International Journal for Numerical and Analytical Methods in Geomechanics

WILEY

RESEARCH ARTICLE OPEN ACCESS

Enhanced Hybrid Algorithms for Segmentation and Reconstruction of Granular Grains From X-Ray Micro Computed-Tomography Images

Ruidong Li[1] | Pin Zhang[2,3] | Zhen-Yu Yin[1,4] | Brian Sheil[3]

[1]Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hong Kong, China | [2]Department of Civil and Environmental Engineering, National University of Singapore, Singapore | [3]Department of Engineering, University of Cambridge, Cambridge, UK | [4]Research Centre for Resources Engineering towards Carbon Neutrality (RCRE), The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China

Correspondence: Pin Zhang (dr.pin.zhang@gmail.com) | Zhen-Yu Yin (zhenyu.yin@polyu.edu.hk)

Received: 6 December 2023 | **Revised:** 24 August 2024 | **Accepted:** 1 September 2024

Funding: This research was financially supported by the Research Grants Council (RGC) of Hong Kong Special Administrative Region Government (HKSARG) of China (Grant No.: 15220221) and the Research Centre for Resources Engineering towards Carbon Neutrality (RCRE) of The Hong Kong Polytechnic University (Grant No.: 1-BBEM). The second author is supported by the Royal Society under the Newton International Fellowship. The fourth author is supported by the Royal Academy of Engineering under the Research Fellowship Scheme.

Keywords: granular soils | image analysis | machine learning | x-ray computed tomography

ABSTRACT

Accurate three-dimensional (3D) reconstruction of granular grains from x-ray micro-computed tomography (μCT) images is a long-standing challenge, particularly for dense soil samples. This study develops a machine learning (ML) enhanced approach to automatically reconstruct granular grains from μCT images. The novel academic contributions of this paper include (a) a hierarchical strategy based on parameter-independent polygonal approximation, area, and concavity analysis, for the first time, to identify and eliminate both intergranular and intragranular voids; (b) incorporation of a recursive segmentation scheme and ML-based grain classifier to avoid over-segmentation; (c) novel modifications on the determination of splitting paths to enhance segmentation accuracy; and (d) an effective approach of assigning initial level set functions for reconstructing granular grains automatically. The hybrid ML algorithm is applied to μCT images of dense Mojave Mars Simulant. The results indicate that the proposed method can accurately segment grain clumps with unclear boundaries. The new automatic reconstruction algorithm eliminates ineffective operations and achieves a three-fold increase in computational speed than previous methods documented in the literature. Ninety-one percent of grains with distinct boundaries can be reconstructed and the reconstruction ratio reaches 81% even for grains without distinct boundaries. The overall reconstruction ratio of grains increases by 20% compared with previous methods, achieving a step-change improvement for one-to-one mapping of real soil samples.

Abbreviations: A_m, A_v, $\overline{A_v}$, prescribed minimum void area, void area, and average void area, respectively; c_1, c_2, acceleration coefficients; d_c, d_p, user-defined threshold; G_c, grain clump; G_i, individual grain; H_G, H_P, gradient factor; l_{max}, maximum size of the initial level set; l_x, l_y, length of the bounding box of grain in the x and y direction, respectively; n_a, number of concave arc pairs; N_A, N_B, arbitrary adjacent nodes; n_p, number of point pairs; N_S, N_T, N_V, source, target, and visited node, respectively; $\boldsymbol{P_c}$, concave points; $\boldsymbol{P_{c-v}}$, concave–convex point pair; P_t, concave arc vertex; $\boldsymbol{P_v}$, convex points; r_1, r_2, random value between 0 and 1; s_x, s_y, length of initial level set in the x and y direction, respectively; V_e, intergranular void; V_i, intragranular void; δ_G, δ_P, scalar factors; θ_{cc}, concavity–concavity alignment; θ_{cl}, concavity-line alignment; $\boldsymbol{A}$, concave arc; A, B, C, type of grains; $\boldsymbol{D}$, training set; $\boldsymbol{G}$, dominant points of grain; l, constant; m, constant; n, constant; N, number of negative cases; $N[\cdot]$, counting function; p, constant; $\boldsymbol{P}$, dominant points of boundary; P, number of positive cases; $\boldsymbol{R}$, $\boldsymbol{r}$, direction vector; $\boldsymbol{V}$, dominant points of void; V, velocity of particle; W, weight; x, input vector; X, position of particle; y, output; Γ, grain boundary; Φ, β, angle value; ψ, level set function; ω, inertia weight; Ω, studied domain.

International Journal for Numerical and Analytical Methods in Geomechanics, 2024; 48:4206–4220
https://doi.org/10.1002/nag.3832

1 | Introduction

Experimental and numerical research has demonstrated that grain morphologies are strongly linked to the mechanical properties of granular materials, such as strength, stiffness, and anisotropy [1–3]. Over the past two decades, x-ray micro-computed tomography (μCT) has emerged as an effective method for three-dimensional (3D) visualization and characterization of the microscopic structures of granular materials [4, 5]. These μCT images offer a way of accurately reproducing the realistic 3D morphology of grains, opening up new possibilities for analyzing and simulating granular materials [6–8].

Pivotal steps in the reconstruction process include the identification of grain boundaries, segmentation of grains in contact, and accurate characterization of grain morphology in 3D volumetric images [9, 10]. Many conventional techniques for classification and segmentation have been proposed, but all exhibit some form of limitation in processing μCT images [11–14]. The threshold-based approach separates objects based on pixel values with one or multiple thresholds and outputs a binarized image [15]. However, thresholding is sensitive to the quality of μCT images and is prone to failure when the quality of the μCT images is low. Another prevalent technique is the watershed algorithm [11], in which the grayscale image is visualized as a topographic surface based on three basic concepts: minima, catchment basins, and watershed lines [16, 17]. The segmentation process is mimicked by filling up the catchment basins from the minima until all watershed lines have been identified. The watershed algorithm suffers from over-segmentation problems due to the incorrect determination of minima, which can be severe for complex-shaped or porous objects [18]. Modified methods such as "bring-up" and "bring-down" have been proposed, but they are still inaccurate and incur high computational costs [19–21]. Thus, developing a more effective method to segment multiple components in μCT images intelligently is necessary.

More recently, machine learning (ML) algorithms have been used as an alternative technique to classify and segment components in μCT images [22]. One popular solution is to use the convolutional neural network (CNN) to classify and eliminate improperly reconstructed grains [23]. This approach significantly reduces manual inspection time for granular material analysis. However, training a good CNN model is not easy, as its performance greatly depends on the quality of training datasets. A lightweight model is more attractive and convenient to be used in practice. For example, the Trainable Weka Segmentation (TWS) proposed by Lai and Chen [9] requires only a small size dataset for training but performs satisfactorily in grain segmentation. This method first uses the random forest (RF) to learn the difference in pixel features between grain and background. Then the raw image can be converted into a probability map based on a trained RF model. The initial level set can be assigned to the high probability area manually. Finally, the level-set algorithm is adopted to achieve automatic grain reconstruction. The manual assignment of the initial level sets greatly limited the performance of TWS. To solve this problem, a novel convolution kernel-based alternative for determining initial level sets has been proposed [24]. This method replaces manual inspection with sliding window detection. While this approach can automatically determine initial contours, it is prone to duplicating assignments for the same grain. These drawbacks in both segmentation and reconstruction pose a significant challenge to achieving an accurate and reliable ML-based approach.

To overcome these limitations, this study proposes a novel end-to-end framework that integrates hybrid ML algorithms with image analysis techniques to improve the accuracy (*ACC*) of μCT image segmentation and reconstruction. The contributions of this paper include (i) automatic identification of grains from a noisy background, (ii) automatic identification and removal of intragranular and intergranular voids, (iii) determination of a criterion for grain segmentation, and (iv) automatic reconstruction of grains. By way of validation, the proposed approach is applied to dense granular Mojave Mars Simulant to demonstrate accurate segmentation and 3D reconstruction from μCT images. Benchmarking against two alternative methods proposed in the literature highlights step-change improvements in the computational efficiency of the present approach.

2 | Methodology

2.1 | Overview of the Proposed Approach

Figure 1 presents a schematic illustration of the proposed automatic grain segmentation and reconstruction scheme comprising: (1) probability map, (2) hierarchical interior void elimination (HIVE), (3) grain segmentation, and (4) grain reconstruction. At a high level, the process begins by converting raw μCT images into a probability map where all pixels are assigned a value ranging between 0 and 1, that is, 0 for background and 1 for grain [24]. The connected pixels with values larger than 0 are defined as grain regions, while others are voids located within the grain regions. HIVE (Step 2) is used to remove voids. Thereafter, grain regions without voids are classified into two types: individual grains G_i and grain clumps G_c (multiple G_i) that require further segmentation. G_c is distinguished by a grain classifier and segmented iteratively by combining the grain classifier and a concavity-analysis-based segmentation scheme. Finally, 3D grains are reconstructed automatically by the level set method based on adaptive initial contour assignment. This end-to-end approach allows for a fast, objective, and accurate segmentation and reconstruction of granular grains.

2.2 | RF-Based Probability Map

RF is a predictor consisting of a collection of decision trees. Suppose there is a training set $D = [(x_1, y_1), \dots, (x_n, y_n)]$ of n independent random input-output pairs (x, y), where x denotes an input vector composed of specific features and y is the output [25]. The training of RF determines the maximum number of features at nodes *mtry* and the number of decision trees *ntree*. Each decision tree casts a unit vote for the most popular output $y(x)$ at input $\boldsymbol{x}$. The final output y is the average value of all decision trees in the RF:

$$\mathrm{y} = \frac{1}{\mathrm{ntree}} \sum_{\mathrm{i}=1}^{\mathrm{ntree}} \mathrm{y_i}(\mathbf{x}) \tag{1}$$

where $y_i(x)$ denotes the prediction of ith tree for input $\boldsymbol{x}$. Particle swarm optimization (PSO) is a typical population-based

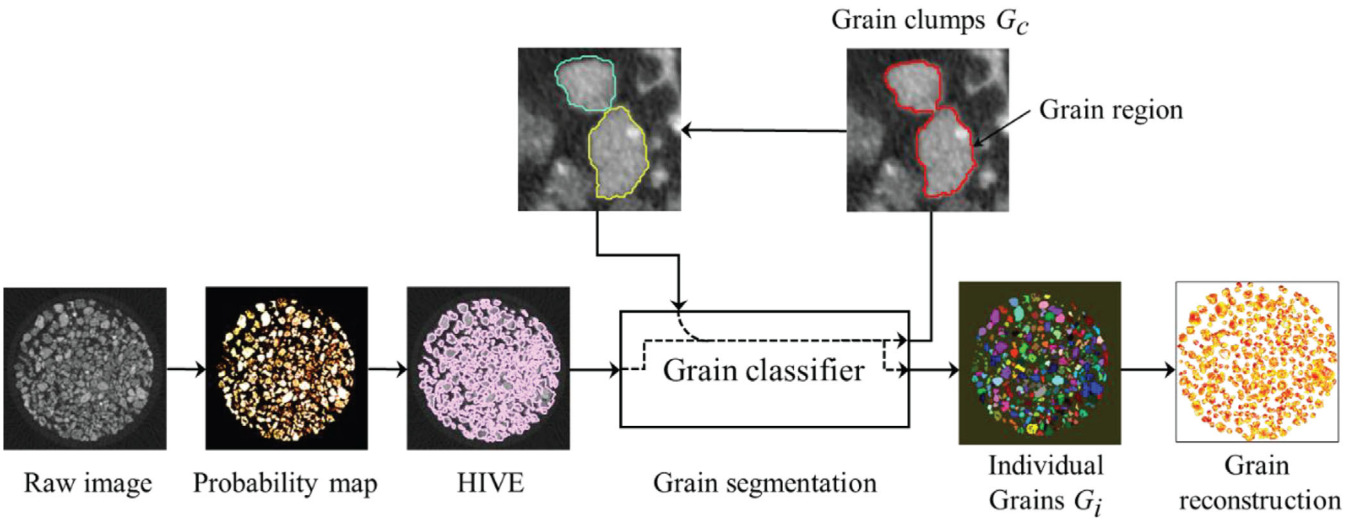


FIGURE 1 | Flowchart of the proposed image-based 3D reconstruction approach.

stochastic optimization algorithm motivated by the social life of animals [26, 27]. The global optimal solution is searched by updating velocity V and position X of particles iteratively as follows:

$$V_k^{t+1} = \omega V_k^t + c_1 r_1 \left(P_k^t - X_k^t\right) + c_2 r_2 \left(P_g^t - X_k^t\right) \tag{2}$$

$$X_k^{t+1} = X_k^t + V_k^{t+1} \tag{3}$$

where c_1 and c_2 are acceleration coefficients, r_1 and r_2 denote the random numbers within the range $[0, 1]$, ω represents the inertia weight, P_k^t is the best location for k th particle at t th iteration and P_g^t is the global best location for all particles at t th iteration. The PSO algorithm aids in the determination of the optimal RF hyperparameters (*mtry* and *ntree*) during the training process.

In this study, the RF and PSO algorithms are adopted to build a PSO-RF-based training framework. This framework is used to convert the raw μCT images into probability maps used to distinguish background and grains [28, 29]. Four methods are adopted to intensify the difference between background and grains: grayscale, top hat filter, salt, and pepper denoising, and Gaussian blur. The raw image can be expanded to four matrices after being processed by these four methods. In this study, n pixels in the raw images, including grains and the background, are selected for training. The labels (0 for background and 1 for grains) and corresponding values in the four matrices of these n pixels form a matrix of size $n \times 5$, regarded as the input. The output is the probability value of a pixel that denotes grains, varying from 0 to 1. Once the RF-based model is trained, the probability map of the input μCT image can be obtained. The grain regions and voids are determined by the intrinsic MATLAB function, *bwboundaries* which is used to determine the boundaries of objects in the image [30, 31].

2.3 | Hierarchical Interior Void Elimination

Figure 2a,b depicts an example raw μCT image and corresponding ground truth (seven grains, numbered from 1 to 7) determined manually. The corresponding probability map, and identification of grain regions and voids are illustrated in Figure 2c,d, respectively. Voids are classified as either (a) intragranular voids V_i or (b) intergranular voids V_e. V_i are typically a result of image noise or impurities. They are generally small but abundant and can be solved by an existing "fill and remove" strategy [18], in which V_i are filled before segmentation and removed after segmentation. V_e represents voids among grains but are incorrectly incorporated into the region due to the partial volume effect [32, 33]. V_e are relatively large and must be eliminated before grain segmentation.

A simple but effective approach for identifying and eliminating V_e is proposed here. The first step is to simplify the boundary using the parameter-independent polygonal approximation (PIPA) method [34, 35]. PIPA can automatically simplify the boundary into a set of dominant points P, which preserves shape characteristics (for more details see [36]); G and V denote the dominant points of grain and void, respectively (Figure 3a).

Thereafter, for a given P with more than three points (to define a closed area), the concave points P_c and convex points P_v in the set are determined by the vector product method [37, 38], which is defined as follows (Figure 3b):

$$\boldsymbol{P}_c = \left\{ P_k \in \boldsymbol{P} | \overrightarrow{P_{k-1}P_k} \cdot \overrightarrow{P_k P_{k+1}} > 0 \right\} \tag{4}$$

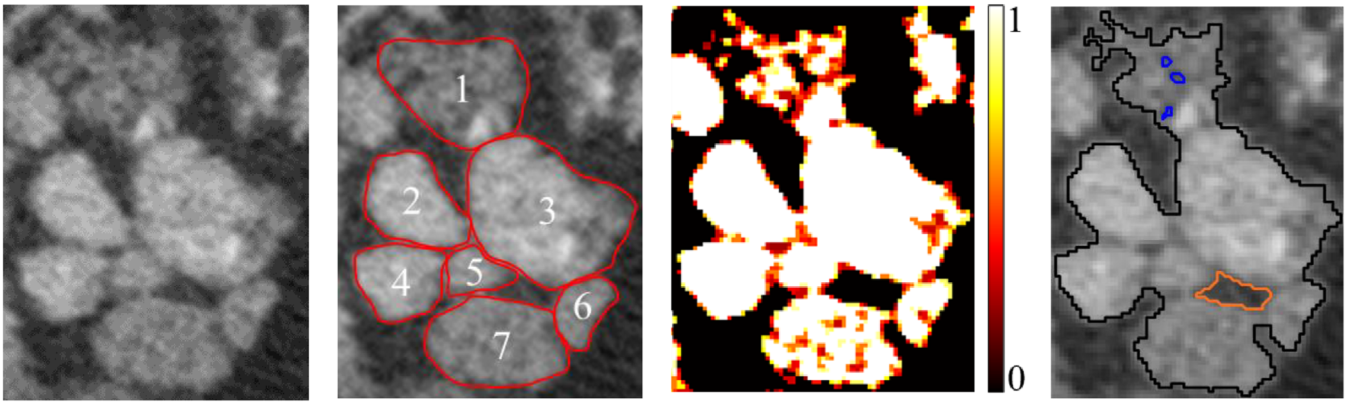
$$\boldsymbol{P}_v = \left\{ P_k \in \boldsymbol{P} | \overrightarrow{P_{k-1}P_k} \cdot \overrightarrow{P_k P_{k+1}} > 0 \right\} \tag{5}$$

where P_k denotes the k th dominant point in the set of P in a clockwise direction. The identified concave and convex points of P in grain region and voids are illustrated in Figure 3c. In this study, the void that conforms to the following three criteria is regarded as V_e:

$$A_v \geq \max(\overline{A_v}, A_m) \tag{6}$$

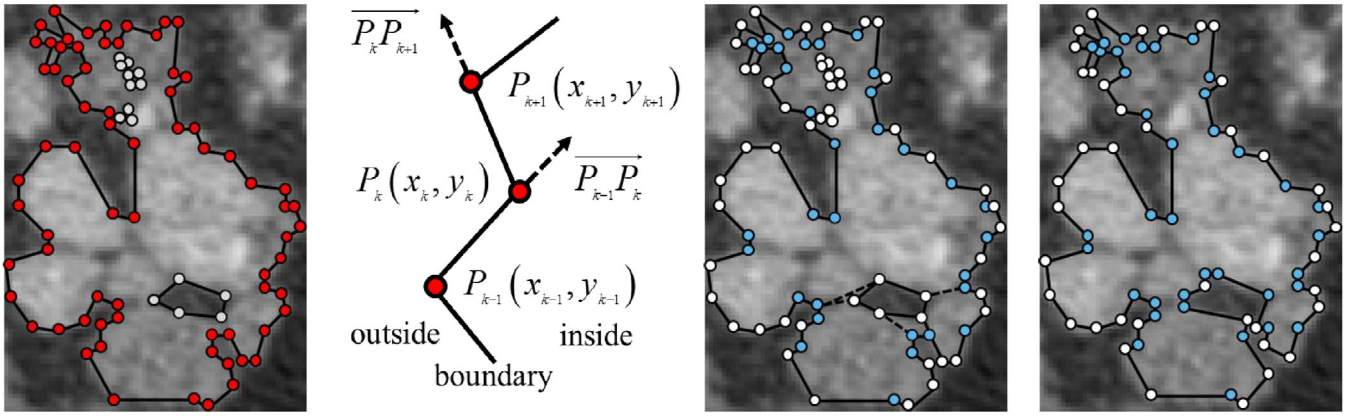
$$\mathrm{N}(V_v) \geq 3 \tag{7}$$

$$\mathrm{N}(P_{c-v}) \geq 2 \tag{8}$$



(a) (b) (c) (d)

FIGURE 2 | Schematic illustration of: (a) an example raw μCT image; (b) ground truth; (c) probability map; and (d) identification of grain boundary (black line), V_i (blue line), and V_e (orange line).



(a) (b) (c) (d)

FIGURE 3 | Schematic illustration of the process to remove V_e: (a) dominant points G (red dots) and V (grey dots); (b) vector product method to identify concave and convex points on a given boundary line; (c) example identified P_c (blue dots) and P_v (white dots); and (d) P_c (blue dots) and P_v (white dots) of the same region, now with V_e removed.

where A_v and $\overline{A_v}$ are the area of the studied void and the average area of all voids within the same grain region, respectively, A_m is a prescribed minimum area of V_e, $\mathrm{N}(\cdot)$ is a function that counts the number of elements in the set, V_v denotes convex points in V, P_{c-v} denotes the concave–convex point pairs where concave points from G while convex points from V, are connected by the black dash lines in Figure 3c for reference. Those concave points are the closest points to V in G with a distance smaller than a user-defined threshold d_p. Once the intergranular voids V_e have been identified, the pixels on the connection line of the concave–convex point pair with the shortest distance, obtained by Bresenham's line algorithm [39], are filled with 0. Then, the boundary of the grain region without V_e can be obtained by MATLAB function *bwboundaries* and simplified into G by PIPA again. Finally, the P_c and P_v of the grain region obtained by the vector product method are shown in Figure 3d.

2.4 | Grain Segmentation

2.4.1 | Grain Classification

G_i are generally smaller and more regular in a convex shape compared to G_c which are substantially larger and irregular in a concave shape. These differences are leveraged to develop a grain classifier by using shape features as inputs. However, there is little guidance in the literature as to the most appropriate shape features to use [40, 41]. Thus, a separate hybrid PSO-RF algorithm is adopted to train the grain classifier. Seven shape features are selected for classifier training: area (A_a), area of the convex hull (A_c), eccentricity (e_c), extent (e_x), perimeter (P_r), solidity (S_l), and circularity (C_c). The area A_a and area of the convex hull A_c are determined by counting the number of pixels within the grain and its convex hull. The perimeter P_r equals the number of pixels on the contour of the grain, while the eccentricity e_c refers to the

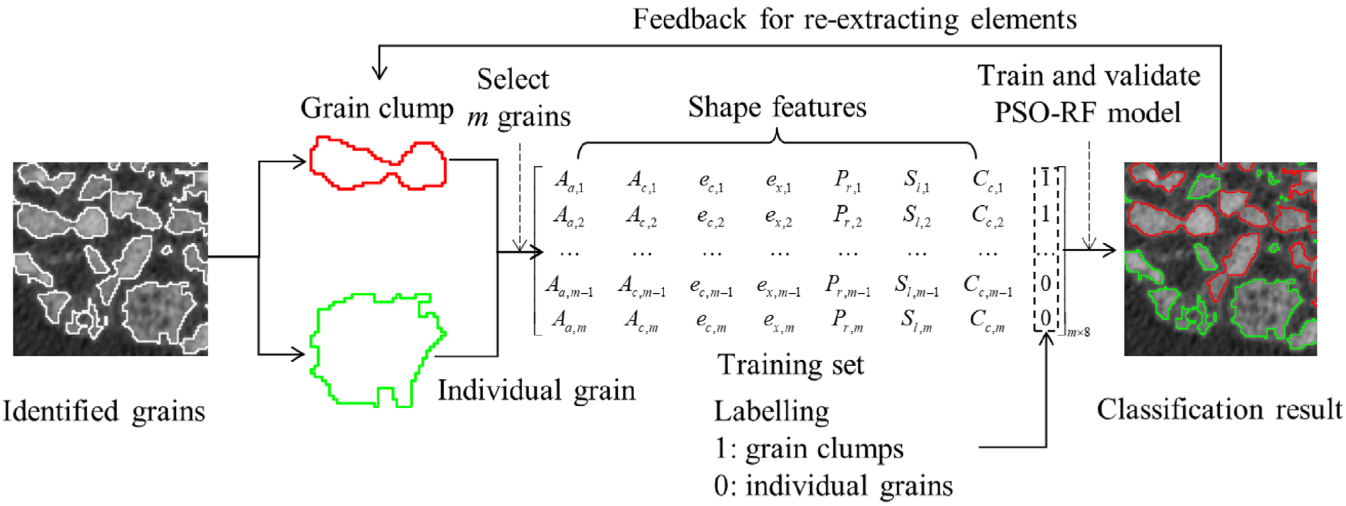


FIGURE 4 | Flowchart of grain classification process.

eccentricity of the eclipse with the same second moment of area as the studied grain. The extent e_x is expressed as follows:

$$e_x = \frac{A_a}{A_b} \tag{9}$$

where A_b denotes the area of the bounding box of the studied grain. The circularity C_c and solidity S_l are calculated as follows:

$$C_c = \frac{4 * A_a * \pi}{P_r^2} \tag{10}$$

$$S_l = \frac{A_a}{A_c} \tag{11}$$

The MATLAB function *regionprops* is adopted to calculate these shape features automatically [42].

For the training datasets, m grains, including grain clumps G_c and individual grains G_i, are labeled manually as 1 and 0, respectively, and form the $m \times 1$ output dataset. Their corresponding seven shape features form the input dataset of size $m \times 7$. A visual explanation of the overall workflow is presented in Figure 4. The trained grain classifier can identify whether the studied region is a grain clump G_c and individual grain G_i.

2.4.2 | Grain Clump Split Point Identification

The purpose of grain clump split point identification is to determine the best split points to convert the identified G_c to G_i [43, 44]. The consecutive P_c and corresponding two closest P_v are regarded as a whole to describe the concave parts of G_c, denoted as the concave arc $\boldsymbol{A}$ (see yellow dashed lines in Figure 5a). Suppose $\boldsymbol{A}$ is composed of a set of k consecutive points in a clockwise direction: $A = \{P_1, P_2, \dots, P_k\}$, the $(k+1)/2$th point (if k is odd) or the midpoint of $(k+1)/2$th and $k/2$th (if k is even) point is selected as the vertex P_t (red dots in Figure 5b). Consequently, the angle value $\overline{\Phi_i}$ and direction $\overline{R_i}$ of i th $\boldsymbol{A}$ are estimated by (**b**):

$$\overline{\Phi_i} = \frac{\sum_{j=1}^{n_p} \phi_j}{n_p}, \phi_j \in \Phi_i \tag{12}$$

$$\overline{R_i} = \sum_{j=1}^{n_p} \frac{r_j}{r_j}, r_j \in R_i \tag{13}$$

where Φ_i denotes the set of angels ϕ_j ($0° < \phi_j < 180°$) that are formed by P_t and nth point pair (P_{t-n}, P_{t+n}), R_i is the set of vectors r_j that are formed by P_t and the midpoint of the nth point pair (P_{t-n}, P_{t+n}), n_p represents the number of point pairs in $\boldsymbol{A}$ that is determined by:

$$n_p = \begin{cases} (\mathrm{k}+1)/2, \text{ if k is odd} \\ \mathrm{k}/2 \ \ 1, \text{ if k is even} \end{cases} \tag{14}$$

To determine the best concave arc pairs for segmentation, all possible concave arc pairs are evaluated by a rule-based matching function. For G_c with p concave arcs, the number of possible concave arc pair combinations n_a is calculated as follows:

$$n_a = \frac{p!}{2!(p-2)!} \tag{15}$$

For those concave arc pairs (A_i, A_j), they are evaluated by the following rule-based cost function (Figure 5c):

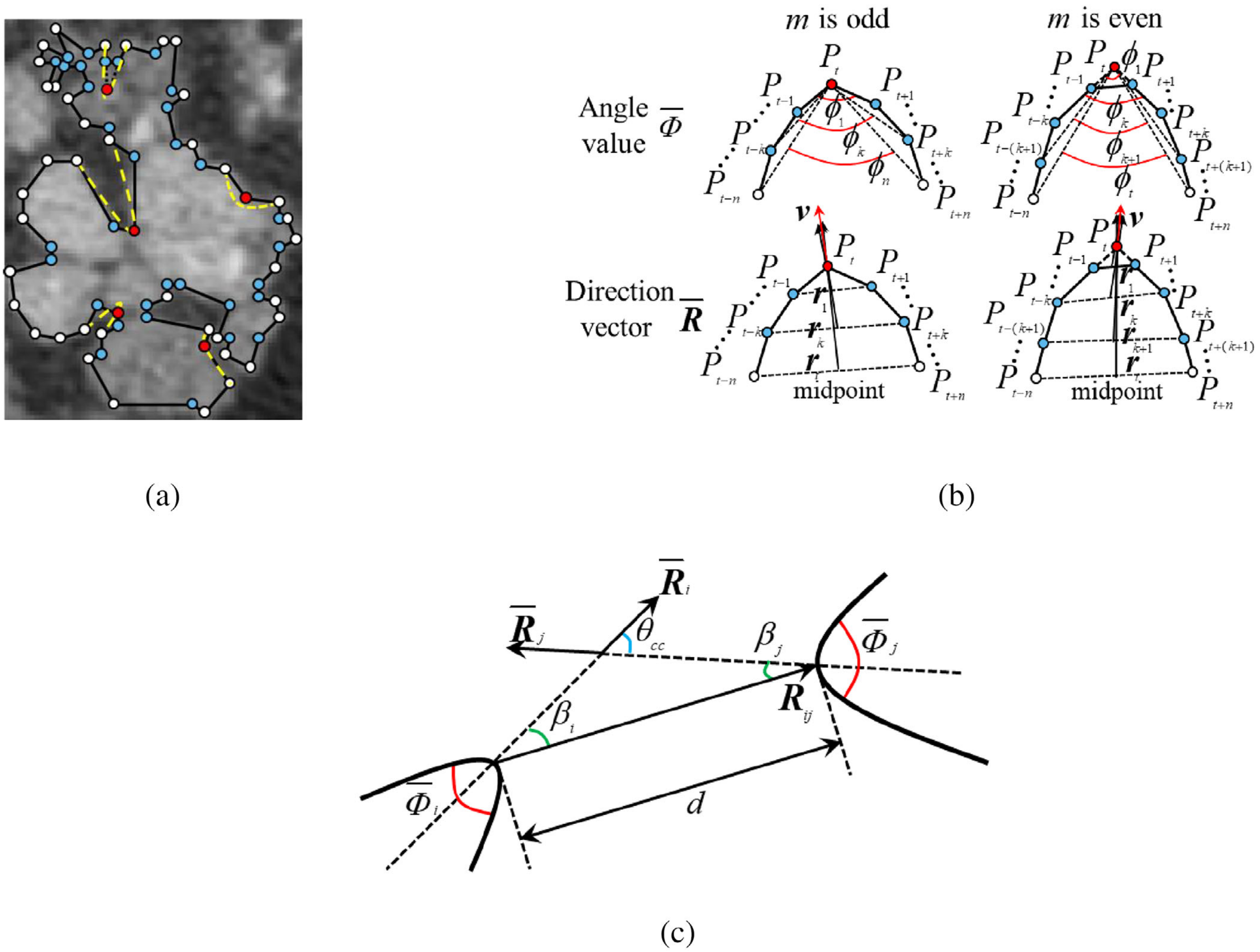
$$C(k) = \begin{cases} (\theta_{cc}^4 + 1) \cdot (\theta_{cl}^2 + 1) \cdot e^d \cdot \sqrt{\bar{\Phi}_i \cdot \bar{\Phi}_j}, d \le d_c \\ (\theta_{cc}^2 + 1) \cdot (\theta_{cl}^4 + 1) \cdot d^2 \cdot \sqrt{\bar{\Phi}_i \cdot \bar{\Phi}_j}, d > d_c \end{cases} \tag{16}$$

where $C(k)$ denotes the cost of selecting k th concave arc pairs, e represents the Euler's number, d and d_c denotes the distance between two split points and the prescribed critical distance, respectively, $\overline{\Phi_i}$ and $\overline{\Phi_j}$ are the angle values of concave arcs (A_i, A_j), respectively, θ_{cl} and θ_{cc} represent the concavity-line alignment and concavity–concavity alignment (for more details, see [45]), respectively, expressed as follows:

$$\vartheta_{cl} = \max(\beta_i, \beta_j) = \max\left(\cos^{-1}(\bar{\boldsymbol{R}}_i \cdot \boldsymbol{R}_{ij}), \cos^{-1}(\bar{\boldsymbol{R}}_j \cdot -\boldsymbol{R}_{ij})\right) \tag{17}$$

$$\vartheta_{cc} = \pi - \cos^{-1}(\overrightarrow{\boldsymbol{R}_i} \cdot \overrightarrow{\boldsymbol{R}_j}.) \tag{18}$$

where $\overline{\boldsymbol{R}_i}$ and $\overline{\boldsymbol{R}_j}$ are the direction of A_i and A_j, respectively. R_{ij} is the direction of the split line. β_i is the angle between $\overline{R_i}$



(a)

(b)

(c)

FIGURE 5 | Concavity-analysis-based matching: (a) concave arcs and arc vertexes; (b) estimation of angle value and direction vector; and (c) rule-based matching function.

and R_{ij}, and β_j is the angle between $\overline{R_j}$ and $-R_{ij}$. The concave arc pair with the minimum cost is nominated as the candidate concave arc pair (A_S, A_T) for segmentation. The arc vertexes of concave arcs A_S and A_T are ultimately selected as the split points.

2.4.3 | Determination of Optimal Grain Clump Splitting Path

The improved Dijkstra's algorithm [46] is used to determine the optimal splitting path between split points. It works based on a directed graph, which includes a source node N_S, a target node N_T, and positive weights between adjacent nodes, as shown in Figure 6a [47]. The circles labeled alphabetically represent the nodes, while numbers denote the weight between ends. This approach begins by determining and setting the neighbor node N_S with the smallest weight as the visited node N_V. Next, for N_V, its neighbor node is determined and given the smallest weight as the new N_V. N_V is continually updated until the target node N_T is reached. In this study, to enhance computational efficiency, for a given G_c, the nodes are pixels both within G_c and the rectangle with diagonal points N_S and N_T, as depicted in Figure 6b. The split points are selected as N_S and N_T. The weight $W(N_A, N_B)$ between arbitrary adjacent nodes N_A and N_B are calculated by:

$$\mathrm{W}(\mathrm{N_A}, \mathrm{N_B}) = \delta_\mathrm{G} \times \mathrm{H_G}(\mathrm{N_B}) + \delta_\mathrm{P} \times \mathrm{H_P}(\mathrm{N_B}) \tag{19}$$

where δ_G and δ_P are scalar factors, $H_G(N_B)$ and $H_P(N_B)$ represent the gradient factor and probability factor, respectively, defined by:

$$H_G(N_B) = 1 - \frac{G(N_B)}{\max(G)} \tag{20}$$

$$H_P(N_B) = 1 - \frac{P(N_B)}{\max(P)} \tag{21}$$

where $\max(G)$ and $\max(P)$ denote the maximum gradient value [48] and probability value in the directed graph, respectively. The gradient value and probability value refer to the image gradient and pixel value at the position in the directed graph, respectively. An image gradient is a directional change in the intensity or color of an image. These gradient values can be determined by the derivative of an image which is approximated through finite differences [49]. $P(N_B)$ is the probability value at the node N_B, and $G(N_B)$ is the gradient value at the node N_B. Finally, the pixels in the optimal splitting path are filled with zero to segment G_c

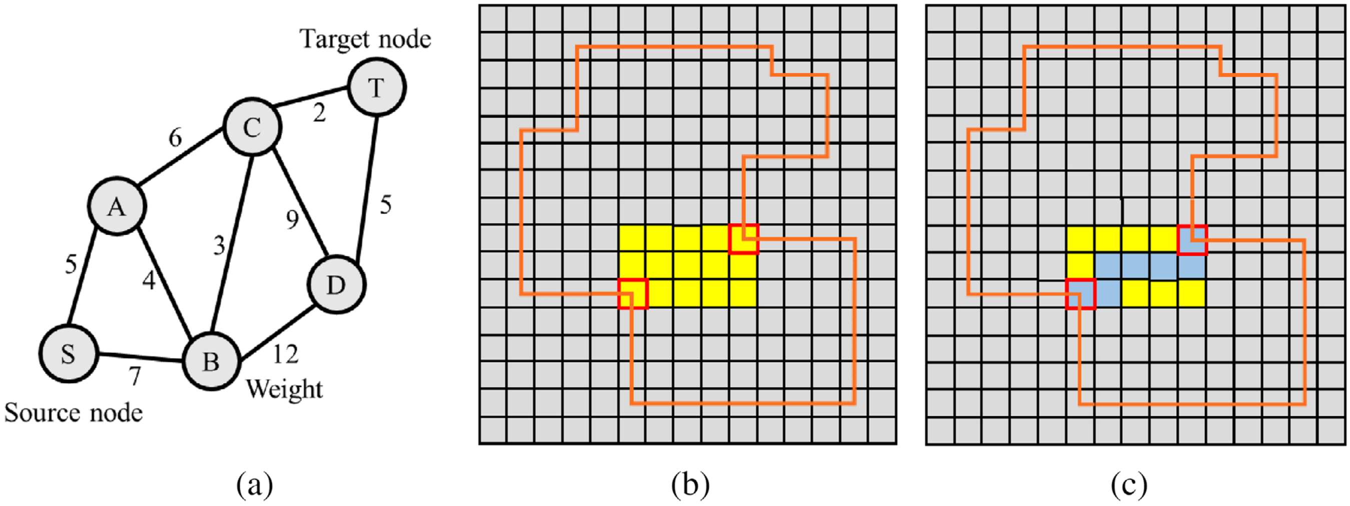


FIGURE 6 | Illustration of optimal grain clump splitting path: (a) directed graph; (b) boundary (orange line), nodes (yellow pixels), N_S and N_T (marked by red) of an example G_c; and (c) optimal path (blue pixels) of an example G_c.

ALGORITHM 1 | Grain segmentation algorithm.

```
function GrainSegment($L$)
    list_ini = SearchRegions($L$); % $L$ is the vector in which each component denotes a grain region; using MATLAB bwboudaries to determine the grain regions
    for $i$ = 1: number of grain regions
        is_Gc = GrainClassifier($L$, $i$); % return 1 if it is a grain clump
        if is_Gc == 1
            [$L$, list_subs] = GrainSegment ($L$, $i$); % implement grain clump split point identification and determine the optimal grain clump splitting path
        end if
    end for
end function
```

Note: searchRegions is a function for determining grain regions; GrainClassifier is a function for achieving grain classification, representing the "Grain classification" part; GrainSegment is a function denoting the "Grain segmentation" part.

into two new grain regions (Figure 6c). The whole procedure, including grain classification, identification of split points, and determination of split path is the so-called grain segmentation function. This function iterates through each grain region and performs itself again and again until no G_c are observed. The pseudocode of the grain segmentation function is presented in Algorithm 1 [50].

2.5 | Grain Reconstruction

Based on the segmentation result, an edge-based level set approach is used to reconstruct the 3D grain surface [9, 24]. It represents the surface as the zero level-set of a higher dimensional function Ψ capable of handling complex changes in topology [51–53]. A grain boundary Γ, defined as the zero level-set, can be expressed as follows:

$$\Gamma = \{(x, y, z) \in \boldsymbol{\Omega} | \Psi(x, y, z) = 0\}. \tag{22}$$

where $\Psi(x, y, z)$ is the implicit level set function. During the reconstruction process, the level set function Ψ evolves by minimizing the appropriate energy function using the framework proposed by Li et al. [51]. Finally, the grain surface is represented by a triangle mesh using the marching cubes algorithm [54].

It is noted that an initial closed area should be assigned for all individual grains before implementing level set evolution. In this study, the size of the initial area for each grain is determined by the following hierarchical strategy:

$$s_x = \min((\mathrm{ceil}(0.5 \times l_x), l_{max}), s_x \geq 2 \tag{23}$$

$$s_y = \min((\mathrm{ceil}(0.5 \times l_y), l_{max}), s_y \geq 2 \tag{24}$$

where ceil(x) is a function that maps x to the smallest integer greater than or equal to x, min(·) denotes the function that returns the minimum value, s_x and s_y denote the size of the initial area in the x and y directions, respectively, l_x and l_y denote the length of the bounding box of grain in the x and y directions, respectively, and l_{max} denotes the maximum size of the initial area. The centroid of the initial area is identical to that of the grain. Once the initial level set is assigned properly, the grain will be reconstructed progressively with level set evolution. More details can be found in [55].

3 | Results

3.1 | Datasets and Model Development

In this study, 100 continuous μCT images of dense Mojave Mars Simulant (grain size ranges from 0.075 to 4 mm) are adopted from Zhang et al. [24] to verify the proposed grain segmentation and reconstruction process. The dimension size of the raw dataset is

TABLE 1 | Main hyperparameters of RF.

Hyperparameters	Description	Value
mtry	Maximum number of features at nodes	By PSO
ntree	Number of decision trees	By PSO
min_sample_split	Minimum number of samples at splitting nodes	2
min_sample_leaf	Minimum number of samples at leaf nodes	5
max_depth	Maximum depth of a tree	Automatic

Abbreviations: PSO, particle swarm optimization; RF, random forest.

TABLE 2 | Parameters used in PSO.

ω	c_1	c_2	Popsize	Maxgen
0.9	2.5	0.5	10	100

Abbreviations: Maxgen, maximum number of generations; Popsize, size of the population.

TABLE 3 | Parameters used in HIVE, grain segmentation, and grain reconstruction.

Parameters	Procedures	Value
A_m	HIVE	20 pixels
d_p	HIVE	10 pixels
d_c	Grain segmentation	5 pixels
δ_G	Grain segmentation	0.5
δ_P	Grain segmentation	0.5
l_{max}	Grain reconstruction	7 pixels

Abbreviation: HIVE, hierarchical interior void elimination.

$512 \times 500 \times 1832$ pixels (height × width × depth, or $x \times y \times z$). In this study, 100 slices are adopted from $z = 300$ to 400. The hyperparameters of PSO-RF-based models originate from Probst et al. [56] and are summarized in Tables 1 and 2. Two PSO-RF models are developed: one for predicting the probability map and the other for classifying grains. The training set for the PSO-RF probability map predictor includes 28 and 10 pixels corresponding to grains and background, respectively, taken from 10 random 2D images. For training the PSO-RF grain classifier, 50 individual grains and 50 grain clumps taken from 100 random 2D images are chosen. The parameters used in the HIVE, grain segmentation, and grain reconstruction algorithms are determined by trial and error and are summarized in Table 3. Moreover, the parameters of the level set method are the same as those adopted in Lai and Chen [9].

3.2 | Probability Map and Void Elimination

The raw μCT image $z = 350$ of Mojave Mars Simulant and the corresponding test result obtained by the trained PSO-RF probability map predictor are shown in Figure 7a,b, respectively. In the raw image, the boundaries of the grains are unclear, especially for smaller grains. For the probability map, black and white denote the probability of a grain of 0 and 1, respectively, such that lighter areas indicate a greater likelihood of representing a grain. Compared to the raw image, grain boundaries are slightly clearer in the probability map.

TABLE 4 | Confusion matrix of void classification.

		Predicted	
		Intergranular void V_e	**Intragranular void V_i**
Actual	Intergranular void V_e	2	6
	Intragranular void V_i	77	1

The ground truth of void classification is determined manually and presented in Figure 7c. The boundaries of identified grain, intragranular voids V_i and intergranular voids V_e are represented by white, green, and red lines, respectively. For the present image, V_i dominate in number and are mainly distributed in large clumps. The number of V_e is relatively few but critical to the subsequent segmentation and reconstruction process. The final void classification result is illustrated in Figure 7d. It can be seen from Figure 7d that almost all voids are correctly identified and classified.

The *ACC* of classification can be determined by calculating the quantity of correctly identified class instances (true positives, *TP*), the quantity of correctly identified instances that do not pertain to the class (true negatives, *TN*), and instances that either were incorrectly assigned to the class (false positives, *FP*) or that were not recognized as class instances (false negatives, *FN*) as follows:

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{P} + \mathrm{N}} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{25}$$

where *P* and *N* denote the number of positive cases and negative cases. The confusion matrix for void classification is presented in Table 4. The *ACC* of hierarchical classification criteria in classifying the interior voids reaches 96.5%, indicating the effectiveness of the proposed strategy for classifying voids. For the identified V_e, they are eliminated by filling the connection line between the candidate concave–convex point pair. Figure 7e presents an example region taken from Figure 7d with two V_e and three V_i. In Figure 7f, two V_e are removed correctly, causing the boundaries of grains to become clearer and easier to be segmented. This reveals the importance of eliminating V_e for grain segmentation and reconstruction.

3.3 | Grain Classification and Segmentation

The loss value generated by RF during training is illustrated in Figure 8a and converges at approximately 10 epochs. Training of RF starts with a small loss value for the sake of its excellent ability of classification; thus, there is no large decrease in the loss value during the training process. The initial grain classification

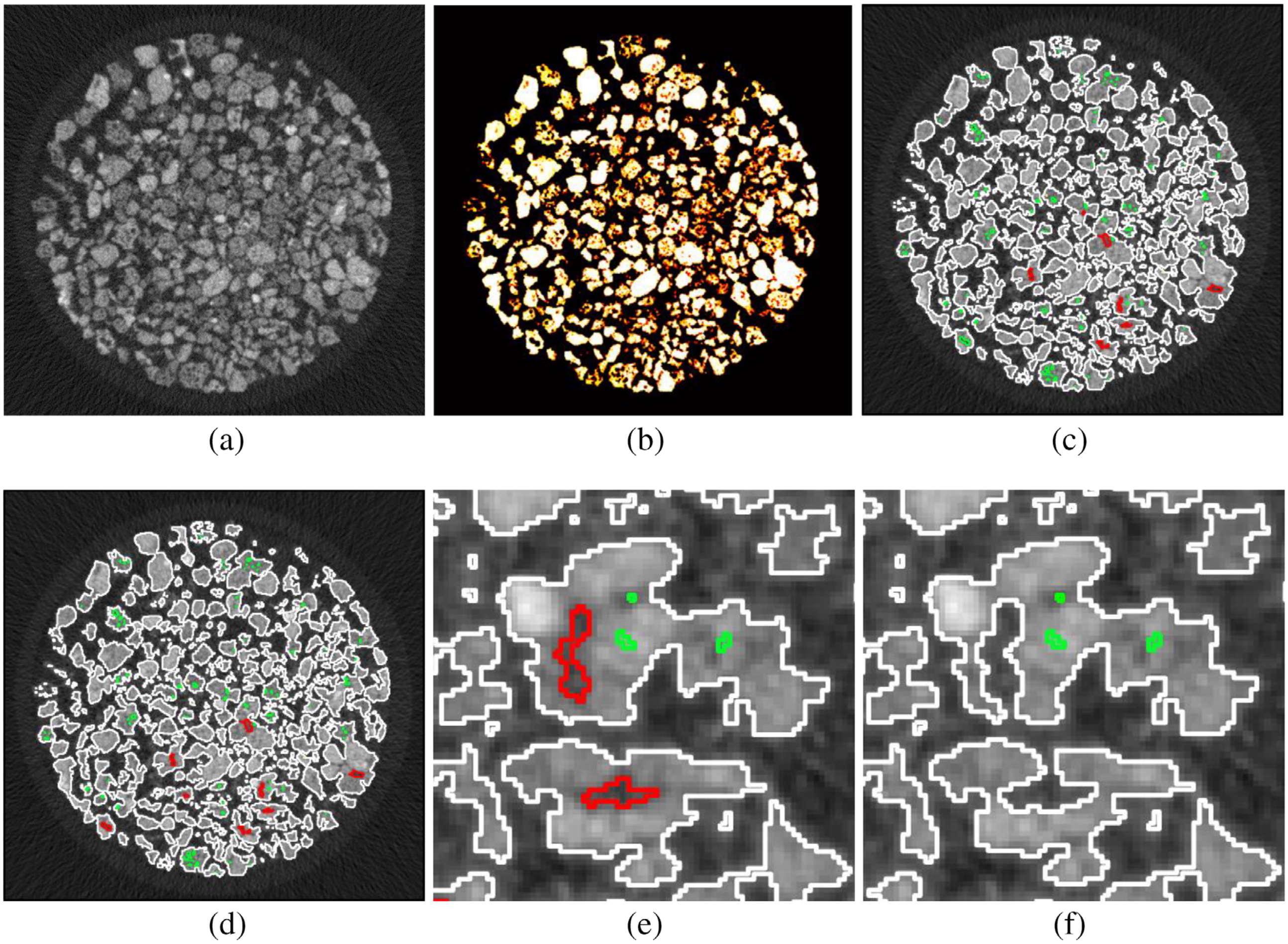


FIGURE 7 | Results of probability map and void elimination: (a) raw image; (b) probability map; (c) ground truth of void classification; (d) void classification result; (e) identified region before void elimination; and (f) identified region after void elimination.

TABLE 5 | Confusion matrix of grain classification.

		Predicted	
		Individual grain G_i	**Grian clump G_c**
Actual	Individual grain G_i	117	2
	Grian clump G_c	5	41

result obtained using the trained PSO-RF grain classifier on the regions without voids (Figure 7d) is presented in Figure 8b. The boundaries of grain clumps G_c and individual grains G_i are denoted by red and green lines, respectively. The confusion matrix for grain classification is presented in Table 5. The *ACC* of hierarchical classification criteria in classifying the grain clumps and individual grains reaches 95.8%, indicating that nearly all G_c and G_i are correctly identified by the grain classifier.

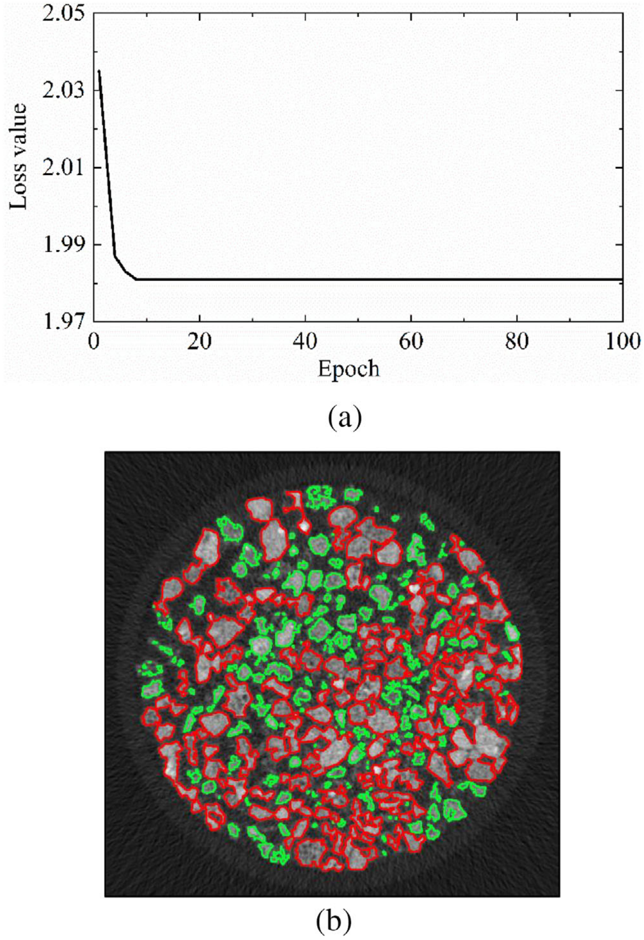
To evaluate the effectiveness of the proposed techniques for grain identification and segmentation, the segmentation results in the G_c of Figure 2d following the procedures with or without HIVE are presented in Figure 9a,b, respectively. Black denotes the background, whereas other colors represent individual grains. The grain boundaries acquired using HIVE are closer to the ground truth (Figure 2b), which is slightly over-segmented. Nevertheless, the result demonstrates the important role of the proposed HIVE algorithm in segmentation.

3.4 | Grain Reconstruction

The reconstruction of the initial contour for each grain using the proposed strategy is shown in Figure 10. The result suggests that almost all assigned initial contours are located near the centroid of the grain. It is acknowledged that some of the initial contours are allocated outside the grain. The reason for this is that the size of these grains is smaller than the initial contour. The reconstruction process of a representative grain at five different generations is illustrated in Figure 11. The estimated extent of the reconstructed grain in the binary image is delineated by the red ellipse. The grain surface is represented in the form of triangular meshes using the marching cube algorithm.

3.5 | Robustness

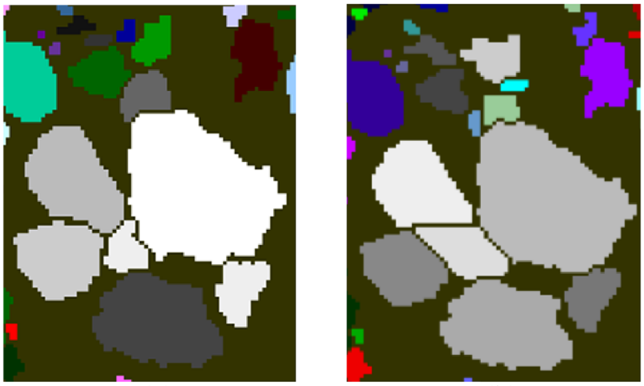
To test the robustness of the trained model in segmentation, a new slice image $z = 870$, far from the selected 100 slice



(a)

(b)

FIGURE 8 | PSO-RF-based model for grain classification: (a) evolution of loss value; and (b) initial classification result.



(a) (b)

FIGURE 9 | Segmentation result: (a) with HIVE; and (b) without HIVE. HIVE, hierarchical interior void elimination.

images, is selected. Following the same operation procedures, the raw image, corresponding segmentation result, and reconstruction result are shown in Figure 12. All particles can be clearly segmented and constructed, indicating the proposed method can accurately identify particles from poor-quality images.

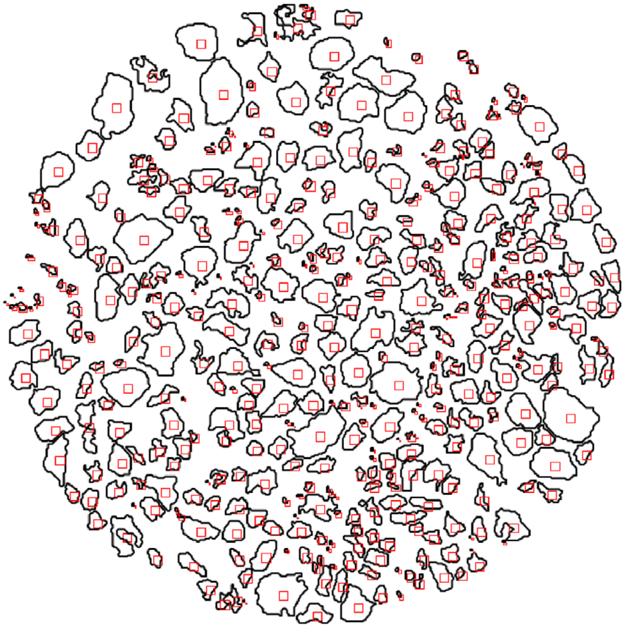


FIGURE 10 | Allocation of initial level sets.

3.6 | Benchmark Study

3.6.1 | Classification and Segmentation

Based on slice $z = 350$, the classification and segmentation performance of the proposed method is compared with the classic watershed algorithm (Figure 13). The proposed approach shows better performance in segmenting arbitrarily large regions with complex boundaries, as the contour of the obtained segments is more reasonable. On the contrary, the result of segmentation using the watershed consists of several unsegmented regions. In addition, as for the watershed algorithm, the problem of over-segmentation is severe.

To quantify the segmentation results, the grains are qualitatively categorized as *A*, *B*, or *C*, according to the image quality, as illustrated in Figure 14a. *A* denotes grains with clear boundaries and texture; *B* denotes grains with a distinguished boundary but with fuzzy texture; finally, *C* denotes grains where both boundary and texture are not clear. Two types of segmentation errors can be identified: incorrect segmentation and over-segmentation.

The schematic diagrams of segmentation results obtained by the watershed algorithm and the proposed approach are shown in Figure 14b,c, respectively. Hollow solid lines denote incorrect segmentation, while hollow dashed lines represent over-segmentation. The segmentation errors of the proposed approach are mainly attributable to over-segmentation. The reason for this is that the grain classifier is not robust enough to guarantee that every decision is correct. Nevertheless, it can still be observed that the proposed method segments most grains correctly. Table 6 summarizes the quantitative analysis of both sets of segmentation results and indicates that nearly 94% of type *A* grains and 83% of type *B* grains are identified correctly by our approach. In addition, 64% of type *C* grains can also be identified. The mean *ACC* in segmenting grains by the proposed approach is 80%, beyond the watershed by 30%. This indicates that the incorporation of the

FIGURE 11 | Process of grain reconstruction using the level set at five different generations.

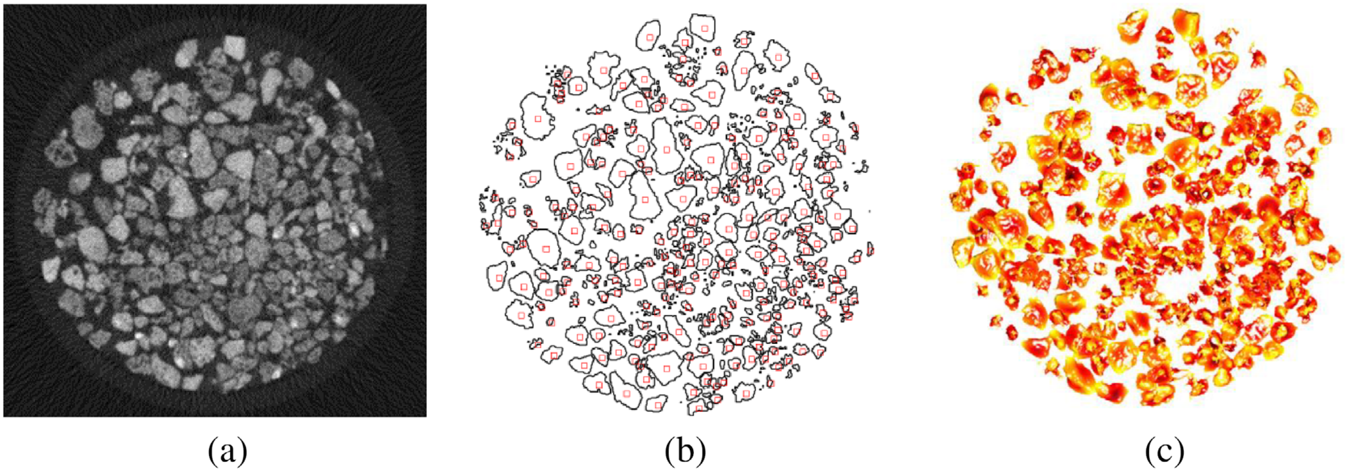

(a)
(b)
(c)

FIGURE 12 | Demonstration of model robustness: (a) raw image; (b) segmentation result; and (c) reconstruction result.

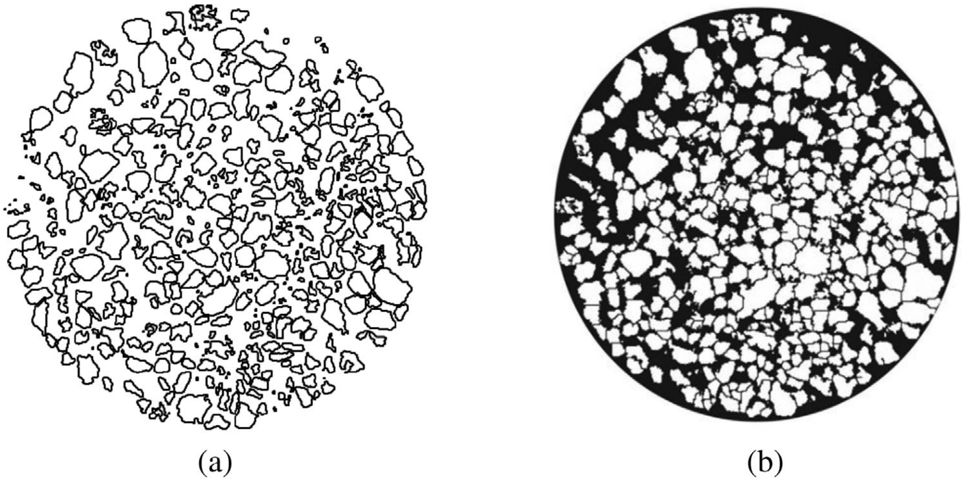

(a)
(b)

FIGURE 13 | Grain segmentation results: (a) proposed; and (b) watershed.

TABLE 6 | Quantitative comparison of segmentation accuracy among two methods.

		TP		*ACC* (%)	
Grain type	***P + N***	**Watershed**	**Proposed**	**Watershed**	**Proposed**
A	125	85	117	68.0	93.6
B	70	43	58	61.4	82.9
C	63	17	40	27.0	63.5
—	—	—	—	**52.1**	**80.0**

Note: Bold fonts denote the mean *ACC* values; $TN = 0$ for three approaches.
Abbreviations: ACC, accuracy; TP, true positives.

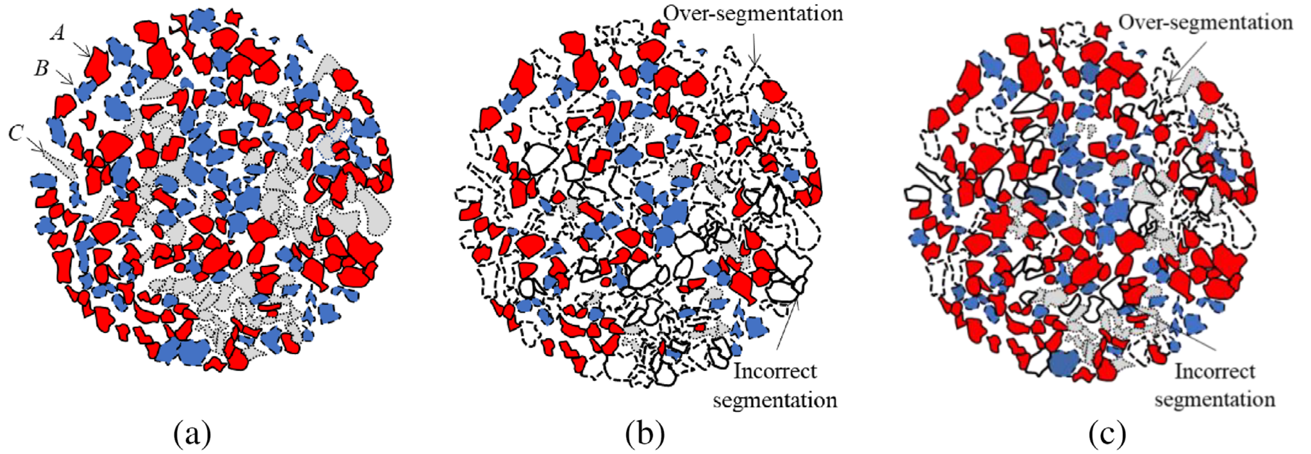


FIGURE 14 | Quantitative analysis segmentation results: (a) real segmentation result; (b) segmented by the watershed; and (c) segmented by our proposed approach. Categories A, B and C are shown in red, blue, and grey, respectively.

TABLE 7 | Evaluation of the reconstruction accuracy of our proposed approach in comparison with the RF-based model and watershed algorithm.

		TP			***ACC* (%)**		
Grain type	***P* + *N***	**PSO-RF**	**Watershed**	**Proposed**	**PSO-RF**	**Watershed**	**Proposed**
A	125	111	73	114	88.8	58.4	91.2
B	70	40	31	57	57.1	44.3	81.4
C	63	15	7	36	23.8	11.1	57.1
—	—	—	—	—	**56.6**	**37.9**	**75.6**

Note: Bold fonts denote the mean *ACC* values; $TN = 0$ for three approaches.
Abbreviations: ACC, accuracy; PSO, particle swarm optimization; RF, random forest; TP, true positives.

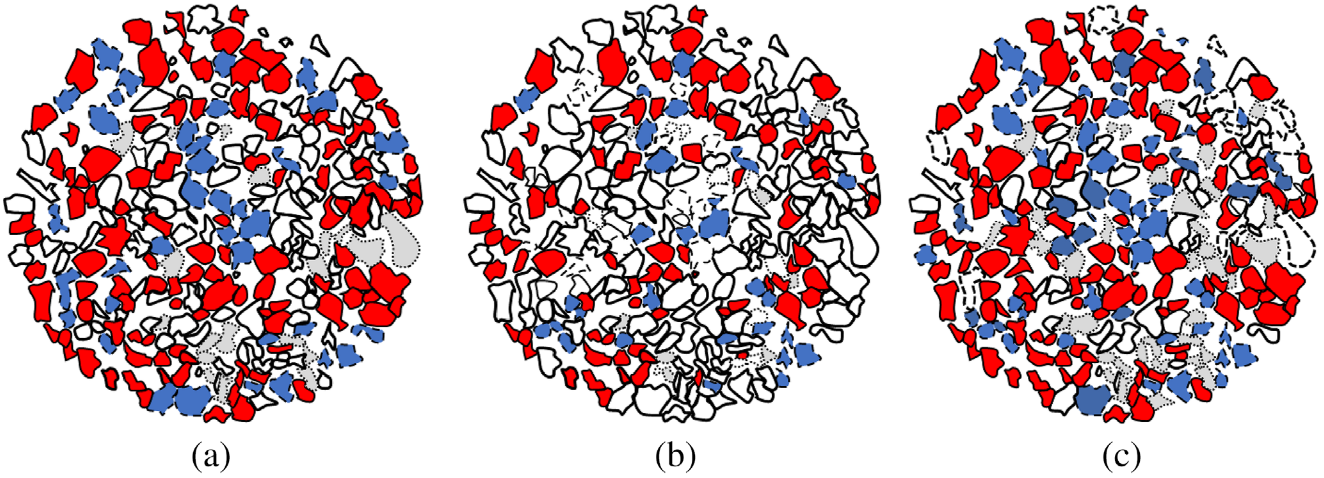


FIGURE 15 | Reconstructed grains based on the segmentation results generated by: (a) PSO-RF-based model; (b) watershed algorithm; and (c) proposed approach.

proposed strategies for determining the splitting path is effective in guaranteeing accurate segmentation.

3.6.2 | Reconstruction

The reconstruction performance of the proposed method is compared with the results presented by Zhang et al. [24], where the reconstruction performance based on the segmentation results obtained by watershed and PSO-RF was clearly quantified, as shown in Figure 15. The hollow grains with a bold contour indicate that such grains are not reconstructed correctly. The statistical results are presented in Table 7. The *ACC* for reconstructing type *A* grains exceeds 90%, which is 30% beyond the *ACC* of the watershed algorithm. Regarding the ratio of reconstructing type *B* and type *C* grains, the results

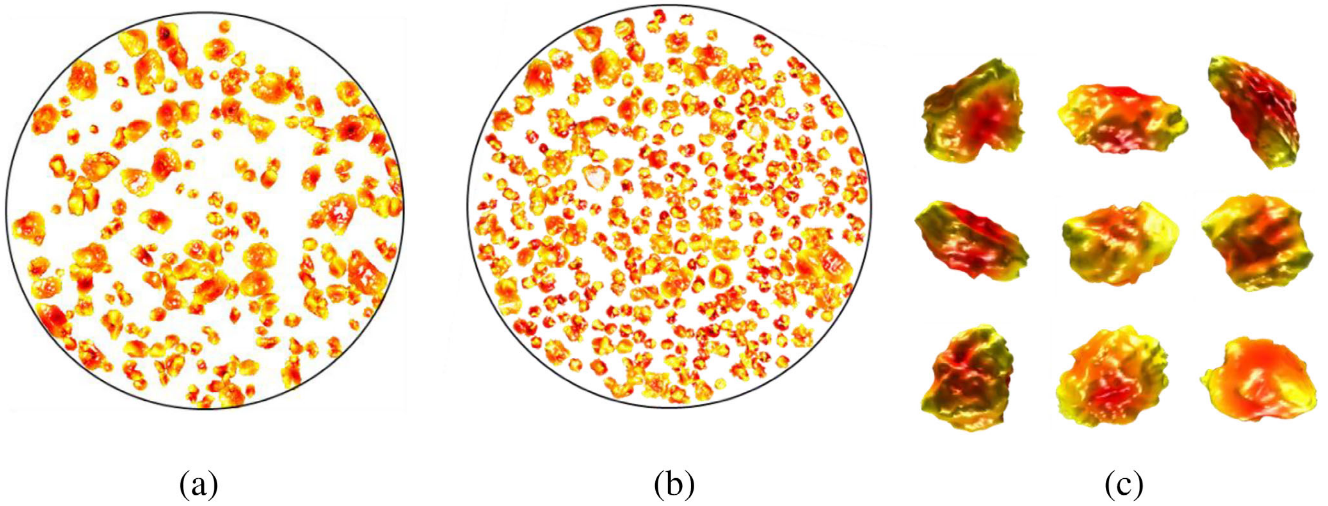


FIGURE 16 | 3D illustrations of reconstructed grains in a slice using: (a) PSO-RF-based model; (b) proposed approach; and (c) our representative grains. PSO, particle swarm optimization; RF, random forest.

TABLE 8 | Comparison of run times for our proposed approach and PSO-RF-based model.

Algorithm	Sample size ($N \times W \times H$)	Pixel numbers	Run time (s)
PSO-RF-based model	100 × 500 × 512	25,600,000	6847
Proposed	100 × 500 × 512	25,600,000	2063

Note: H, width of the image; *N*, number of images; *W*, width of the image.
Abbreviations: PSO, particle swarm optimization; RF, random forest.

indicate that ours outperforms the PSO-RF-based model and watershed algorithm with a significant increase in *ACC*, reaching 81% and 57%, respectively. The mean *ACC* of ours is 75%, which is significantly greater than the other two methods. The reconstructed grains based on the segmentation result of the proposed approach and the PSO-RF-based model are presented in Figure 16a,b. Nine representative grains obtained by our proposed approach are shown in Figure 16c.

3.6.3 | Computational Cost

The computational costs of the proposed approach compared with the PSO-RF-based approaches [24] using a PC with an NVIDIA GTX 3090 GPU, 12th Gen Intel(R) Core(TM) i9-12900K CPU, and 80 GB RAM in a Z690M AORUS ELITE board are presented in Table 8. The computational cost of the present approach is 2063 s which is significantly faster than the PSO-RF-based model (6847 s). This demonstrates the efficiency of the proposed scheme in reconstructing granular grains again.

4 | Conclusions

This study has described a ML based framework to automatically reconstruct 3D granular grains from poor-quality μCT images. The hybrid ML-based algorithm integrating particle swarm optimization (PSO) and RF was first adopted to distinguish grains and other constituents. Thereafter, the intergranular and intragranular voids were distinguished by criteria based on area and concavity and eliminated by filling. Moreover, a PSO-RF-based grain classifier was trained for classifying grain clumps and individual grains with inputs of seven shapes, that is, circularity. For grain clumps, they were iteratively segmented into sub-grains through a concavity analysis-based segmentation technique. The final segmentation accuracy was 80%, achieving a step-change improvement over conventional methods.

Subsequently, an adaptive hierarchical approach was proposed to allocate the initial level set for grains for 3D reconstruction. This method enabled the automatic activation of the level set and grain reconstruction. For three types of grains: clear boundaries and textures, clear boundaries and fuzzy textures, and unclear boundaries and textures, the reconstruction ratios were 90%, 80%, and 60%, respectively. This proposed approach increased the reconstruction rate by 20% over previous methods, and the computational efficiency improved three times.

Acknowledgments

This research was financially supported by the Research Grants Council (RGC) of Hong Kong Special Administrative Region Government (HKSARG) of China (Grant No.: 15220221) and the Research Centre for Resources Engineering towards Carbon Neutrality (RCRE) of The Hong Kong Polytechnic University (No. 1-BBEM). The second author is supported by the Royal Society under the Newton International Fellowship. The fourth author is supported by the Royal Academy of Engineering under the Research Fellowship Scheme.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

1. J. Shen, X. Wang, Y. Shan, et al., "Particle Breakage and Shape Analysis of Calcareous Sand Under Consolidated-Undrained Triaxial

Shear,” *Bulletin of Engineering Geology and the Environment* 81, no. 6 (2022): 232, https://doi.org/10.1007/s10064-022-02685-0.

2. X. Wang, Y. Wang, C. Liao, et al., “Particle Breakage Mechanism and Particle Shape Evolution of Calcareous Sand Under Impact Loading,” *Bulletin of Engineering Geology and the Environment* 81, no. 9 (2022): 372, https://doi.org/10.1007/s10064-022-02868-9.

3. S. Zhao, G. Zhang, D. Zhang, D. Tan, and H. Huang, “A Hybrid Attention Deep Learning Network for Refined Segmentation of Cracks From Shield Tunnel Lining Images,” *Journal of Rock Mechanics and Geotechnical Engineering* 15, no. 12 (2023): 3105–3117.

4. M. Beister, D. Kolditz, and W. A. Kalender, “Iterative Reconstruction Methods in X-Ray CT,” *Physical Medicine* 28, no. 2 (2012): 94–108, https://doi.org/10.1016/j.ejmp.2012.01.003.

5. B. Zhou and J. Wang, “Generation of a Realistic 3D Sand Assembly Using X-Ray Micro-Computed Tomography and Spherical Harmonic-Based Principal Component Analysis,” *International Journal for Numerical and Analytical Methods in Geomechanics* 41, no. 1 (2017): 93–109.

6. F. N. Altuhafi, M. R. Coop, and V. N. Georgiannou, “Effect of Particle Shape on the Mechanical Behavior of Natural Sands,” *Journal of Geotechnical and Geoenvironmental Engineering* 142, no. 12 (2016): 04016071, https://doi.org/10.1061/(ASCE)GT.1943-5606.0001569.

7. M. R. Cox and M. Budhu, “A Practical Approach to Grain Shape Quantification,” *Engineering Geology* 96, no. 1–2 (2008): 1–16, https://doi.org/10.1016/j.enggeo.2007.05.005.

8. S. K. Dwivedi, M. Vishwakarma, and A. Soni, “Advances and Researches on Non Destructive Testing: A Review,” *Materials Today: Proceedings* 5, no. 2 pt. 1 (2018): 3690–3698, https://doi.org/10.1016/j.matpr.2017.11.620.

9. Z. Lai and Q. Chen, “Reconstructing Granular Particles From X-Ray Computed Tomography Using the TWS Machine Learning Tool and the Level Set Method,” *Acta Geotechnica* 14, no. 1 (2019): 1–18, https://doi.org/10.1007/s11440-018-0759-x.

10. D. Su and W. M. Yan, “3D Characterization of General-Shape Sand Particles Using Microfocus X-Ray Computed Tomography and Spherical Harmonic,” *Powder Technology* 323 (2018): 8–23, https://doi.org/10.1016/j.powtec.2017.09.030.

11. A. Kornilov, I. Safonov, and I. Yakimchuk, “A Review of Watershed Implementations for Segmentation of Volumetric Images,” *Journal of Imaging* 8, no. 5 (2022): 127, https://doi.org/10.3390/jimaging8050127.

12. S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, “Image Segmentation Using Deep Learning: A Survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, no. 7 (2021): 3523–3542, https://doi.org/10.1109/TPAMI.2021.3059968.

13. N. R. Pal and S. K. Pal, “A Review on Image Segmentation Techniques,” *Pattern Recognition* 26, no. 9 (1993): 1277–1294, https://doi.org/10.1016/0031-3203(93)90135-J.

14. E. H. Van den Berg, A. Meesters, J. A. M. Kenter, and W. Schlager, “Automated Separation of Touching Grains in Digital Images of Thin Sections,” *Computers & Geosciences* 28, no. 2 (2002): 179–190, https://doi.org/10.1016/S0098-3004(01)00038-3.

15. R. Kohler, “A Segmentation System Based on Thresholding,” *Computer Graphics and Image Processing* 15, no. 4 (1981): 319–338, https://doi.org/10.1016/S0146-664X(81)80015-9.

16. S. Beucher, “Watershed, Hierarchical Segmentation and Waterfall Algorithm,” in *Mathematical Morphology and Its Applications to Signal Processing*, eds. J. Serra and P. Soille (Dordrecht: Computational Imaging and Vision, 1994), 69–76.

17. L. Vincent and P. Soille, “Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13, no. 06 (1991): 583–598.

18. D. Kong and J. Fonseca, “Quantification of the Morphology of Shelly Carbonate Sands Using 3D Images,” *Géotechnique* 68, no. 3 (2018): 249–261, https://doi.org/10.1680/jgeot.16.P.278.

19. Y. Shi and W. M. Yan, “Segmentation of Irregular Porous Particles of Various Sizes From X-Ray Microfocus Computer Tomography Images Using a Novel Adaptive Watershed Approach,” *Géotechnique Letters* 5, no. 4 (2015): 299–305, https://doi.org/10.1680/jgele.15.00100.

20. Q. Sun, J. Zheng, and C. Li, “Improved Watershed Analysis for Segmenting Contacting Particles of Coarse Granular Soils in Volumetric Images,” *Powder Technology* 356 (2019): 295–303, https://doi.org/10.1016/j.powtec.2019.08.028.

21. X. Zhang, F. Jia, S. Luo, G. Liu, and Q. Hu, “A Marker-Based Watershed Method for X-Ray Image Segmentation,” *Computer Methods and Programs in Biomedicine* 113, no. 3 (2014): 894–903, https://doi.org/10.1016/j.cmpb.2013.12.025.

22. S. Zhao, D. Tan, S. Lin, Z. Yin, and J. Yin, “A Deep Learning-Based Approach With Anti-Noise Ability for Identification of Rock Microcracks Using Distributed Fibre Optic Sensing Data,” *International Journal of Rock Mechanics and Mining Sciences* 170 (2023): 105525.

23. S. B. Cevallos, A. X. Jerves, U. Torgersrud, et al., “Towards a More Accurate Characterization of Granular Media 2.0: Involving AI in the Process,” *Computers and Geotechnics* 160 (2023): 105510, https://doi.org/10.1016/j.compgeo.2023.105510.

24. P. Zhang, Z.-Y. Yin, and Q. Chen, “Image-Based 3D Reconstruction of Granular Grains via Hybrid Algorithm and Level Set With Convolution Kernel,” *Journal of Geotechnical and Geoenvironmental Engineering* 148, no. 5 (2022): 04022021, https://doi.org/10.1061/(ASCE)GT.1943-5606.0002790.

25. G. Biau and E. Scornet, “A Random Forest Guided Tour,” *Test: An Official Journal of the Spanish Society of Statistics and Operations Research* 25, no. 2 (2016): 197–227, https://doi.org/10.1007/s11749-016-0481-7.

26. D. Wang, D. Tan, and L. Liu, “Particle Swarm Optimization Algorithm: An Overview,” *Soft Computing* 22, no. 2 (2018): 387–408, https://doi.org/10.1007/s00500-016-2474-6.

27. P. Zhang, Z.-Y. Yin, Y.-F. Jin, and T. H. T. Chan, “A Novel Hybrid Surrogate Intelligent Model for Creep Index Prediction Based on Particle Swarm Optimization and Random Forest,” *Engineering Geology* 265 (2020): 105328, https://doi.org/10.1016/j.enggeo.2019.105328.

28. L. Breiman, “Random Forests,” *Machine Learning* 45, no. 1 (2001): 5–32, https://doi.org/10.1023/A:1010933404324.

29. J. Kennedy and R. Eberhart, “Particle Swarm Optimization.” in *Proceedings of ICNN'95 - International Conference on Neural Networks*, (Perth, WA, Australia: IEEE, 1995), 1942–1948.

30. R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB* (London, UK: Gatesmark Publishing, 2004).

31. E. F. Moore, “Machine Models of Self-Reproduction,” in Proceedings of Symposia in Applied Mathematics, ed. R. E. Bellman (Rhode Island: American Mathematical Society, 1962), 17–33.

32. S. Schlüter, U. Weller, and H.-J. Vogel, “Segmentation of X-Ray Microtomography Images of Soil Using Gradient Masks,” *Computers & Geosciences* 36, no. 10 (2010): 1246–1251, https://doi.org/10.1016/j.cageo.2010.02.007.

33. Y. Wang, C. L. Lin, and J. D. Miller, “Quantitative Analysis of Exposed Grain Surface Area for Multiphase Particles Using X-Ray Microtomography,” *Powder Technology* 308 (2017): 368–377, https://doi.org/10.1016/j.powtec.2016.11.047.

34. D. H. Douglas and T. K. Peucker, “Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or Its Caricature,” *Cartographica: International Journal of Geographical Information Geovisualization* 10, no. 2 (1973): 112–122, https://doi.org/10.3138/FM57-6770-U75U-7727.

35. U. Ramer, “An Iterative Procedure for the Polygonal Approximation of Plane Curves,” *Computer Graphics and Image Processing* 1, no. 3 (1972): 244–256, https://doi.org/10.1016/S0146-664X(72)80017-0.

36. D. K. Prasad and M. K. Leung, “Polygonal Representation of Digital Curves,” in *Digital Image Processing*, eds. S. G. Stanciu (New Delhi, India: CBS Publishers & Distributors Pvt. Ltd., 2012), 71–90.

37. S. Zafari, T. Eerola, J. Sampo, H. Kälviäinen, and H. Haario, “Comparison of Concave Point Detection Methods for Overlapping Convex Objects Segmentation,” in *Scandinavian Conference on Image Analysis*, (Dordrecht, the Netherlands: Springer, 2017), 245–256.

38. W.-H. Zhang, X. Jiang, and Y.-M. Liu, “A Method for Recognizing Overlapping Elliptical Bubbles in Bubble Image,” *Pattern Recognition Letters* 33, no. 12 (2012): 1543–1548, https://doi.org/10.1016/j.patrec.2012.03.027.

39. J. E. Bresenham, “Algorithm for Computer Control of a Digital Plotter,” *IBM Systems Journal* 4, no. 1 (1965): 25–30.

40. T. Janssens, L. Antanas, S. Derde, I. Vanhorebeek, G. Van den Berghe, and F. G. Grandas, “CHARISMA: An Integrated Approach to Automatic H&E-Stained Skeletal Muscle Cell Segmentation Using Supervised Learning and Novel Robust Clump Splitting,” *Medical Image Analysis* 17, no. 8 (2013): 1206–1219, https://doi.org/10.1016/j.media.2013.07.007.

41. H. Wang, H. Zhang, and N. Ray, “Clump Splitting Via Bottleneck Detection and Shape Classification,” *Pattern Recognition* 45, no. 7 (2012): 2780–2787, https://doi.org/10.1016/j.patcog.2011.12.020.

42. F. Attneave and M. D. Arnoult, “The Quantitative Study of Shape and Pattern Perception,” *Psychological Bulletin* 53, no. 6 (1956): 452–471.

43. W. X. Wang, “Binary Image Segmentation of Aggregates Based on Polygonal Approximation and Classification of Concavities,” *Pattern Recognition* 31, no. 10 (1998): 1503–1524, https://doi.org/10.1016/S0031-3203(97)00145-3.

44. S. Zafari, T. Eerola, J. Sampo, H. Kälviäinen, and H. Haario, “Segmentation of Partially Overlapping Nanoparticles Using Concave Points,” in *Lecture Notes in Computer Science*, eds. G. Bebis, R. Boyle, B. Parvin, et al., (Cham: Springer International Publishing, 2015): 187–197.

45. S. Kumar, S. H. Ong, S. Ranganath, T. C. Ong, and F. T. Chew, “A Rule-Based Approach for Robust Clump Splitting,” *Pattern Recognition* 39, no. 6 (2006): 1088–1098, https://doi.org/10.1016/j.patcog.2005.11.014.

46. E. W. Dijkstra, “A Note on Two Problems in Connexion With Graphs,” *Numerische Mathematik* 1, no. 1 (1959): 269–271, https://doi.org/10.1007/BF01386390.

47. D. B. Johnson, “A Note on Dijkstra’s Shortest Path Algorithm,” *Journal of the ACM* 20, no. 3 (1973): 385–388, https://doi.org/10.1145/321765.321768.

48. K. R. Castleman, *Digital Image Processing* (Hoboken, NJ: Prentice Hall Press, 1996).

49. D. Jacobs, Image Gradients: Class Notes for CMSC (College Park, MD: Fall, 2005), 1–3.

50. A. Leonti, J. Fonseca, I. Valova, et al., “Optimized 3D Segmentation Algorithm for Shelly Sand Images,” in *Proceedings of the 6th World Congress on Electrical Engineering and Computer Systems and Science*, (CIST, 2020), Virtual Conference.

51. C. Li, C. Xu, C. Gui, and M. D. Fox, “Distance Regularized Level Set Evolution and Its Application to Image Segmentation,” *IEEE Transactions on Image Processing* 19, no. 12 (2010): 3243–3254, https://doi.org/10.1109/TIP.2010.2069690.

52. F. Gibou, R. Fedkiw, and S. Osher, “A Review of Level-Set Methods and Some Recent Applications,” *Journal of Computational Physics* 353 (2018): 82–109, https://doi.org/10.1016/j.jcp.2017.10.006.

53. S. Osher and J. A. Sethian, “Fronts Propagating With Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations,” *Journal of Computational Physics* 79, no. 1 (1988): 12–49, https://doi.org/10.1016/0021-9991(88)90002-2.

54. W. E. Lorensen and H. E. Cline, “Marching Cubes: A High Resolution 3D Surface Construction Algorithm,” *ACM SIGGRAPH Computer Graphics* 21, no. 4 (1987): 163–169, https://doi.org/10.1145/37402.37422.

55. C. Li, C. Xu, C. Gui, and M. D. Fox, “Level Set Evolution Without Re-initialization: A New Variational Formulation,” in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) (San Diego, CA: IEEE, 2005), 430–436.

56. P. Probst, M. N. Wright, and A. Boulesteix, “Hyperparameters and Tuning Strategies for Random Forest,” *WIREs Data Mining and Knowledge Discovery* 9, no. 3 (2019): e1301, https://doi.org/10.1002/widm.1301.