Computers and Geosciences 123 (2019) 83–94

Contents lists available at ScienceDirect

Computers and Geosciences

journal homepage: www.elsevier.com/locate/cageo

An integrated method for 3D reconstruction model of porous geomaterials through 2D CT images

Zhi Zhao[a,1], Xiao-Ping Zhou[a,b,*,2]

[a] School of Civil Engineering, Chongqing University, Chongqing, 400045, China
[b] State Key Laboratory of Coal Mine Disaster Dynamics and Control, Chongqing University, Chongqing, 400045, China

ARTICLE INFO

Keywords:
Sandstone
CT images
Multiple point statistics
Three-dimensional reconstruction
Pore structure extraction
Connectivity
Permeability

ABSTRACT

Three-dimensional reconstruction technique plays a key role in understanding the characteristics of geomaterial pore structures through computed tomography images. In this paper, multiple-point statistics and a marching cube algorithm are integrated to reconstruct three-dimensional model to extract the pore structure, analyze connectivity and predict the permeability of the pore structure. This method is applicable not only to two-dimensional images to determine simple characteristics of geomaterials, but also to better understand inner pore structure via the reconstructed three-dimensional models. The results obtained by the proposed reconstruction method agree well with those obtained from the previous methods.

1. Introduction

The characteristics of pore structures of geomaterials plays a vital role in analyzing petrophysical and transport parameters such as porosity, specific surface area, permeability and capillary pressure (Øren and Bakke, 2003; Al-Raoush and Willson, 2005). 3D reconstruction techniques offer efficient approaches for investigating such typical parameters using high resolution CT images. With the development of image processing technologies, computed tomography has been shown to be superior in defining the 3D pore structure of geomaterials by generating micron-scale 3D data sets (Knackstedt et al., 2004; Ohnishi et al., 2014). The ability to visualize 3D pore structures accelerates the reliability and understanding of calculation results based on the CT data (Arns et al., 2004; Olafuyi et al., 2006). To generate a 3D pore network model and describe the pore structure of geomaterials, a suitable method should be considered due to the interference from the resolution of CT images. Generally, models representing the 3D models of geomaterials have been classified into three types (Al-Kharusi and Blunt, 2007; Rabbani et al., 2014). The first interprets sedimentary rock processes using the measured distribution of grain centers. However, it computes grain sizes inefficiently, and grain locations are difficult to validate (Øren, and Bakke, 2003). The second represents 3D models using the statistical properties of thin sections, and are divided into two algorithms. One reconstructs the 3D realization using a two-point correlation function (Ioannidis and Chatzis, 2000). However, its structural representation of sphere packing is poor. The other utilizes a multiple-point statistical method. However, its applications still are constrained, and the models are pseudo-3D images showing large differences between the reconstructed model and original sample (Okabe and Blunt, 2005). The third creates 3D realizations using the direct outputs of 3D CT scanning tools. However, the microstructure reproduced suffers from a high random access memory (RAM) usage and high cost of computation time for large datasets (Ohnishi et al., 2014). Fortunately, parallel computing (CPU/GPU) approach can accelerate this procedure with sharing the cache and memory (Straubhaar et al., 2013; Zhang et al., 2015). However, it inevitability suffers from high economic cost for computing architecture of super computer and complex multi-thread computing models.

Once 3D models are reconstructed, petrophysical and transport parameters can be predicted. Recently, the prediction of such parameters has improved through the simulation of pore network extraction (Øren and Bakke, 2003; Bashtani et al., 2016; Sharqawy, 2016). An initial rigorous pore network extraction method was defined by Lindquist et al. (1996), in which pore space distributions were quantitatively discussed by the defined medial axis. Several other methods have also shown promise for determining the pore structure by the pore space distances and maximally inscribed spheres (Sweeney and Martin, 2003; Silin and Patzek, 2006; Arand and Hesser, 2017), and by the

* Corresponding author. School of Civil Engineering, Chongqing University, Chongqing, 400045, China.
E-mail address: cqxpzhou@hotmail.com (X.-P. Zhou).
[1] Zhi Zhao wrote the implementation codes for this manuscript.
[2] Xiao-Ping Zhou conducted all the experiments, and provided valuable suggestions on algorithms.

https://doi.org/10.1016/j.cageo.2018.11.012
Received 5 May 2018; Received in revised form 19 November 2018; Accepted 28 November 2018
Available online 29 November 2018
0098-3004/

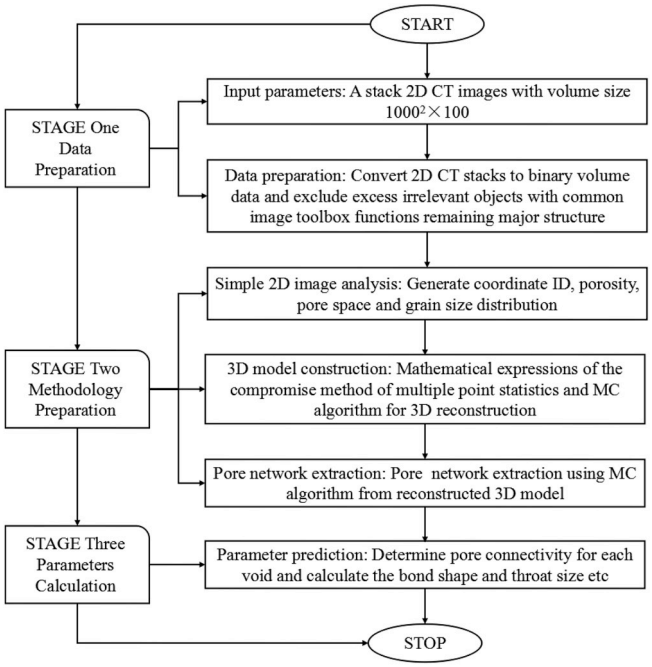


Fig. 1. The flow chart of the whole methodology for 3D reconstruction and pore network extraction.

extracted skeletonization models (Al-Raoush and Willson, 2005; Homberg et al., 2012; Wildenschild and Sheppard, 2013). Although both can provide reliable pore structure information, each is inevitably affected by the digitization errors during pores and throats identification (Byholm et al., 2006).

In this paper, multiple point statistics (MPS) and a marching cube algorithm (MC) are integrated to reconstruct a 3D visualization model of sandstone. The main purpose of this paper is to decrease the effect of digitization errors of 3D spatial information of micro structure. Moreover, we make efforts to decrease the computing time, RAM usage and economic cost by using a strut array representation of datasets, similar to the parallel computing approaches with CPU and GPU. Compared with previous methods, the extraction of the pore network structure is similar to the classes of methods mentioned above with pore structure represented by pore space distances and maximally inscribed spheres. However, the proposed method accurately simulates target images organized into struct arrays, and has the following advantages: (1) the classification and clustering of training image are not needed; (2) the computing efficiency significantly improves due to the binary-decimal transformation of the pattern library; (3) the retrieval of the pattern library is streamlined by the utilization of a sub-pattern; (4) the records of all patterns in each training image is replaced by the records of the central location of each pattern, and RAM usage is decreased.

This paper is organized as follows. The MPS and MC algorithms are integrated to reconstruct the 3D model as described in Section 2. The reconstructed 3D model and extracted pore structure are described in Section 3, and a brief summary and conclusions are given in Section 4.

2. Experiment and methodology

The 3D data were obtained from a Siemens-Somatom scope CT scanner by rotating the Chongqing sandstone samples (Φ50 × 100 mm) through a full 360° range at intervals of 0.2°. A total number of 100

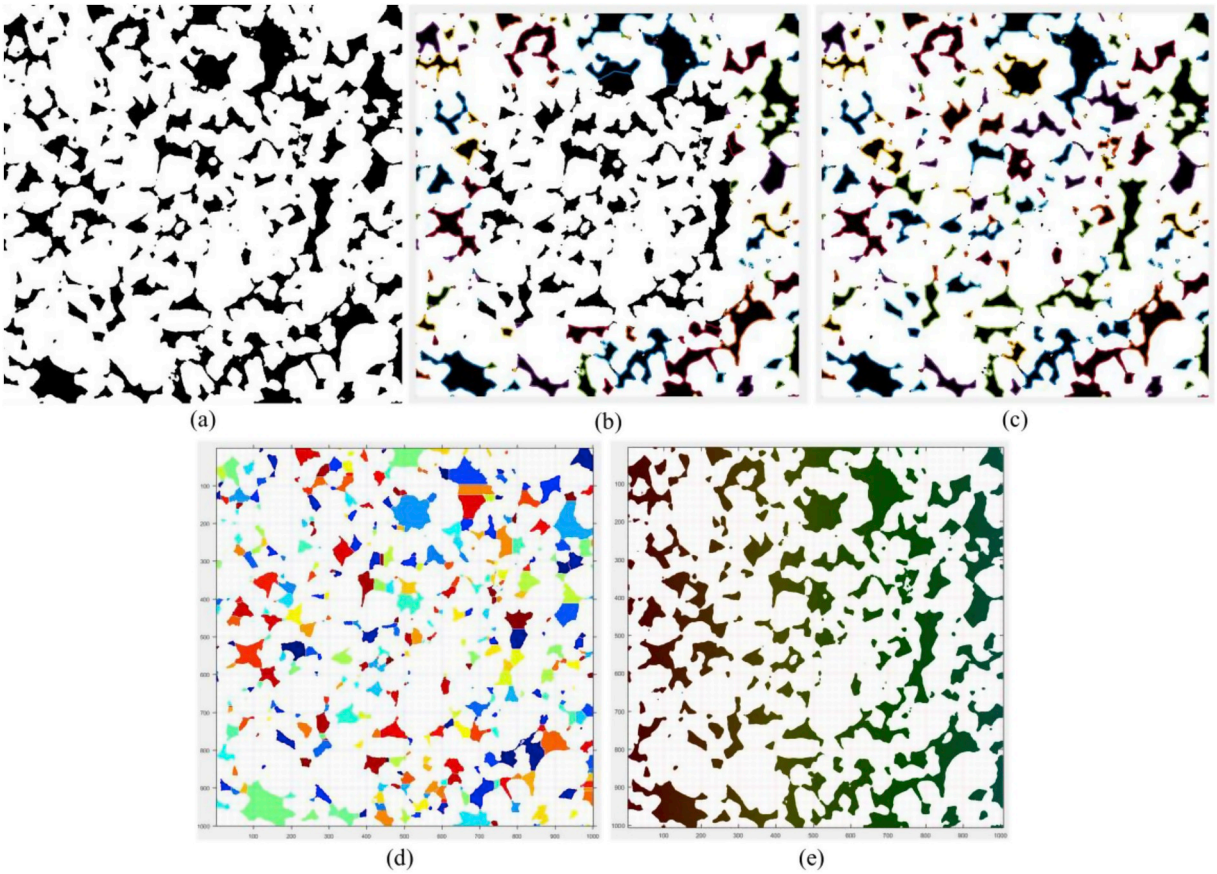


Fig. 2. Data preparation process, (a) the original CT images with 1000^2 pixels, (b) the middle procedure and (c) the recognition of pore (color) and solid objects (white), (d) the segmented 2D image, in which the sizes of pores are represented in different colors, (e) the 3D realization of single CT image. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

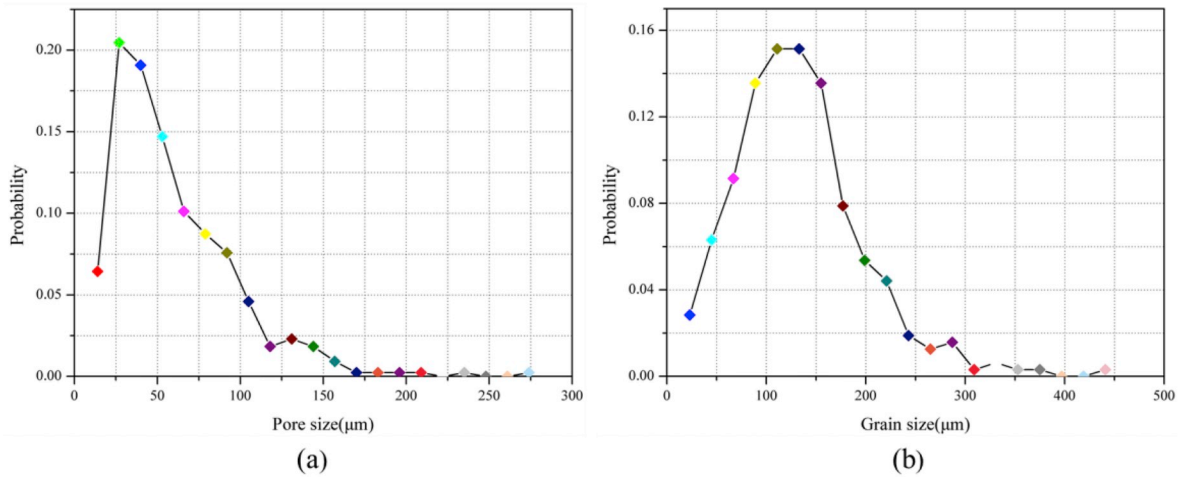
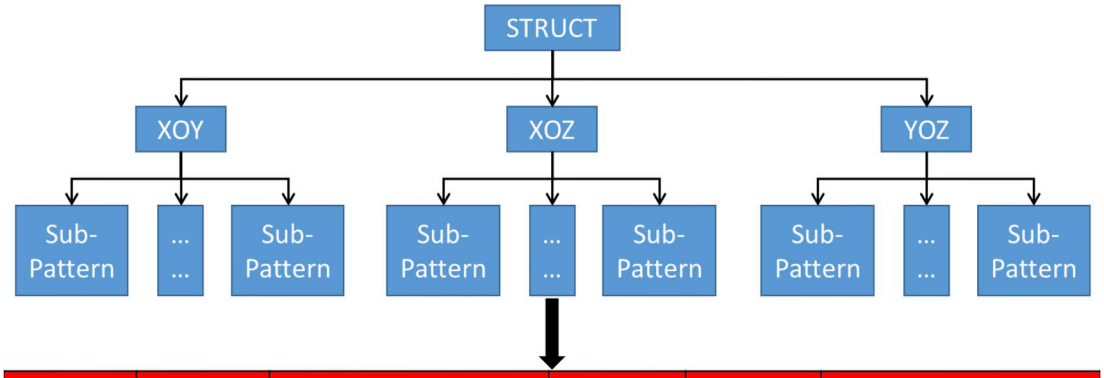


Fig. 3. Probability distributions of pore and grain sizes of an CT image.



Column1	Column2		Column3	Column4	Column5		Column6
B-Pattern	Counters		Coordinate ID	D-Pattern	Counters		Coordinate ID
	0	1			0	1	
1011	1	1	(1,1,m△z),(1,3, m△z)	11	1	1	(1,1,m△z),(1,3, m△z)
0100	1	0	(2,1, m△z)	4	1	0	(2,1, m△z)
.....	.....	.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....	.....	.....
1111	1	0	(4,4, m△z)	15	1	0	(4,4, m△z)

Fig. 4. The struct representation of pattern library.

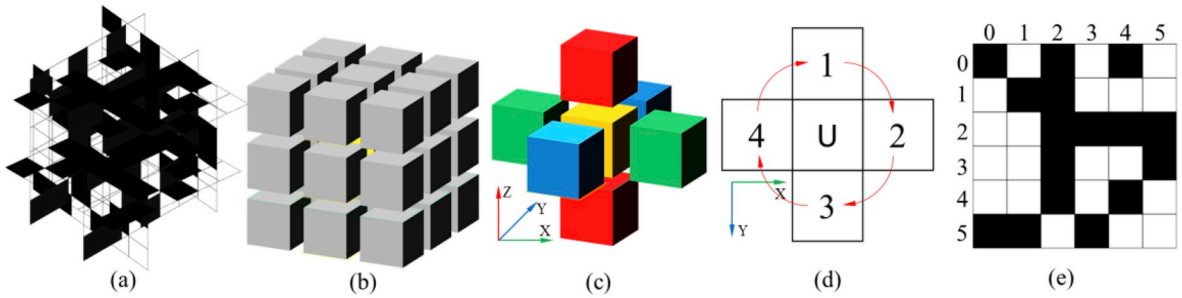


Fig. 5. Demonstration of the scanning procedure using template; (a) 3D 6 × 6 training image, (b) 3D template representation with yellow central node U, (c) simplified 3D template in XOY plane, (d) simplified 2D 3 × 3 template, (e) 6 × 6 training image in XOY plane. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

Table 1
Struct array representation of the training image and template in XOY plane in Fig. 5.

B-Pattern	Counters		Coordinate ID	D-Pattern	Counters		Coordinate ID
	0	1			0	1	
1011	1	1	(1,1,m△z), (1,3,m△z)	11	1	1	(1,1, m△z)
0100	1	0	(2,1, m△z)	4	1	0	(2,1, m△z)
1100	0	1	(3,1, m△z)	12	0	1	(3,1, m△z)
0101	1	1	(4,1, m△z), (2,3, m△z)	5	1	1	(4,1, m△z)
0011	0	1	(1,2, m△z),	3	0	1	(1,2, m△z),
0001	1	1	(2,2, m△z), (4,3, m△z)	1	1	1	(2,2, m△z)
1010	2	0	(3,2, m△z), (4,2, m△z)	10	2	0	(3,2, m△z)
0110	0	1	(3,3, m△z)	6	0	1	(3,3, m△z)
1001	0	1	(1,4, m△z)	9	0	1	(1,4, m△z)
0111	1	0	(2,4, m△z)	7	1	0	(2,4, m△z)
1000	0	1	(3,4, m△z)	8	0	1	(3,4, m△z)
1111	1	0	(4,4, m△z)	15	1	0	(4,4, m△z)

Notice: B-Pattern is defined as binary pattern; D-pattern represents the decimal pattern.

slices were produced and cropped at a scale of $1000^2 \times 100$ with a resolution of 3 μm per-pixel.

Furthermore, to reduce the digitization errors and identification difficulty of pores and throats, the methodology is divided into four steps:

(1) 2D image preprocessing is applied to reduce the identification difficulty of pores or throats, and a quantitative analysis is implemented.
(2) The integrated method of MPS and MC algorithms is applied to reconstruct the 3D model.
(3) The 3D solid and skeletonization models are reconstructed, and used to extract the pore structure.
(4) The petrophysical parameters and transport properties are predicted by the extracted pore structure using MC algorithm.

2.1. Characterization of 2D CT slice

Data preparation step is firstly required to exclude excess pores and throats, which are considered as noise in rock CT images. This procedure is shown in the stage 1 (Fig. 1). Then, the histogram estimation method, which belongs to the threshold-based segmentation method, is applied to segment the CT images due to the unimodal distribution of gray level of CT images (Jonassen et al., 2008), and these images are converted to binary images with 0 (black) for voids and 1 (white) for solid objects using threshold segmentation function

$$f_b(x, y) = \begin{cases} 1, & f_g(x, y) \geq T \\ 0, & otherwise \end{cases} \tag{1}$$

where $f_b(x, y)$ and $f_g(x, y)$ are the binary value and gray level of each pixel, respectively, and T is the threshold value used in the pixel-based binary conversion, which is the average value of $f_g(x, y)$ obtained from the histogram estimation method.

Once the suitable threshold value (the average value) is determined using the histogram estimation method, the images can be segmented reasonably and reliably with considering the function of all pixels. In addition, accurate pore network can be obtained based on the segmented image. Finally, we excluded a few tiny irrelevant pixels while retaining the primary structural information using median filtering function.

The methodology preparation in the stage 2 is crucial for the success of the whole reconstruction scheme. For simplification, we classify this stage into three steps: (1) the coordinate ID, porosity, pore and grain sizes are extracted by simple 2D image analysis; (2) the integrated method of MPS and MC algorithms is used to reconstruct the 3D model; (3) the pore network is extracted from the reconstructed model.

In this work, to compute the pore and grain size distributions, the Chan-Vese model (Chan and Vese, 2001) is used to recognize each pore and solid object for given random CT images. Based on the Chan-Vese model, the energy function of the object is written as

$$E(\Omega, f_1, f_2) = \iint_{\Omega_{inside}} |I - f_1|^2 dxdy + \iint_{\Omega_{outside}} |I - f_2|^2 dxdy + \mu L(\Omega_{outside}) \tag{2}$$

where $I(x, y)$ are gray level intensities $f(x, y)$ with each coordinate ID, $L(\Omega)$denotes the contour line length of each object, $\Omega$is the contour line in the original images, and f_1 and f_2 are the different average gray levels inside and outside the contour line.

When the contour line Ω is replaced by the level set function Φ, Eq. (2) can be rewritten as

$$E\left(\Phi, f_1, f_2\right) = \iint_{\Omega} |I - f_1|^2 H(\Phi) dxdy + \iint_{\Omega} |I - f_2|^2 \Big(1 - H(\Phi)\Big) dxdy + \iint_{\Omega} \delta(\Phi) |\nabla \Phi| dxdy \tag{3}$$

where H denotes the Heaviside function, δ denotes the Dirac function, and the expressions of f_1 and f_2 are (Chan and Vese, 2001)

$$\begin{cases} f_1(\Phi) = \iint_{\Omega} I(x, y) H(\Phi) dxdy / \iint_{\Omega} H(\Phi) dxdy \\ f_2(\Phi) = \iint_{\Omega} I(x, y)(1 - H(\Phi)) dxdy / \iint_{\Omega} (1 - H(\Phi)) dxdy \end{cases} \tag{4}$$

Thus, reliable information on pores and solid objects can be acquired efficiently from Eq. (2)-Eq. (4), and the results are shown in Fig. 2.

Once the pores and solid objects are separated, the probability distribution of pores (PSD) and grain sizes (GSD) are estimated from

$$PSD = \frac{\sum_{\mathbb{R}_{color}} N\left[f_{\mathbb{R}}(x, y)\right]}{\sum_{\mathbb{R}_{color}} N\left[f_b(x, y)\right]} \times 100\% \tag{5}$$

where$N$is the pixel counter, $f_{\mathbb{R}}(x, y)$ is gray level of a single color set, $f_b(x, y)$ is the binary value of each pixel, and $\mathbb{R}_{color}$ is the pore or grain subset represented by different colors.

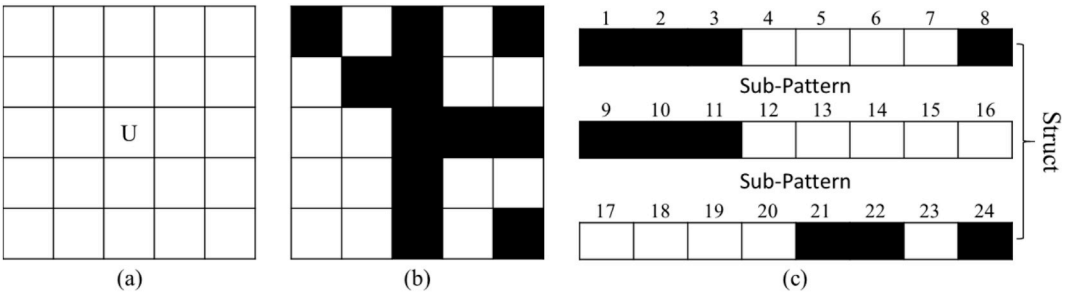


Fig. 6. The generation of sub-pattern library; (a) 5 × 5 template, (b) a 5 × 5 training image, (c) the struct array of sub-patterns.

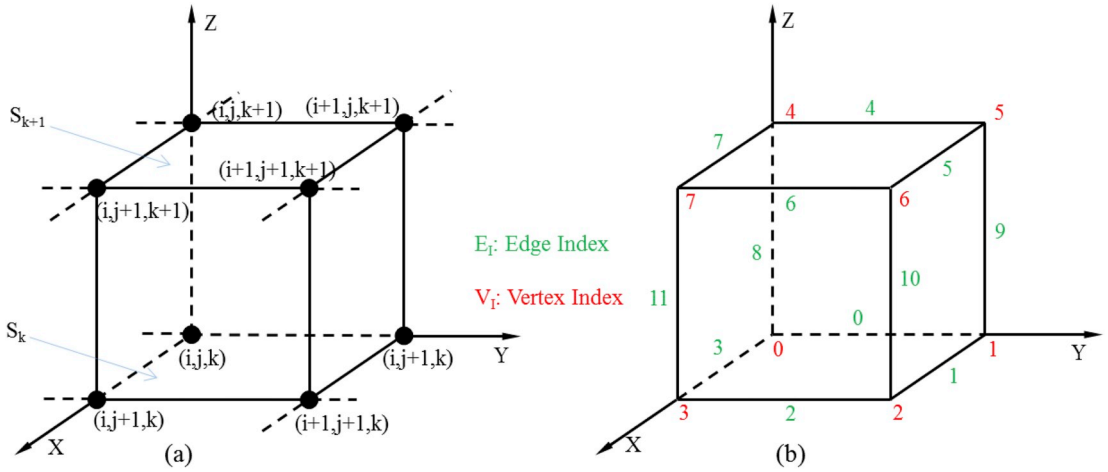


Fig. 7. (a) Demonstration of a formatted cube, (b) sketch map of Edge and Vertex index.

Considering the complicated internal pore structure, and the difference between the model and real sample, some simple assumptions must be made to obtain accurate pore and grain sizes. Here, irregular pores are assumed to be circles in 2D space. The assumed circle diameters are considered to be the same as the real pore diameters. Thus, the pore or grain radii (Fig. 3) can be calculated from the 2D cross-sections by

$$R = P_{resolution} \times \frac{\sqrt{N[f_R(x,y)]}}{\sqrt{\pi}} \tag{6}$$

where R is the radius of the pore or grain, and $P_{resolution}$ is the pixel resolution.

Although pore structure information can be obtained by a brief analysis of 2D cross-sections for most porous media, some tiny pore spaces are inevitably merged into adjacent larger pores or grains, which decrease the accuracy of extracted pore and throat information. Therefore, a 3D visualization model is introduced to improve estimation accuracy.

2.2. Description of the integrated approach

The MC algorithm is the most prevalent method for reconstructing 3D visualization models of interesting objects in medicine and industry. However, its applications in geotechnical engineering are rare, where the 3D visualization model of geomaterials is constructed by geostatistics techniques (e.g., MPS). Generally, the MC algorithm is efficient for the bulk processing of large data sets, and in recognizing holes while accurately estimating volume. However, its recognition of geometry and topology remains a chief drawback, because of the typically anisotropic data between slices. In contrast, the MPS algorithm is superior for small-scale data processing, and reproduces curve features and recognizes connectivity well between slices. Unfortunately, its fatal limitations are its performance with large data and its difficulty in 3D recognization despite plentiful faces in practical applications. Here, we integrate these techniques making full use of their complementary advantages to obtain an accurate 3D model.

2.3. Recognition and reproduction of pattern library

During the recognition and construction of the pattern library, the training image is initially prepared and defined as the set of gray levels $f(x, y)$ with coordinate IDs, in which only two faces with black (0) and white (1) are included. Then, the 3D scanning template T is specified artificially, with the center node U and N_t adjacent nodes, which is written as

$$U_\alpha = U(x_0, y_0, z_0) + h_\alpha, \quad (\alpha = \alpha_x, \alpha_y, \alpha_z; \ \forall x, \forall y, \forall z) \tag{7}$$

where h_α is the 3D vector describing the geometry of the template.

Once the template is prepared, the template will collect the pattern libraries at every center node U by scanning the training image along x-, y-, and z-axes. The collection procedure of the pattern library at every center node U is written as

$$PL(U) = \{f(U); f(U + h_\alpha) | \alpha = \alpha_x, \alpha_y, \alpha_z; \ \forall x, \forall y, \forall y\} \tag{8}$$

where $f(U)$ denotes the faces at the center node U in the template and $PL(U)$ is the collected pattern library.

The pattern libraries can be considered as a collection of N_t condition data. Each pattern library can confirm the value of a local underlying spatial connectivity variable by computing the occurrence probability of the replicated face in the pattern library of training image. The probability is called conditional probability distribution function (CDF). Based on Bayesian theory, the CDF of a pattern library extracted from single training image in all planes are

$$CDF(f(U)) = \frac{prob(f(u) = k) | PL_j(U))}{prob(PL_j(U))} = C_k^j / \sum_{k=0}^{1} C_k^j, \ 1 \le j \le N_t \tag{9}$$

where C_k^j is the replicated value of the j^{th} pattern library with face k at each center node.

Next, the pattern library can be reproduced by retrieving the CDF. Finally, the image can be simulated by the pattern library.

2.3.1. Struct array representation of pattern library

The classical pattern library is stored in structured tree, which is limited by computing efficiency, memory space, and recognition of spatial information (Boucher, 2009; Mariethoz and Renard, 2010). Therefore, an array representation (Xu et al., 2012) is introduced to substitute for the tree structure. However, the retrieved image data obtained from the Xu's method is limited to the 2D plane. Although 3D spatial information is supplemented by the resampling data in the other two planes, inevitable errors occur (e.g., the precise location and distribution probability) due to the stochastic retrieval of the target image.

In this paper, the location of target point in the training image is initially determined. Then, the pattern libraries of the target points in all planes are extracted synchronously. Finally, the extracted pattern library information is stored in a struct array (Fig. 4) with 6 components including the binary and decimal pattern library, counters of replicated pattern in faces (0 or 1) and corresponding coordinate ID information, respectively.

To clearly describe the procedure for constructing the struct array of a pattern library, the training image and template in Fig. 5 are taken as an example. First, a 3D multi-orthoslice is projected onto the XOY, XOZ

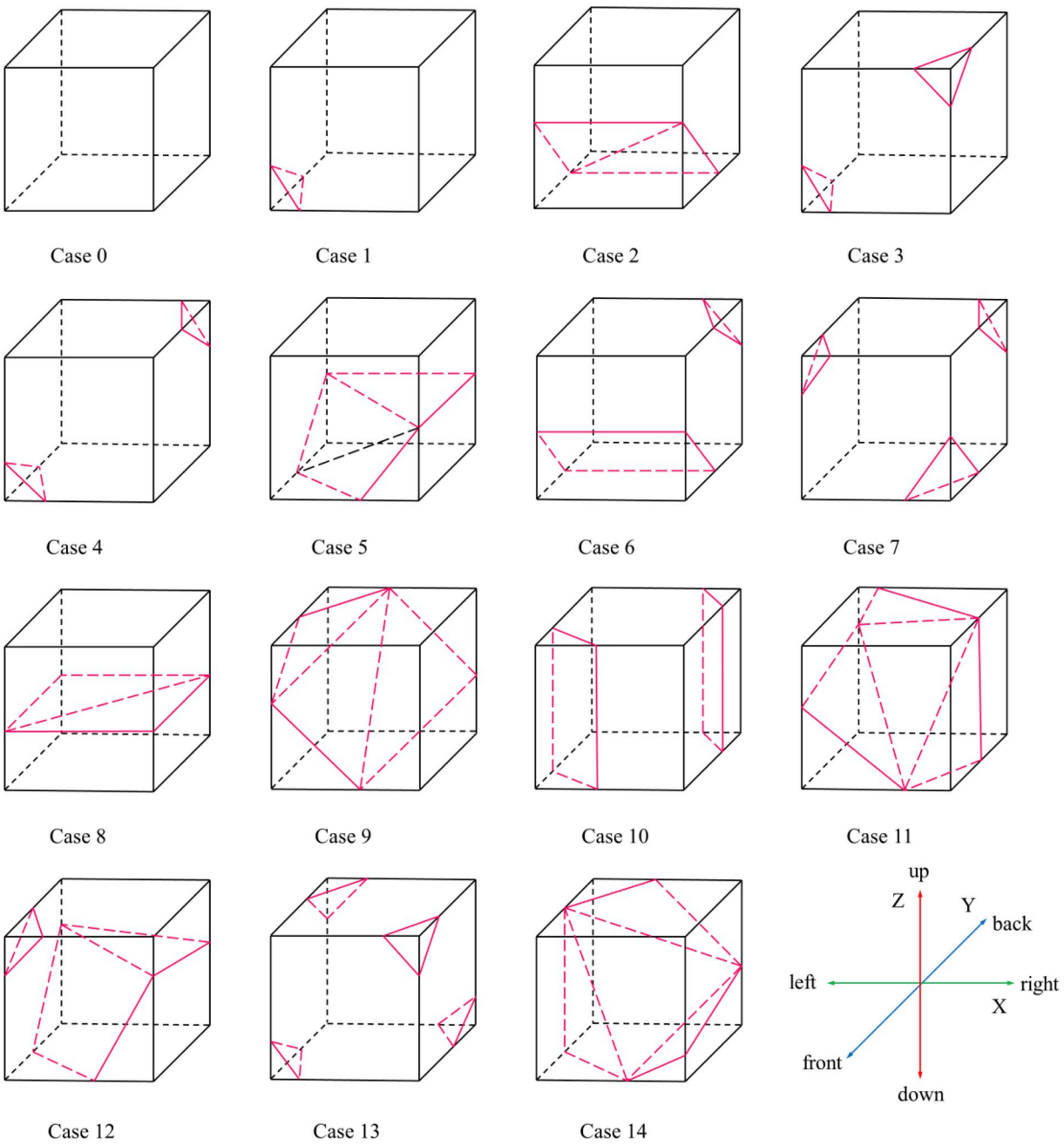


Fig. 8. The 15 basic topology cubes with isosurface facet.

and YOZ planes. Then, the scanning procedure is implemented from left to right and top to bottom, as illustrated in Fig. 5(d) and Fig.5(e). Finally, the procedure is terminated until all the pixels in the training image have been scanned (Table 1).

In the scanning procedure, the pattern library is considered as a vector consisting of binary values, as shown in the first column of Table 1. The library values are then converted to scalars, as shown in the fourth column of Table 1, which is just a simple transformation between binary and decimal numbers. By combing the binary and decimal patterns with the statistical face counters (0 or 1) and the coordination ID in the second, the third, fifth and sixth columns of Table 1, respectively, the struct array is defined as

$$A_{Struct} = \{XOY = (A_{Sub-pattern}),XOZ,YOZ\} \tag{10}$$

where A_{Struct} is the struct array consisting of the XOY, XOZ and YOZ pattern library, $A_{Sub-pattern}$ is the sub-pattern generated by scanning the training image using the large-scale template. The sub-pattern can be written as

$$A_{Sub-pattern} = [S_1 = (\ B- \text{pattern,Counters,Coordination ID,D} - \text{pattern,Counters,Coordination ID,}),\ S_2, \cdots, S_M] \tag{11}$$

where M is the index of the sub-pattern.

In practice, there exist many pattern libraries for large-scale template. Therefore, a sub-pattern is designed to deal with these cases, in which large pattern libraries are considered as dynamic struct arrays consisting of a series of small sub-patterns.

Fig. 6 demonstrates the generation of sub-patterns. A pattern is divided into three sub-patterns, whose values can be converted to scalars through the binary-decimal transformation. Points closer to the center node U have a stronger effect on the center point. Thus, initially simulating the sub-pattern near to the center point is crucial, and the target image can be rapidly simulated.

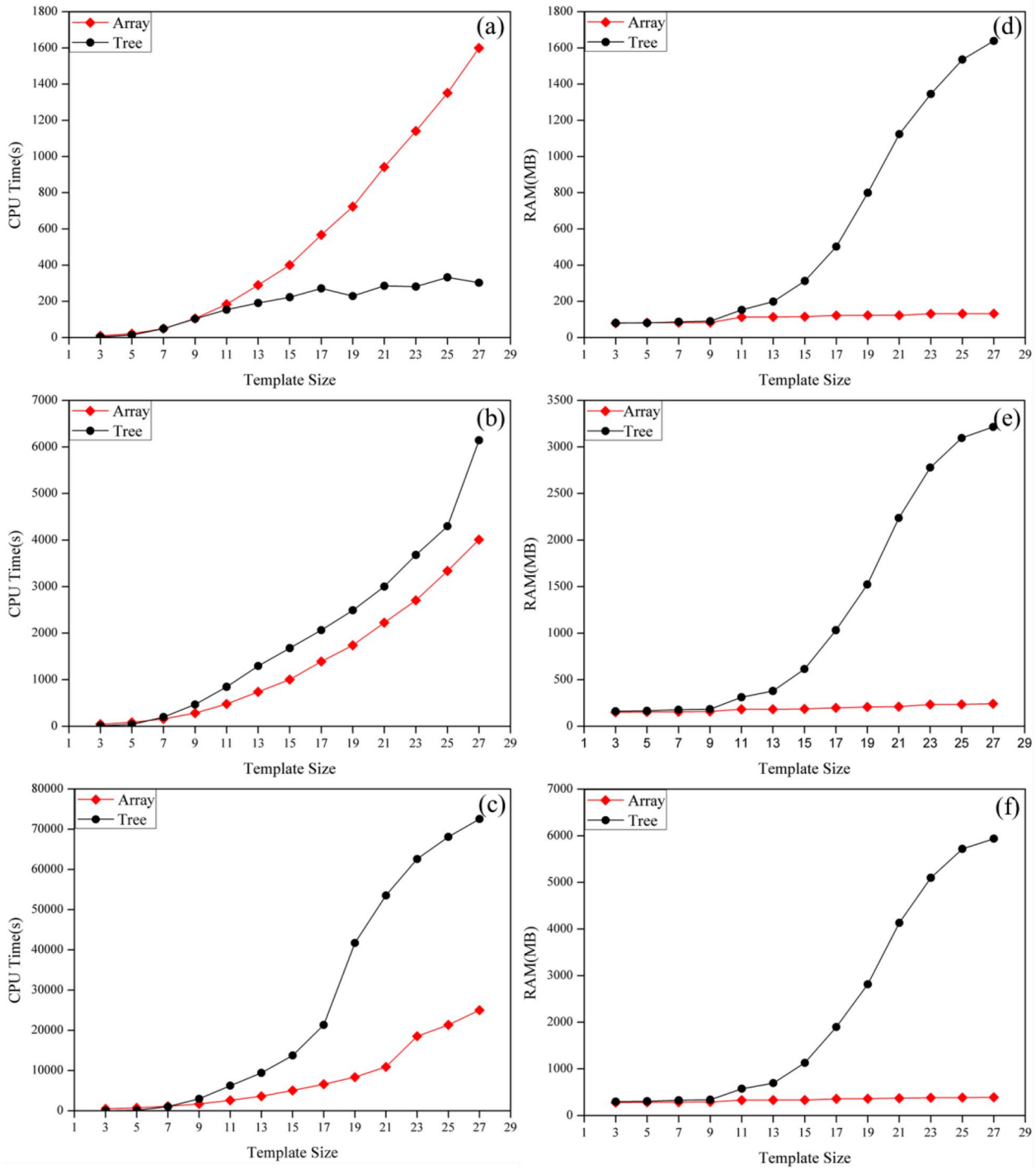


Fig. 9. CPU time and RAM consumption Comparison of different template size, (a)–(c) CPU time of simulated image at scale of 300^2, 500^2 and 1000^2, (d)–(f) the corresponding RAM usage of struct array and tree structure for the simulated image in size of 300^2, 500^2 and 1000^2.

2.4. Target image simulation

Since all the information of the training image is stored in the struct array, the single normal equation simulation algorithm (SNESIM) (Strebelle, 2000) is applied to simulate target images.

At first, a blank image is provided to simulate the target image. The grids of the blank image are scanned randomly, and each unknown pixel value is determined by the probability from the marginal distribution. Then, it encounters with neighbors that include previously-known information. Meanwhile, a pattern library with the current unknown node at its center and the previously-known encountered neighbors are constructed. By comparing the struct arrays of the pattern library extracted from the training image, the unknown node can be determined by the simple retrieve of CDF. To interpret the retrieval procedure of an unknown pixel point clearly, Fig. 5(d) and Fig. 5(e) are taken as an example. Assume the binary values at the top (1) and bottom (3) of Fig. 5(d) are 1. Then, the pattern library can be written as

$$PL(U) = (1, ?, 1, ?) \tag{12}$$

where?is the unknown neighbor nodes, and the CDF can be calculated by searching Table 1, where the counters of face 0 and 1 respectively are

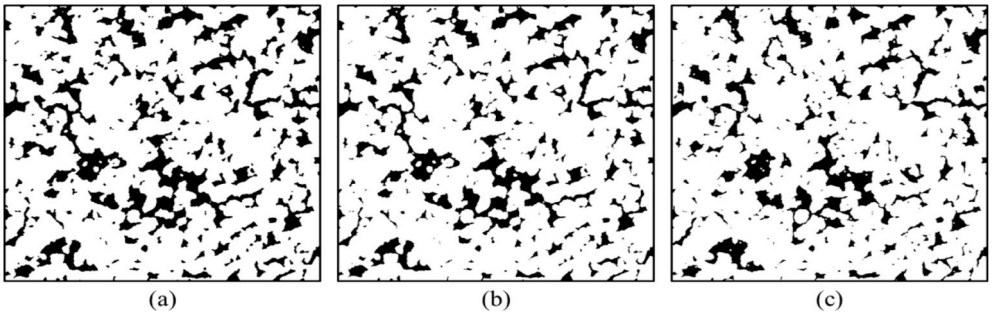


Fig. 10. The simulated image using different methods, (a) the original CT image, (b) the simulated image by struct array, (c) the simulated image by tree structure.

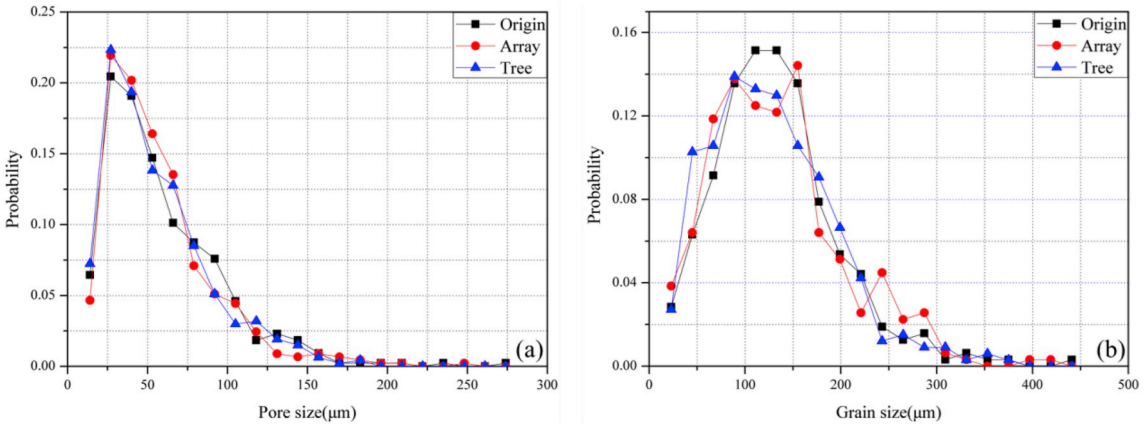


Fig. 11. The probability distribution of pore and grain sizes of a random origin CT image with corresponding simulation image by struct array and tree structure.

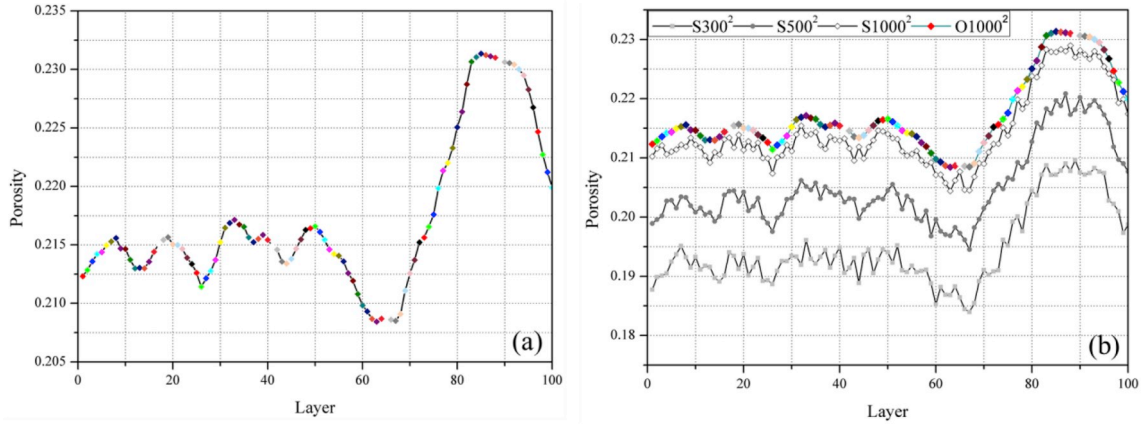


Fig. 12. Comparison of porosity distributions of the original image and the simulated images in different sizes (a) porosity distribution of the original image (b) porosity distributions of the original image and the simulated images.

$$C_0^2 = 4,\ C_1^2 = 1 \tag{13}$$

The conditional distribution probabilities in face 0 and 1 are 4/5 and 1/5, respectively. Thus, the unknown point should be 0 or 1 by retrieving the CDF (4/5 and 1/5). This procedure is ultimately looped until the simulated image is completely formed.

Generally, to produce 3D information, the whole procedures can be simply summarized as follows:

(1) The struct array including pattern library, counters of replicated faces (0 or 1), and corresponding coordinate ID is constructed by scanning a 3D ortho-slice.
(2) The pattern library with an unknown center pixel point is reproduced by scanning the simulated blank image.
(3) The value of unknown pixel point is determined by retrieving the CDF until the simulated image is completely constructed.
(4) Through looping (1)–(3), all needed 2D simulated images are reproduced, and are considered to be input parameters of the 3D reconstruction model.

2.4.1. *Visualization model of 3D reconstruction*

The 3D visualization model is reconstructed using the MC algorithm (He, 1987), in which the cube with slices S_k and S_{k+1} including eight pixel points is shown in Fig. 7(a), and including edge and vertex indexes is shown in Fig. 7(b). If the thickness between slices is represented by ID information along z-axis direction, the cube represented by MC algorithm can be written as

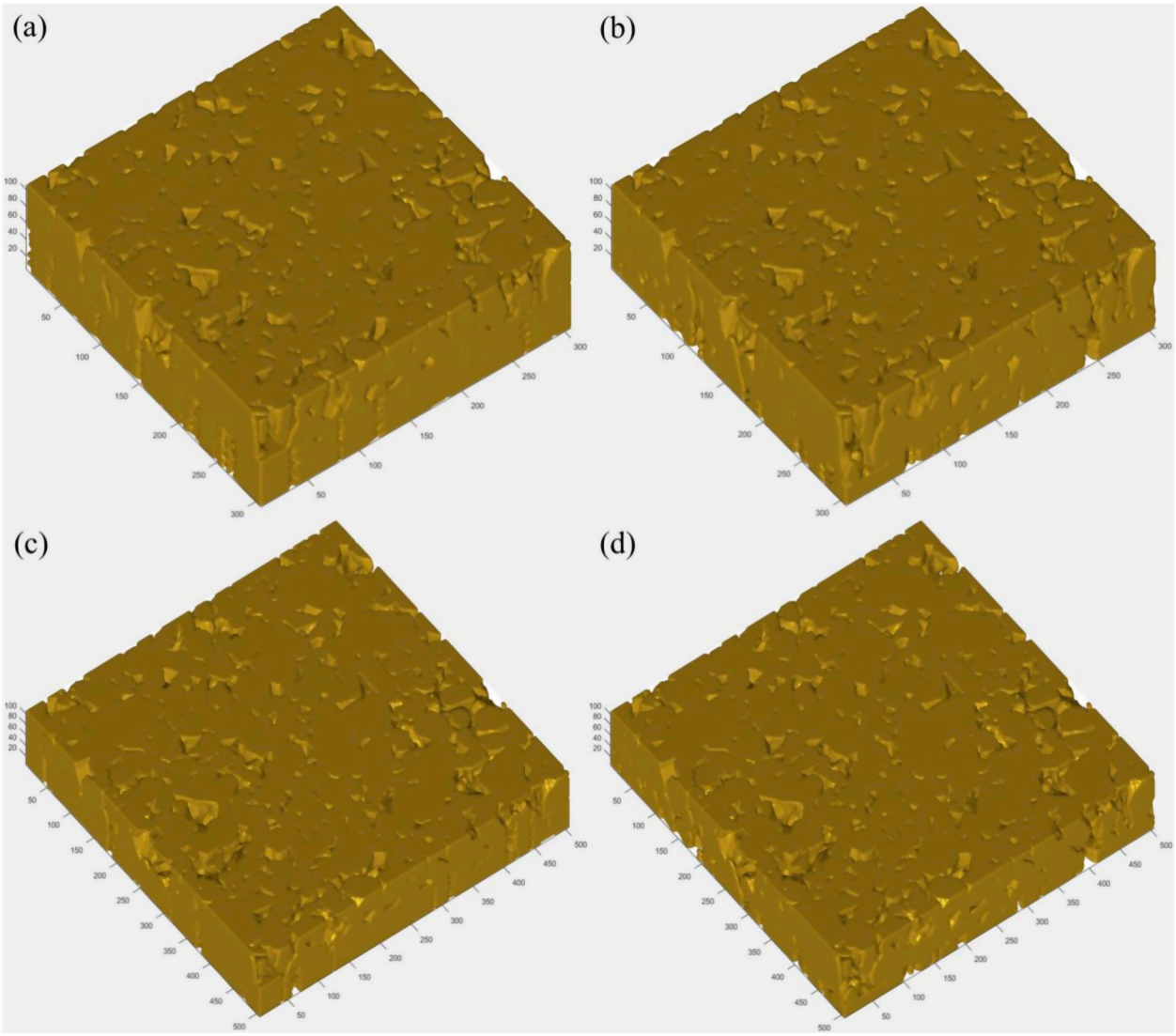


Fig. 13. 3D visualization models reconstructed by simulated images in different sizes, (a) the reconstructed half-3D models with size of $300^2 \times 100$ using 50 simulated images only containing XOY plane information and 50 simulated images containing all information, (b) the total 3D model with size of $300^2 \times 100$, (c) the reconstructed half-3D models with size of $500^2 \times 100$ using 50 simulated images only containing XOY plane information and 50 simulated images containing all information, (d) the total 3D model with the size of $500^2 \times 100$

$$\Gamma(i, j, k) = f(x_0, y_0, z_0) = \begin{cases} x + i\Delta x \le x_0 \le x + (i+1)\Delta x \\ y + j\Delta y \le y_0 \le y + (j+1)\Delta y \\ z + k\Delta z \le z_0 \le z + (k+1)\Delta z \end{cases} \tag{14}$$

where Γ is the cube, $f(x_0, y_0, z_0)$ is the gray level with spatial coordination ID.

One assumption of the MC algorithm is that the volumetric scale data are locally continuous, and each V_I has two types (negative 0 or positive 1) of voxels with eight pixel points each after comparison with the given isovalue. If the value of vertex is larger than or equal to the isovalue $\wp$, it is assigned as 1, otherwise as 0. Thus, 256 (2^8) topology cubes are possible. Considering reflection and rotation symmetry, this can be further simplified to 15 basic topology cubes (Fig. 8).

After intersecting edge E_I at each cube with$\wp$, vertex types can be assigned. In addition, using linear interpolation method, the location and normal vector of intersecting points between the isosurface and voxel edges can be respectively written as

$$P(x, y, z) = P_1(x, y, z) + \frac{\wp - f_1}{f_0 - f_1}(P_0(x, y, z) - P_1(x, y, z)) \tag{15}$$

$$N(x, y, z) = N_1(x, y, z) + \frac{\wp - f_1}{f_0 - f_1}(N_0(x, y, z) - N_1(x, y, z)) \tag{16}$$

where $P(x, y, z)$ and $N(x, y, z)$ are the coordinate and normal vector of the assigned vertex, and f_0 and f_1 are the gray levels of assigned negative 0 and positive 1 points, respectively.

Once the location and normal vector of intersecting points are determined, the 3D realization can be generated by collecting the triangular facets of the cubes.

3. Results and discussion

3.1. Comparison of 2D size effect

Fig. 9 shows the time consumption and RAM usage versus different template sizes using the struct array and tree structure representations. Both representations are adopted to simulate the images with the sizes of 300^2, 500^2 and 1000^2.

As shown in Fig. 9(a)–(c), time consumption increases with increasing scale. Initially, the times consumed are nearly the same when the struct array and tree structure representations are used. However,

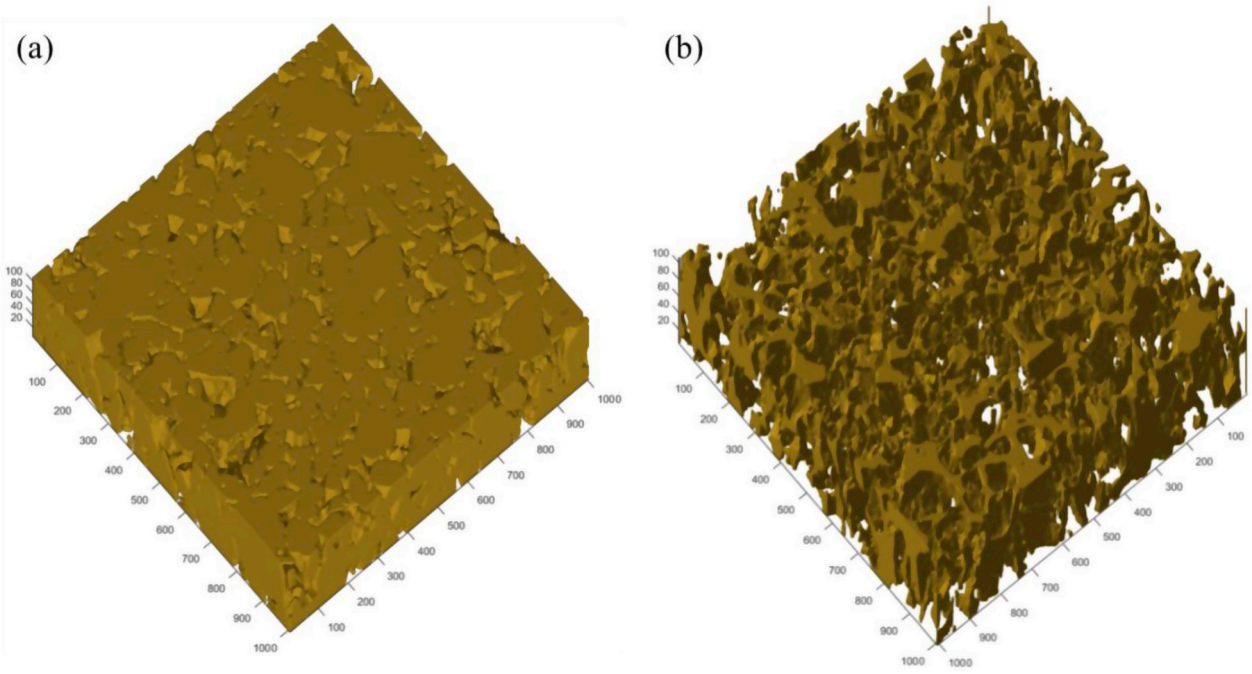


Fig. 14. 3D models with the original size, (a) 3D model reconstructed by original images, (b) 3D realistic pore structure.

Table 2
Comparison of the parameters in the present work and the previous works.

Categories	Scale 1	Scale 2	Scale 3	Previous works	
Image size	$300^2 \times 100$	$500^2 \times 100$	$1000^2 \times 100$	Øren & Bakkle (2003)	Al-Kharusi & Blunt (2007)
Porosity (%)	19.5	20.5	21.4	14.8	13.7
Find balls	1373225	4281238	21709196		
Invalid voxel	3344	3772	4470		
Number of throats	1989	2557	2622	2419	8192
Number of pores	1179	1490	1660	848	4997
Permeability (mD)	433	457	471	545	423

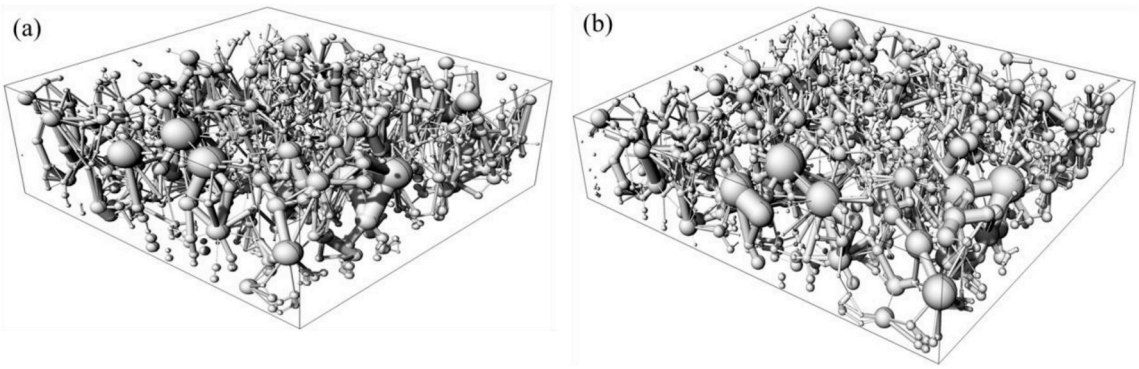


Fig. 15. The extracted network, (a) the extracted network from 3D reconstruction model with the simulated size of 300^2 (b) extracted network from 3D reconstruction model with the simulated size of 500^2.

the difference appears in template size of 9×9. Moreover, a gradual increase of the gap is shown between the two representations at scales of 300^2 and 1000^2, while the increase of difference is relatively stable at a scale of 500^2. For the scale of 300^2, the time consumed by struct array is more than that by tree structure, which is opposite of the situation with scales of 500^2 and 1000^2. Meanwhile, the increasing trend of time consumption for struct array is relatively stable in an exponential manner. However, this tendency changes at a scale of 1000^2.

Fig. 9(d)–(f) show the relationship between RAM usage and different template sizes. The struct array and tree structure representations have nearly the same RAM usage with template sizes smaller than 11×11, while the difference between these two representations increases with larger templates. Moreover, the RAM usage for tree structure increases dramatically with increasing template size, while it keeps nearly constant for struct array, which benefits large-scale simulation. Thus, the struct array can be a better substitute for tree structure, especially for large-scale training images scanned by large template.

Fig. 10 shows the quality of the original image, and the simulated images using the struct array and trees. The image quality simulated by

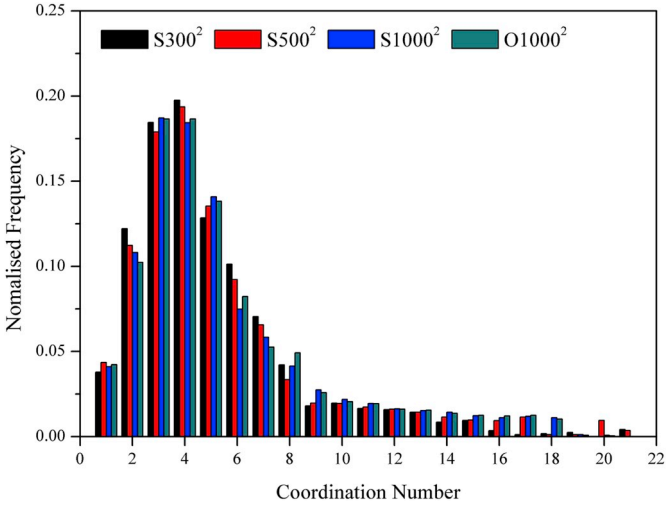


Fig. 16. Coordination numbers for the original image, and the simulated image with the size of 300^2, 500^2 and 1000^2.

struct array is better than that simulated by the tree, and closer to the original image.

Fig. 11 shows the pore and grain probability distributions of the original image and the images simulated by the struct array and tree structure. The pore and grain probability distribution curves of the simulated image agree well with those of the original image. Moreover, the grain distribution curve obtained when using the struct array performs better than that obtained when using tree structure.

Fig. 12 shows the porosity distributions of the original image, and the simulated images using the struct array with sizes of 300^2, 500^2, and 1000^2. Fig. 12(a) shows the porosity distribution of the original image. Fig. 12(b) shows the porosity distribution of the simulated image at different scales. As the simulation size increases, the porosity distributions of the simulated images agree well with that of the original image. However, the porosity of the simulated image is less than that of the original image, which may have been caused by the stochasticity of the struct array-based MPS method for the retrieving of the CDF. Moreover, when the simulated size is the same as the original, the porosity of the simulated images is nearly the same as that of the original images.

3.2. Comparison of 3D reconstruction model

The 3D visualization model is reconstructed as the images are simulated, which is followed by the extraction of pore structure. Fig. 13 shows the 3D models at the scales of $300 \times 300 \times 100$, and $500 \times 500 \times 100$. Fig. 13(a) and (c) show the half-3D models using 50 simulated images only containing XOY plane information and 50 simulated images containing all information. Fig. 13(b) and (d) show the total 3D models at these two scales. The 3D model in the original scale and its 3D realistic pore structure are plotted in Fig. 14, allowing comparison of the pore structure extracted from simulated images by MC algorithm.

From Figs. 13 and 14, we find Fig. 13(a) and (c) are pseudo-3D models, losing the true 3D spatial information, because the pattern library is only extracted in XOY plane similar to the classical tree structure. Fortunately, the proposed struct array is able to retrieve complete 3D information, and so is a promising technique for simulating the 3D models at engineering scales.

3.3. Comparison of permeability and coordination number

Table 2 shows the parameters extracted from the 3D reconstruction models in this and previous studies. The average coordination number is 4.76 with less than 0.2% invalid voxels excluded. The images are investigated for different simulation sizes, and are larger than that in the previous work, proving the superiority and flexibility of the struct array for the bulk processing of larger data sets.

Based on the 3D models at scales of 300^2, 500^2 and 1000^2, the permeability value is calculated using the Lattice Boltzmann method (Jeong, 2010) as

$$\begin{cases} K = D^2 \exp\left\{C_1 \ln\left(\frac{\varepsilon^{11/3}}{(1-\varepsilon)^2}\right) - C_2\right\} \\ C_1 = 0.709 - 1.62Kn_o + 5.982Kn_o^2 \\ C_2 = 5.09 - 14.14Kn_o + 36.84Kn_o^2 \end{cases} \quad (17)$$

where K is the permeability, D is the equivalent diameter of gain and Kn_o is the Knudsen number of overlapping porous structure, ε is the porosity, C_1 and C_2 are the correlation coefficient determined by Knudsen number.

The values obtained from each scale agree with those from previous studies (Table 2) but with a little difference, which may be caused by the CDF retrieval from the pattern library.

Fig. 15 represents the pore structure extracted from the reconstructed 3D model with sizes of 300^2 and 500^2. By comparing these models, we find that the extracted pore structure, in which the pores and throats are simplified as spheres and cylinders with its realistic sizes, agree well with the realistic pore structure Fig. 15(b). As the size increases, the 3D model shows more details such as the boundary pore. Moreover, the computing efficiency of reconstructing the 3D model improves based on this integrated method.

Fig. 16 shows the coordination numbers of the original image, and the simulated images with size of 300^2, 500^2, and 1000^2, which are obtained from the 3D models reconstructed by the proposed integration method. The coordination numbers are similar to those extracted from the original 3D model, demonstrating the reliability and applicability of the integration method of the MPS and MC algorithms.

4. Summary and conclusions

In this paper, 2D image analysis investigates pore and grain distribution. A struct array representation of a pattern library is then proposed to simulate target images. The 3D visualization models are reconstructed by the integrated method, and the permeability and coordination number are calculated from the pore structure extracted. The main conclusions can be drawn as follows:

(1) The time consumption and RAM usage when using a struct array are lower than those when using a tree structure.
(2) The image quality simulated by struct array is better than that simulated by tree structure. When the size of the simulated image is closer to the original size, the porosity distributions of the simulated image agree better with that of the original image.
(3) The pore and grain size distributions extracted from images simulated by struct array and tree structures are coincident with those of the original image. Moreover, the struct array performs better than the tree structure.
(4) The coordination number and permeability agree well with those in the previous studies, showing the reliability and applicability of the proposed integration method.

Acknowledgement

The implementation codes for this paper are available by contacting the corresponding author at cqxpzhou@hotmail.com.This work was supported by the National Natural Science Foundation of China (Nos. 51839009 and 51679017), project 973 (Grant no. 2014CB046903), Graduate Scientific Research and Innovation foundation of Chongqing,

China (Grant No. CYB16012), Natural Science Foundation Project of CQ CSTC (Grant Nos. cstc2017jcyj-yszxX0013 and cstc2016jcyjys0005).

References

Al-Kharusi, A.S., Blunt, M.J., 2007. Network extraction from sandstone and carbonate pore space images. J. Pet. Sci. Eng. 56, 219–231.
Al-Raoush, R.I., Willson, C.S., 2005. Extraction of physically realistic pore network properties from three-dimensional synchrotron x-ray microtomography images of unconsolidated porous media systems. J. Hydrol. 300, 44–64.
Arns, J.Y., Robins, V., Sheppard, A.P., Sok, R.M., Pinczewski, W.V., Knackstedt, M.A., 2004. Effect of network topology on relative permeability. Transport Porous Media 55, 21–46.
Arand, F., Hesser, J., 2017. Accurate and efficient maximal ball algorithm for pore network extraction. Comput. Geosci. 101, 28–37.
Bashtani, F., Maini, B., Kantzas, A., 2016. Single-phase and two-phase flow properties of mesaverde tight sandstone formation; random-network modeling approach. Adv. Water Resour. 94, 174–184.
Boucher, A., 2009. Considering complex training images with search tree partitioning. Comput. Geosci. 35, 1151–1158.
Byholm, T., Toivakka, M., Westerholm, J., 2006. The application of morphological algorithms on 3-dimensional porous structures for identifying pores and gathering statistical data. In: Wseas International Conference on Simulation, Modelling and Optimization. World Scientific and Engineering Academy and Society (WSEAS), pp. 264–270.
Chan, T.F., Vese, L.A., 2001. Active Contours without Edges, vol. 10. IEEE Press, pp. 266–277.
He, C., 1987. Marching cubes: a high resolution 3d surface construction algorithm. acm siggraph comput graph. Comput. Graph. 21, 163–169.
Homberg, U., Baum, D., Prohaska, S., Kalbe, U., Witt, K.J., 2012. Automatic extraction and analysis of realistic pore structures from μCT data for pore space characterization of graded soil. Eur. Urol. 60, 1074–1080.
Ioannidis, M.A., Chatzis, I.I., 2000. On the geometry and topology of 3d stochastic porous media. J. Colloid Interface Sci. 229, 323–334.
Jeong, N., 2010. Advanced study about the permeability for micro-porous structures using the lattice Boltzmann method. Transport Porous Med. 83, 271–288.
Jonassen, D., Spector, M.J., Driscoll, M., Merrill, M.D., van Merrienboer, J., Driscoll, M.P., 2008. Handbook of Research on Educational Communications and Technology: a Project of the Association for Educational Communications and Technology. Routledge.
Knackstedt, M.A., Arns, C.H., Limaye, A., Sakellariou, A., Senden, T.J., Sheppard, A.P., et al., 2004. Digital core laboratory: properties of reservoir core derived from 3d images. In: Spwla Annual Logging Symposium, vol. 56. pp. 66–68.
Lindquist, W.B., Lee, S.M., Coker, D.A., Jones, K.W., Spanne, P., 1996. Medial axis analysis of void structure in three-dimensional tomographic images of porous media. J. Geophys. Res-Sol. EA. 101, 8297–8310.
Mariethoz, G., Renard, P., 2010. Reconstruction of incomplete data sets or images using direct sampling. Math. Geosci. 42, 245–268.
Ohnishi, K., Suzuki, Y., Watanabe, Y., 2014. Quantitative 3d characterisation of the pore space of real rocks: improved μ-ct resolution and pore extraction methodology. Animal 8, 1339–1348.
Okabe, H., Blunt, M.J., 2005. Pore space reconstruction using multiple-point statistics. J. Pet. Sci. Eng. 46, 121–137.
Olafuyi, O.A., Sheppard, A.P., Arns, C.H., Sok, R.M., Cinar, Y., Knackstedt, M.A., et al., 2006. Experimental Investigation of Drainage Capillary Pressure Computed from Digitized Tomographic Images, vol. 1. Trans. Tech. Publications, pp. 1–10.
Øren, P.E., Bakke, S., 2003. Reconstruction of berea sandstone and pore-scale modelling of wettability effects. J. Pet. Sci. Eng. 39, 177–199.
Rabbani, A., Jamshidi, S., Salehi, S., 2014. An automated simple algorithm for realistic pore network extraction from micro-tomography images. J. Pet. Sci. Eng. 123, 164–171.
Sharqawy, M.H., 2016. Construction of pore network models for berea and fontainebleau sandstones using non-linear programing and optimization techniques. Adv. Water Resour. 98, 198–210.
Silin, D., Patzek, T., 2006. Pore space morphology analysis using maximal inscribed spheres. Physica A 371, 336–360.
Straubhaar, J., Walgenwitz, A., Renard, P., 2013. Parallel multiple-point statistics algorithm based on list and tree structures. Math. Geosci. 45 (2), 131–147.
Strebelle, S., 2000. Sequential Simulation Drawing Structures from Training Images. Ph. D. thesis. Stanford University, Stanford, pp. 1–116.
Sweeney, S.M., Martin, C.L., 2003. Pore size distributions calculated from 3-d images of dem-simulated powder compacts. Acta Mater. 51, 3635–3649.
Wildenschild, D., Sheppard, A.P., 2013. X-ray imaging and analysis techniques for quantifying pore-scale structure and processes in subsurface porous medium systems. Adv. Water Resour. 51, 217–246.
Xu, Z., Teng, Q., He, X., Yang, X., Li, Z., 2012. Multiple-point statistics method based on array structure for 3d reconstruction of fontainebleau sandstone. J. Pet. Sci. Eng. 100, 71–80.
Zhang, T., Du, Y., Huang, T., Li, X., 2015. Gpu-accelerated 3d reconstruction of porous media using multiple-point statistics. Computat. Geosci. 19 (1), 79–98.