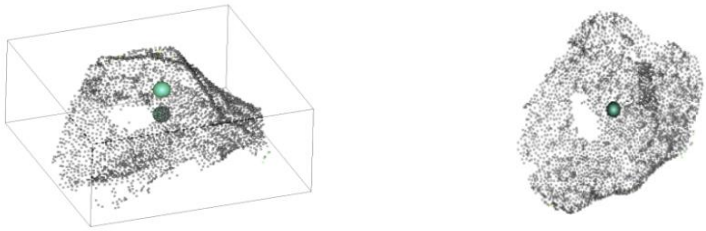
$$y_g = \frac{1}{3S}\sum_{i=1}^{3}\left(y_i + y_j\right)\left(x_i y_j + x_j y_i\right) \tag{13}$$

$$z_g = \frac{1}{3S}\sum_{i=1}^{3} z_i \tag{14}$$

where i and j are permutations of two unequal numbers in 1, 2, and 3. Then, the total center of mass coordinates of the point cloud can be calculated by weighted averaging as shown in Equation 15.

$$C = \frac{1}{\sum_i S_i}\sum_i S_i C_i \tag{15}$$

Where C denotes the total center of mass coordinates of the point cloud, S_i denotes the area of the i-th triangle, and C_i denotes the coordinates of the center of gravity of the i-th triangle. The center of mass calculated using this method is shown in Figure 12.



(a) Side view of ore point cloud (b) Top view of ore point cloud

Fig. 12. Schematic of the center of mass of the ore point cloud

Rys. 12. Schemat środka ciężkości chmury punktów rudy

3. Research on the position measurement method for crushing ore by manipulators

3.1. Planar inspection methods for ore point cloud surfaces

Following the acquisition of the ore point cloud's center of mass position, it is imperative to ensure that the crushing position does not deviate from the center of mass, thereby averting the occurrence of slippage during the crushing process. To this end, it is essential to circumscribe the crushing points to the proximate area surrounding the center of mass. The center of mass is taken as the center, with the Z-axis serving as the centerline of the cylinder. The appropriate radius is then selected to define the cylindrical domain, thereby circumnavigating the set of candidate crushing points.

In order to accurately determine the point cloud within the cylindrical domain, it is first necessary to construct a KD-tree index, with a view to enhancing the efficiency of the subsequent spatial search. The cylindrical domain search algorithm is utilized to accurately determine the center position, radius, and height of the cylinder, thereby facilitating the precise location of the point cloud within the cylindrical domain.

As demonstrated in Figure 13, the following series of steps will yield an ore point cloud within a cylindrical domain with a radius equivalent to two-thirds of the width of the ore, centered at the center of mass.

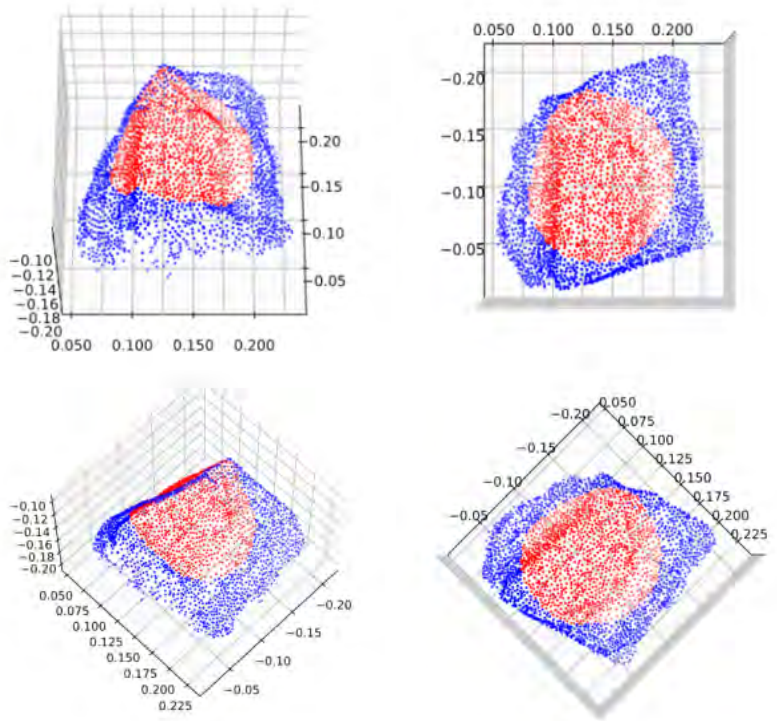


Fig. 13. Ore point cloud in the cylinder domain

Rys. 13. Chmura punktów rudy w domenie cylindra

The RANSAC algorithm is employed to identify the appropriate crushing planes within the aforementioned range of point clouds.

The specific steps are as follows:

1. The initial step involves the input of ore point cloud data. A set of data points is provided, with each data point comprising three-dimensional coordinates (x, y, z).
2. The subsequent step entails the configuration of the algorithm parameters. To this end, the iteration number N is set to 50, the threshold T is established at 100, the distance threshold K is set to 100 millimeters, and the sample number M is set to 50.

The number of iterations, N, indicates the total number of iterations to be carried out by the algorithm. The threshold, T, designates the maximum distance at which a data point is considered to be a point in the plane. The number of samples, M, indicates the number of points selected at each iteration from the dataset, with these points being selected at random.

3. The iteration process involves the selection of a sample number, M, from the data set. This sample number, M, is defined as the set of interior points. The least-squares fitting method is employed to fit a planar model using a set of interior points. The general expression of the plane equation is $Ax + By + Cz + D = 0$ ($C \neq 0$). The transformation of the equation into the following form is necessary: $z = -\frac{A}{C}x - \frac{B}{C}y - \frac{D}{C}$, so that $a_0 = -\frac{A}{C}; a_1 = -\frac{B}{C}; a_2 = -\frac{D}{C};$ then $z = a_0x + a_1y + a_2$. At this time, the corresponding least squares matrix form is shown in Equation 16.

$$A = \begin{pmatrix} x_1 & y_1 & \\ x_2 & y_2 & \\ \vdots & \vdots & \vdots \\ x_n & y_n & \end{pmatrix}; \quad x = \begin{pmatrix} \\ a \\ \\ \end{pmatrix}; \quad b = \begin{pmatrix} z_1 \\ z_2 \\ \cdots \\ z_n \end{pmatrix}; \quad (n \geq 3) \tag{16}$$

where the interior points (x_1, y_1, z_1), (x_2, y_2, z_2), …, (x_n, y_n, z_n) are the input 3D coordinates. Using the solution $x = (A^TA)^{-1}A^Tb$ of the regular system of equations, (a_0, a_1, a_2) is obtained. For each point in the dataset, the Euclidean distance to the fitting plane is calculated as shown in Equation 17.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{17}$$

For each point, if its distance to the fitting plane is less than the threshold K, it is to be added to the inner point set. In the event that the current inner point set exceeds the size of the previous maximum inner point set, it is necessary to update it to the maximum inner point set. It is imperative to reiterate the aforementioned steps until the stipulated number of iterations, N, has been attained to obtain the ultimate maximum inner point set.

4. The planar model is refitted using the maximum set of interior points to obtain the final fit, as illustrated in Figure 14.

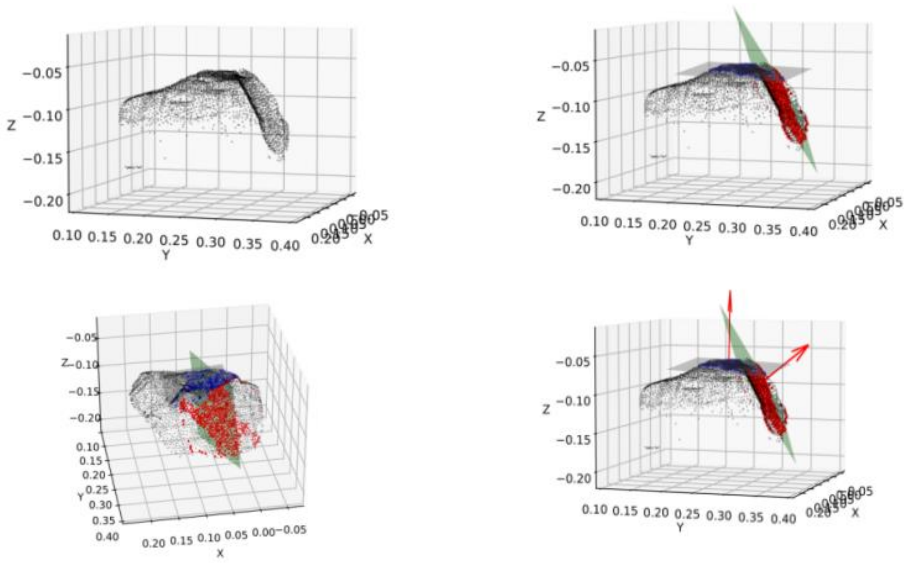


Fig. 14. Plane detection results on the surface of the ore point cloud

Rys. 14. Wyniki wykrywania płaszczyzny na powierzchni chmury punktów rudy

3.2. Calculation of the position of a robot crushing ore

As the collected ore point cloud data undergoes plane detection, the potential for identifying multiple crushing planes is revealed. In this regard, it is necessary to filter suitable crushing planes by setting a tilt angle threshold. In the event that the tilt angle of a plane exceeds the established threshold, the plane is deemed unsuitable for the purpose of crushing and is consequently excluded from further consideration. It is imperative to consider that the point cloud data is acquired in the camera coordinate system, wherein the Z-axis of the camera is perpendicular to the grid screen. Assuming that the plane normal vector is $n = (nx, ny, nz)$ and the Z-axis unit vector $e = (0, 0, -1)$, the angle between the vector and the negative direction of the Z-axis is computed by Equation 18.

$$\cos\theta = \frac{\vec{n}_1\vec{n}_2}{|\vec{n}_1||\vec{n}_2|} \tag{18}$$

It has been established that when the plane inclination angle threshold α is set to a specific value and then adjusted to 45 degrees, the plane is eliminated if α is less than θ. The selection of the crushing plane is contingent upon the prevailing ore conditions, which can be classified into the following categories:

1. In the event that multiple planes are detected within the established tilt angle threshold, the plane exhibiting the smallest tilt angle is designated as the crushing surface.
2. The detection of inclination angles that exceed the established threshold for all planes indicates that the ore surface is, in general, characterized by a substantial inclination, which may be indicative of the presence of sharp vertices. In such instances, direct crushing may result in ore slippage, necessitating adjustments to the ore's position and the selection of alternative crushing angles that are more conducive to the process. It is generally recommended to align the direction of the ore's largest inclination angle with the ore turning process to ensure optimal outcomes.
3. In the absence of plane detection, it can be deduced that the ore's surface exhibits significant unevenness, characterized by substantial curvature variations. The present issue can be addressed by subjecting the ore's center to direct compression.

In the crushing point localization process, a suitable fitting plane is selected by the aforementioned strategy, and the center coordinates of the inner point set are designated as the initial crushing point. The presence of a curvature that exceeds a predetermined threshold serves as an indication of the existence of sharp features or surface unevenness. Consequently, the location of the crushing point must undergo readjustment. The specific method entails the calculation of the curvature distribution of the candidate points and the subsequent selection of the flat area with the smallest curvature as the final crushing point. This approach effectively avoids the high curvature risk area and ensures the safety

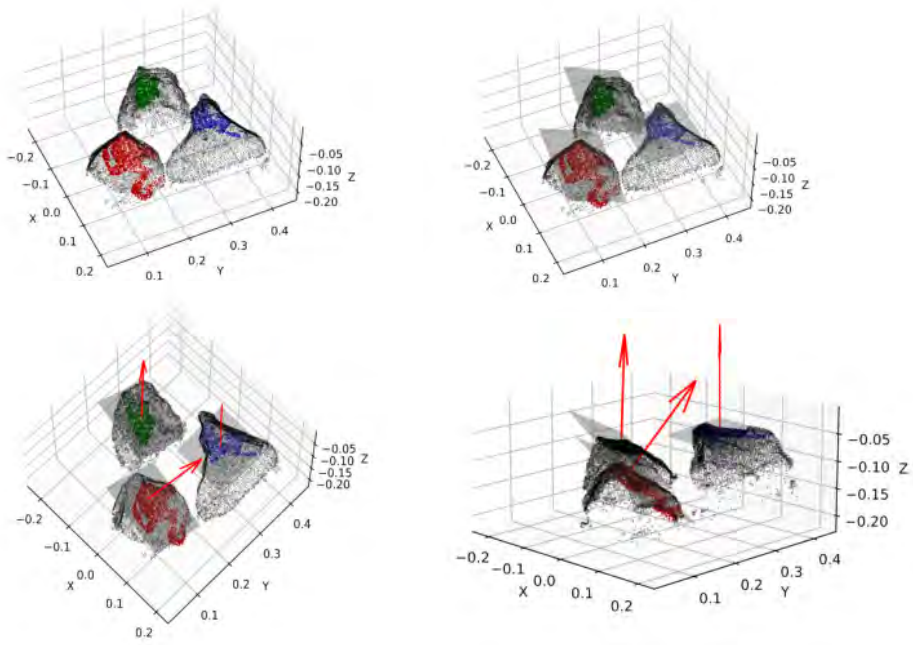


Fig. 15. Determination of the position of the manipulator crushing the ore

Rys. 15. Określenie położenia manipulatora rozdrabniającego rudę

and operational efficiency of the crushing process. The final crushing point orientation is obtained as shown in Figure 15.

Conclusion

In this paper, we propose a segmentation method for regionally grown point cloud instances based on ore concavity and convexity characteristics. This method can effectively segment stacked ore samples. In the case of point clouds that may contain voids, a triangulation algorithm is employed to reconstruct the complete surface. The location of the point cloud center of mass is then calculated with high precision based on the triangulation mesh. The RANSAC sampling consistency algorithm is employed to detect multiple potential crushing surface areas. In conjunction with the plane screening rules proposed in this paper, the plane that meets the attribute characteristics of the ore is selected. This process enables the accurate determination of the direction of ore crushing. The method can be adapted to the processing needs of different types of ores in the field.

The Authors have no conflict of interest to declare.

REFERENCES

Cheng et al. 2024 – Cheng, X., Liu, S., Zhou, J. et al. 2024. FDG-PointNet: fusion of dense connectivity and Gaussian distance for 3D target detection. *Journal of Jilin University (Engineering Edition)*.

Dreier et al. 2025 – Dreier, A.,Tobies, A., Kuhlmann, H., Klingbeil, L. 2025. Stone instance segmentation of rubble masonry based on laser scanning point clouds. *Measurement* 242(PB), https://doi.org/10.1016/j.measurement.2024.115905.

Daif, H. and Marzouk, M., 2025. Point cloud classification and part segmentation of steel structure elements. *Neural Computing and Applications* 37(6), https://doi.org/10.1007/s00521-024-10733-x.

Du et al. 2025 – Du, J., Cai, G., Wang, Z., Jinhe, S., Huang, M., Zelek, J., Marcato Junior, J. and Li, J. 2025. 2025. MTCloud: Multi-type convolutional linkage network for point cloud instance segmentation. *Expert Systems With Applications* 270(2), https://doi.org/10.1016/j.eswa.2025.126432.

Fang et al. 2025 – Fang, Z., Zhuang, C., Lu, Z., Wang, Y., Liu, L. and Xiao, J. 2025. BGPSeg: Boundary-Guided Primitive Instance Segmentation of Point Clouds. *IEEE Trans Image Process*, https://doi.org/10.1109/tip.2025.3540586.

Ghahremani et al. 2021 – Ghahremani M., Williams, K., Corke, F.M.K., Tiddeman, B., Liu, Y. and Doonan, J.H. 2021. Deep Segmentation of Point Clouds of Wheat. *Frontiers in plant science* 12, https://doi.org/10.3389/fpls.2021.608732.

Guo et al. 2024 – Guo, L., He, Y., Du, X. et al. 2024. Lidar volume measurement method based on visual correction. *Journal of Instrumentation* 1–12.

Huang et al. 2018 – Huang, Q., Wang, W. and Neumann, U. 2018. Recurrent slice networks for 3d segmentation of point clouds. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, https://doi.org/10.1109/CVPR.2018.00278.

Huang et al. 2023 – Huang H., Wen, X., Yang, L., Wang, H.F., Gao, T. and Ru, F. 2023. K-mean complementary iterative clustering optimization based on the crazy hunting vulture algorithm. *Journal of Zhejiang University (Engineering Edition)* 57(11), https://doi.org/10.3785/j.issn.1008-973X.2023.11.002.

Huang et al. 2025 – Huang, H., Bu, Y., Xu, H. and Wang, X. 2025. Point Cloud Segmentation Algorithm Based on Contrastive Learning and Label Mining. *Journal of Computer Research and Development* 62(1), https://dx.doi.org/10.7544/issn1000-1239.202330491.

Hugues et al. 2019 – Hugues T., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F. and Guibas, L.J. 2019. KPConv: Flexible and deformable convolution for point clouds. *IEEE/CVF International Conference on Computer Vision (ICCV)*, https://doi.org/10.48550/arXiv.1904.08889.

Jiang et al. 2018 – Jiang, M., Wu, Y., Zhao, T., Zhao, Z. and Lu, C. 2018. Pointsift: A sift-like network module for 3d point cloud semantic segmentation. *Computer Vision and Pattern Recognition (cs.CV)*, https://doi.org/10.48550/arXiv.1807.00652.

Ma et al. 2020 – Ma, X., Xue, H., Zhou, Y. et al. 2020. Point cloud segmentation of sheep body and measurement of body size parameters based on improved regional growth method. *Journal of China Agricultural University* 25(03), pp. 99–105 (*in Chinese* with English abstract)

Li et al. 2025 – Li, Y., Zhang, Z., Li, H. and Zhang, W. 2025. DPPCN: density and position-based point convolution network for point cloud segmentation. *Pattern Analysis and Applications* 28(2), https://doi.org/10.1007/s10044-025-01436-7.

Qi et al. 2017a – Qi, C.R., Su, H., Kaichun, M. and Guibas, L.J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 652–660, https://doi.org/10.1109/CVPR.2017.16.

Qi et al. 2017b – Qi, C.R., Yi, L., Su, H. and Guibas, L. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems 30, https://doi.org/10.48550/arXiv.1706.02413.

Shahraki et al. 2025 – Shahraki, M., Elamin, A. and El-Rabbany, A. 2025. SAMNet++: A Segment Anything Model forSupervised 3D Point Cloud SemanticSegmentation. Remote Sensing 17(7), https://doi.org/10.3390/rs17071256.

Wang et al. 2019 – Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M. and Solomon, J.M. 2019. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)* 38(5), https://doi.org/10.1145/3326362.

Wang et al. 2023 – Wang, J., Jiang, Y. and Wang, L. 2023. Segmentation of building roof surfaces in airborne LiDAR point clouds based on voxel growth. *Journal of Geo-Information Science* 25(12), pp. 2468–2486.

Wu et al. 2020 – Wu, W., Qi, Z. and Li, F. 2020. PointConv: Deep convolutional networks on 3D Point Clouds. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, https://doi.org/10.1109/CVPR.2019.00985.

Wu et al. 2023 – Wu, W.G., Tian, S.Y., Zhang, Z.Y. and Zhang B. 2023. Research on semantic segmentation method for uneven features of unpaved roads. *Automotive Engineering* 45(8), https://doi.org/10.19562/j.chinasae.qcgc.2023.08.017.

Yang et al. 2024 – Yang, W., Pei, S., Liu, Y. et al. 2024. Research on semantic segmentation of point cloud at critical parts of transmission lines based on improved PointNet++. *High Voltage Technology* 1–14.

Zhao, Z.B. 2023. *Construction of point cloud dataset for typical mechanical parts*. University of Electronic Science and Technology.

Zhao et al. 2025 – Zhao, F., Huang, H., Xiao, N., Yu, J. and Geng, G. 2025. A point cloud segmentation algorithm based on multi-feature training and weighted random forest. *Measurement Science and Technology* 36(1), https://doi.org/10.1088/1361-6501/ad824d.

Zhu et al. 2023 – Zhu, J., Bian, Y., Han, F. et al. 2023. Research on binocular active vision point cloud target detection based on improved Euclidean clustering algorithm. *Optoelectronics (Laser)* 34(12), pp. 1288–1297.

Zhu et al. 2024 – Zhu, A., Fei-Peng Da, F.P. and Gai, S.Y. 2024. Three-dimensional point cloud recognition and segmentation sensitive to fused features. *Journal of Xi'an Jiaotong University* 5.

Zolotarev et al. 2025 – Zolotarev, F., Eerola, T. and Kauppi, T. 2025. Deep Unsupervised Segmentation of Log Point Clouds. *Computer Vision and Pattern Recognition (cs.CV)*, https://doi.org/10.48550/arXiv.2503.14244.

ORE CRUSHING POSITION STUDY METHOD BASED ON STACKED ORE POINT CLOUD INSTANCE SEGMENTATION

Keywords

stacked ores, instance segmentation, hypervoxel clustering, crushing pose

Abstract

To address the technical challenges in industrial applications involving stacked ores – particularly the difficulties in distinguishing individual particles resulting from mutual occlusion and geometric irregularities, and the limitations of conventional approaches in precisely determining optimal crushing positions – this study develops a multimodal feature fusion framework integrating ore segmentation with intelligent crushing position determination. Instance Segmentation Level: a hierarchical segmentation framework is constructed by integrating super-voxel clustering with 3D point cloud surface concavity-convexity analysis. To address edge segmentation optimization, a curvature-constrained region growing algorithm is introduced. Furthermore, an adhesion-aware concavity-convexity evaluation function is established to achieve precise separation of adherent ores, effectively mitigating the over-segmentation issues inherent in traditional Euclidean clustering methods when handling complex stacked scenarios. Positional Decision-Making Level: we propose a crushing point localization method incorporating multi-scale geometric feature fusion. Poisson surface reconstruction is employed to construct a continuous geometric model of the ore surface. This is combined with an enhanced RANSAC plane detection algorithm to identify optimal crushing planes, followed by a comprehensive analysis to determine crushing direction vectors. Experimental results demonstrate that the method can effectively segment individual ores in complex stacking scenarios and optimize crushing position determination based on geometric features, providing reliable technical support for automated crushing operations.

METODA BADANIA POŁOŻENIA KRUSZENIA RUDY OPARTA NA SEGMENTACJI INSTANCJI CHMURY PUNKTÓW UŁOŻONYCH W STOSY

Słowa kluczowe

ułożone rudy, segmentacja instancji, klasteryzacja hiperwokseli, pozycja kruszenia

Streszczenie

Aby sprostać wyzwaniom technicznym w zastosowaniach przemysłowych z rudami ułożonymi w stosy – w szczególności trudnościom w rozróżnianiu poszczególnych cząstek, wynikającym ze wzajemnej okluzji i nieregularności geometrycznych, a także ograniczeniom konwencjonalnych metod w precyzyjnym określaniu optymalnych pozycji kruszenia – w niniejszym badaniu opracowano multimodalny model łączenia cech, integrujący segmentację rudy z inteligentnym określaniem

pozycji kruszenia. Poziom segmentacji instancji: hierarchiczny model segmentacji został skonstruowany poprzez integrację klastrowania superwokseli z analizą wklęsłości i wypukłości powierzchni chmury punktów 3D. Aby zoptymalizować segmentację krawędzi, wprowadzono algorytm wzrostu regionu z ograniczeniami krzywizny. Ponadto opracowano funkcję oceny wklęsłości i wypukłości uwzględniającą adhezję, aby uzyskać precyzyjną separację przylegających rud, skutecznie łagodząc problemy z nadsegmentacją, nieodłącznie związane z tradycyjnymi euklidesowymi metodami klasteryzacji podczas obsługi złożonych scenariuszy ułożenia warstw. Poziom podejmowania decyzji pozycyjnych: proponujemy metodę lokalizacji punktu kruszenia, wykorzystującą wieloskalową fuzję cech geometrycznych. Rekonstrukcja powierzchni Poissona została wykorzystana do zbudowania ciągłego modelu geometrycznego powierzchni rudy. Jest ona połączona z ulepszonym algorytmem detekcji płaszczyzn RANSAC w celu identyfikacji optymalnych płaszczyzn kruszenia. Następnie przeprowadzana jest kompleksowa analiza w celu określenia wektorów kierunku kruszenia. Wyniki eksperymentalne pokazują, że metoda ta może skutecznie segmentować poszczególne rudy w złożonych scenariuszach ułożenia warstw i optymalizować określanie pozycji kruszenia w oparciu o cechy geometryczne, zapewniając niezawodne wsparcie techniczne dla zautomatyzowanych operacji kruszenia.