IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 11 October 2022, accepted 4 November 2022, date of publication 10 November 2022, date of current version 16 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3221458

RESEARCH ARTICLE

An Improved 3D Box-Counting Dimension Computing Technology for Estimating the Complexity of 3D Models

CHONG WANG[1] AND WEIQIANG AN[2]
[1]Network Security and Information Office, Shandong University of Science and Technology, Qingdao 266590, China
[2]College of Civil Engineering and Architecture, Shandong University of Science and Technology, Qingdao 266590, China

Corresponding author: Weiqiang An (skd991823@sdust.edu.cn)

This work was supported in part by the Open Research Fund Program of Beijing Key Laboratory on Integration and Analysis of Large-Scale Stream Data under Grant KF2018004, and in part by the Virtual Simulation Experiment Teaching Project of Shandong University of Science and Technology under Grant XNFZ2019M26.

ABSTRACT The box-counting dimension, which can effectively reflect the complexity and self-similarity of models, is an important method for calculating the fractal dimension of models. To improve the precision of the box-counting dimension algorithm, in this study, we propose an improved 3D box-counting dimension algorithm for models composed of triangular meshes. Through linear coding and optimization of intersection detection between triangles and octree cells, we can accurately calculate the box dimension of models with less memory expense. Through the results of measuring regular geometric bodies and spatial Voronoi bodies, it can be seen that the method has good performance, and the ability of filling space of fractal bodies can be calculated accurately. In the circumstance of 3D box-counting dimension algorithm cannot work well for comparing curvature changes of non-fractal surfaces, the weighted box-counting dimension algorithm can be used to quantitatively analyze the complexity of surface curvature.

INDEX TERMS Box dimension, triangular mesh, octree, self-similarity, fractal dimension.

I. INTRODUCTION

Many objects in nature present structural complexity and self-similarity, which are difficult to be described using Euclidean geometry. Thus, Mandelbrot proposed fractal theory to quantitatively analyze the characteristics of objects [1], [2]. Inspired by fractal theory, many scholars have attempted to apply it to computer image processing. Then the box-counting dimension method has developed rapidly [3]. The differential box counting was proposed to solve the problem of texture segmentation [4]. The generalized box-counting worked for any arbitrarily sized (both squared and rectangular) images and gave a higher rate of accuracy in terms of less fitting error in detecting exact surface roughness [5]. The isarithm method was employed to estimate the box dimensions of grey images [6]. Several fractal dimension (FD) estimators based on polyhedral generation algorithms had been compared in literature [7], [8]. In recent years, FD has been widely used in fields of geology and medicine, such as quantifying the shape of fluid inclusions [9], analyzing the organization patterns of complex river networks and so on [10], [11], [12], and [13]. In architecture, FD was applied to classification the building image, and a survey of the connection between a building's fractal dimension trend and observation distance [14], [15]. For estimation of three-dimensional FD, traditional research methods mostly use vertices of surface to roughly calculate the number of intersecting boxes [16]. However, these methods may be problematic because only three boxes at the vertices of each triangle are counted. Some boxes that are within a triangle or cross triangles by edges may be lost [17].

In this study, an improved precision measurement means for 3D box-counting dimension (3D-BCD) of 3D models composed of triangular meshes is proposed. In additional, a 3D weighted box-counting dimension (3D-WBD) algorithm is introduced to quantitatively analyze the complexity of

The associate editor coordinating the review of this manuscript and approving it for publication was Songwen Pei.

surfaces, while similar results obtained by the 3D-BCD method. Furthermore, the algorithm was used to estimate the FD value of architectural models, which more clearly reflected the complex characteristics of the models.

The remainder of the paper is organized as follows. Section II provides a review of FD and box-counting dimension. Section III addresses the implementation steps of the improved 3D-BCD method and the 3D-WBD method, including creating octree and coding for cubes, intersection judgment and formula calculation. Section IV provides three experiments about measuring the box dimension of regular 3D bodies, Voronoi polyhedrons and real building bodies respectively. we can compute the value of complexity of surface curvature by the 3D-WBD method. Section V addresses the problems encountered in the experiments and discusses them in detail. Finally, Section VI summarizes our conclusion.

II. FRACTAL DIMENSION AND BOX-COUNTING DIMENSION

In fractal geometry, Hausdorff–Besicovitch dimension is a main method for calculating FD, which is used to represent the ability of filling space. However, in many cases it is difficult to obtain the value of Hausdorff–Besicovitch dimension, and it may not even exist. In certain ways the box-counting dimension can be employed to gain the approximate value of the Hausdorff–Besicovitch dimension. Through a large number of experiments, it can be found that within a certain scale, box-counting dimensions can be used to approximatively analyze the planar or spatial complexity of objects. We call it fractal parameter [18].

Surfaces of a model to be measured is defined as a set T, and T is covered by adjacent cubic cells of side lengths δ. The expansion of T is represented as the number of intersections between T and all cells. The box-counting dimension reflects how the irregularity of the set expands rapidly during $\delta \to 0$ [19]. The formula used is as follows:

$$D(\delta) = \lim_{\delta \to 0} \frac{\lg N(\delta)}{\lg \delta} \tag{1}$$

where $N(\delta)$ is the number of intersections between the cubic cells and the surface of the model. δ is the side length of each cell.

Let us consider L as the edge length of the model and m as the maximum number of subdivision levels. Each parent cell is divided into several equal parts in the i-th subdivision. The length of each child node is $\delta_i = \delta_{i-1}/2 (i = 1, 2, \ldots, m)$, specially $\delta_0 = L$.

Thus, the i-th function D_i is defined as follows:

$$D_i = \frac{\lg(N_i/N_{i-1})}{\lg 2} \tag{2}$$

where N_i is short for $N(\delta_i)$.

The function used to determine box-counting dimension was herein extended from 2D to 3D. A square in plane was subdivided into four small ones each time in 2D, while a cube in space was subdivided into eight small ones in 3D. To compute the number of cubes intersected in each layer, octree was employed [20].

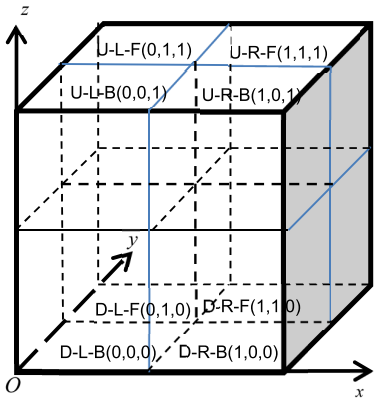


FIGURE 1. Coding rules of the subdivision of spatial octree.

III. ALGORITHM OF AN IMPROVED 3D BOX-COUNTING DIMENSION

On the basis of the previous work [18], we optimized the relevant algorithms including linear coding and high-precision intersection detection to improve the efficiency and accuracy of 3D-BCD. Furthermore, the 3D-WBD algorithm was proposed to quantify the complexity of surface curvature.

A. CREATING OCTREE AND CODING FOR CUBES

The model is divided into a set of triangular patches $T(n)$, where n is the number of triangles. The minimum x, y, z components of all triangles in the set are taken as the coordinate origin, and the minimum axial bounding box surrounding the model is taken as the root node to establish an octree. The linear codes of the eight nodes in the first subdivision level are defined as $Q_1 = Z + Y \times 2 + X \times 2^2$, where $X, Y, Z = \{0, 1\}$, which represents the number of one node in each direction of the coordinate axis, as shown in Fig. 1. If q_i is the code of one cell in the process of iterative decomposition of the octree, the set of child nodes of q_i is coded as $Q_{i-children} = q_i \times 2^3 + Q_1$.

The bounding box of the model is initialized as the root node, and the algorithm for calculating the number of intersecting cells is presented in Algorithm 1.

In Algorithm 1, c_i is i-th cube object with six vertices and eight links to its child nodes. q_i is the code of c_i. TS is a set of all triangles in model mesh, and m is the maximum number of times the root node is subdivided. intersectSet stores all codes of intersecting cells. The intersection detection algorithm is described in the next section.

B. INTERSECTION DETECTION OF A TRIANGLE AND A CUBIC CELL OF OCTREE

The intersection of one spatial triangle (T) and one cubic cell (C) of octree can be divided into four categories:

Algorithm 1 Subdivide Node and Code for Child Nodes

Input: c_i, q_i, $TS = (T_1, T_2, \ldots, T_i, \ldots, T_n)$ and m
Output: intersectSet
foreach T_j *in TS:*
  if T_j intersects with c_i **then**
    intersectSet.push(q_i);
    if subdivided times of $c_i < m$ **then**
      for $k = 1; k \leq 8; k++$ **do**
        coding child node $q_{i-k} = q_i * 8 + k\text{-}1$;
        generate child-node object c_{i-k};
        c_i.children.push(c_{i-k});
        Algorithm1(c_{i-k}, q_{i-k}, TS, m);
    break;
return intersectSet

Algorithm 2 Judge Intersection of Triangle T and Cubic Cell C

Input: $V(T) = (v_1, v_2, v_3)$, C
Output: isIntersect
initial isIntersect = false;
if C contains v_1 or C contains v_2 or C contains v_3 **then**
  isIntersect = true;
else
  create T's Bounding box $boxT$;
  if $boxT$ intersects C **then**
    create six faces of C $faceSet(C) = \{f_1, f_2, \ldots, f_6\}$;
    for $k = 1; k \leq 6; k++$ **do**
      if f_k coplanar with T **then**
        compute normal vector $\overrightarrow{n_{f_k}}$ and $\overrightarrow{n_T}$;
        if $\overrightarrow{n_{f_k}} = \overrightarrow{n_T}$ **then**
          isIntersect = true;
          break;
      else if f_k intersects T **then**
        isIntersect = true;
        break;
return isIntersect

C containing T, C intersecting T **with** T's vertex(es) in C, C intersecting T **without** T's vertex(es) in C, and coplanar (Fig. 2). Whether triangle vertices are present in a cell can be regarded as a sufficient but unnecessary condition for intersection. Whether the triangle intersects with the six faces of the cubic cell must be detected individually to accurately determine if they intersect or not, as shown in Fig. 2(c). The process to judge one triangle intersecting another triangle or not is based on the Tomas Möller algorithm [21]. Thus, the whole implementation of intersection detection is presented in Algorithm 2.

When the axial bounding box of the triangle intersects the cubic cell, and the distances between the vertices of the triangle and the plane in which one face of the cubic cell is located is equal to zero, the triangle is coplanar with the plane. If the normal direction of the plane is the same as that of the triangle, it is judged to be intersecting. Otherwise, if the two normal directions were opposite, it is judged as disjoint to avoid repeated counting.

C. CALCULATION OF WEIGHTED BOX DIMENSION

For each cubic cell, if it intersects with the model, the cell's code is recorded, and the number of intersecting cells at the level is increased by 1. Simultaneously, the cell is decomposed to eight child cubes. Cells that do not intersect all triangles are no longer subdivided, so that unnecessary computing and memory expense can be saved. Through this iterative process, we can calculate the actual number of intersection cells at each level, and then substitute it into formula (2) to obtain the value of 3D-BCD.

The box-counting dimension of 3D model is not the same as that of the nature. Affected by the modeling precision, its result tends to 2 mostly. It is sometimes difficult to quantitatively compare the complexity of building models only by the 3D-BCD method. Therefore, we propose a method that is more sensitive to curvature changes.

Let N_i be the number of intersections between the model and the cells of the i-th level octree. $\boldsymbol{P_i} = [p_1^i, p_2^i, \cdots, p_8^i]$, where $p_k^i (k = 1 \ldots 8)$ represents the number vector of cells with k child cell(s) intersecting with the model at the i-th level. Therefore, we can obtain $N_i = \sum_{k=1}^{8} p_k^i$. The weights vector $\boldsymbol{W} = [w_1, w_2, \cdots, w_8]$, where $w_k \in [0, 1]$, is set. We do dot product between $\boldsymbol{W}$ and $\boldsymbol{P_i}$ to control the weight of each component of $\boldsymbol{P_i}$. Further, the weighted box dimension at i-th level is defined in formula (3).

$$D_w^i = \log \left(\frac{\sum_{k=1}^{8} w_k k p_k^i}{\sum_{k=1}^{8} w_k p_k^i} \right) / \log 2 \tag{3}$$

That can be abbreviated as formula (4).

$$D_w^i = \log \left(\frac{kW \cdot P^i}{W \cdot P^i} \right) / \log 2 \tag{4}$$

IV. EXPERIMENTAL ANALYSIS

Three experiments were carried out in order to make sense of the reliability and accuracy of the method. First, we measure a cube, a sphere and a Menger sponge model to compare the FD results of the proposed method and others. Then we employed the method to Voronoi models and a real engineering project in order to quantitatively analyze the curvature changes of surfaces of seven building structures.

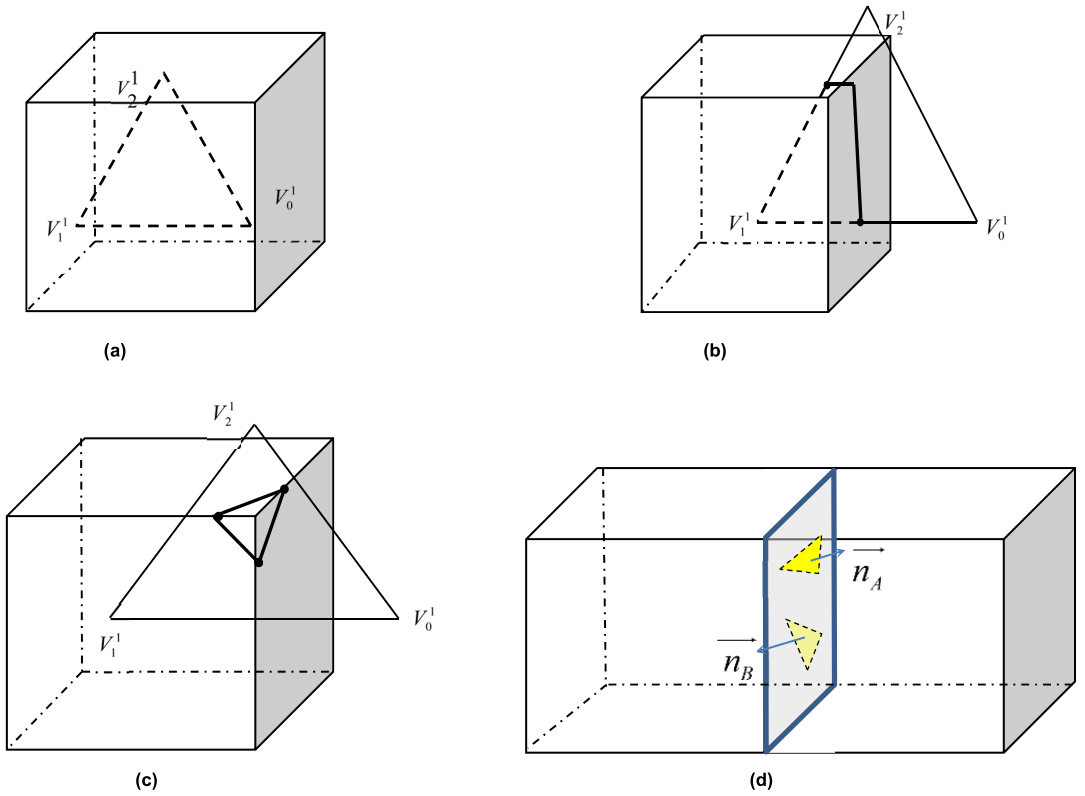


FIGURE 2. Types of intersection of one spatial triangle (T) and one cubic cell (C). (a) C containing T. (b) C intersecting T with T's vertex(es) in C. (c) C intersecting T without T's vertex(es) in C. (d) coplanar.

TABLE 1. The FD results obtained using the theoretical FD, typical traditional methods, and the proposed 3D-BCD method.

Phantom Models	3D Cubic Surface	3D Spherical Surface	Menger Sponge
Theoretical FD [1, 19]	2	2	2.7268
Traditional BC FD [16, 17]	1.891	2.003	2.721
3D-TBC FD [17, 22]	2.000	2.001	2.671
The previous 3D-BCD [18]	2.0019	2.0002	2.7659
The Improved 3D-BCD	2.0000	2.0002	2.7258

TABLE 2. Results of N_i and D_w^i of voronoi models.

Octree level	N_i				
	Blocks=30	Blocks=60	Blocks=90	Blocks=120	Blocks=150
1	8	8	8	8	8
2	32	32	32	32	32
3	96	96	96	96	96
4	607	634	636	640	640
5	2803	3138	3306	3480	3555
6	11930	13976	15181	16293	17164
7	51010	62663	69601	75528	80663
D_w	**2.0962**	**2.1647**	**2.1968**	**2.2128**	**2.2325**

$W = [1,1,1,1,1,1,1,1]$

A. TEST FOR REGULAR MODELS

To test the box-counting dimensions of a regular model by the 3D-BCD method, an axial cubic bounding box was established. The bounding box was considered as the root node of octree. The nodes were divided iteratively and the intersection cells at each level were recorded (Fig. 3). Table 1 provides the FD results obtained using the theoretical FD, typical traditional methods, and the proposed 3D-BCD method. The theoretical FD of objects can be computed by formula (1). Details of the theoretical values can be found in [1] and [19]. This example shows that the method has high accuracy and good performance.

B. TEST OF SPATIAL VORONOI MODELS

A series of spatial Voronoi models, with the same volume (length: width: height = 160: 100: 24), were created to calculate the 3D box-counting dimension of fractal structure models. five Voronoi models are composed of 30, 60, 90, 120 and 150 blocks individually. The specific divisions are

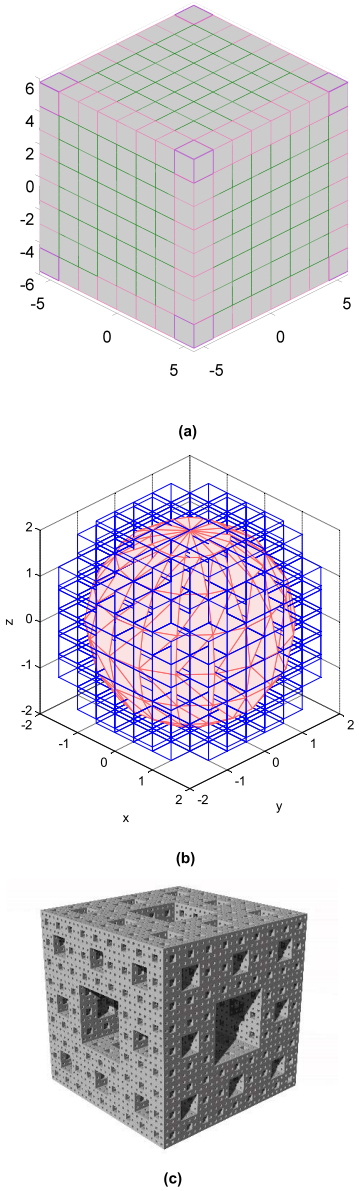


FIGURE 3. Phantom three dimension models. (a) cube occupied by 3rd-level cells. (b) sphere occupied by 3rd-level cells. (c) Menger sponge.

shown in Fig. 4 (Designed and generated by grasshopper in Rhino 7.0). After the operation of triangulation, the models were imported into the program one by one for 3D-WBD calculation. The numbers of intersecting nodes were integrated into the formula (3). The obtained values D_w are shown in Table 2. The FD values of five models increased accordingly as the number of blocks increases.

C. PROJECT APPLICATION

The Yu Qingcheng Art Museum [22], which is located in Tianjin, China, was designed by Professor Zhang Hua, who is a member of our project group. He was awarded the German Design Award in 2018 for the design of the museum, which comprises seven interrelated sculpture-like blocks that seems to be in a perpetual state of motion. The gradual change in the design of the seven individual blocks of the Yu Qingcheng Art Museum is based on homotopy equivalence transformation from blocks 1 to 7 (Fig. 5). The difference in seven FD values of calculated box-counting dimensions were not obvious and they all trend to 2 because of these buildings are not fractal. The 3D-BCD method cannot be used to calculate the curvature change of these blocks. To compare the rate of curvature change, more-children nodes were reserved and less-children nodes were ignored by the 3D-WBD method.

Let $\boldsymbol{W} = [0,0,1,1,1,1,1,1]$. The nodes which only had one and two child nodes intersecting the model during subdivision were filtered. Substitute $\boldsymbol{W}$ into the formula (4), and we can obtain the results. The changes in the spatial complexity of the blocks from left to right can be seen more clearly, which is consistent with the trend of the proportion of non-zero Gaussian curvature (Fig. 6-7).

V. DISCUSSION

More experiments about measuring surface complexity of architecture by the 3D-BCD method have been described in previous work [18]. For a complex 3D model, due to the large number of triangles and necessary of intersection detection, the algorithm requires much running time and computing memory. Therefore, we try to improve the previous algorithm by linear coding and detection optimization. The required memory for each subdivision level would increased nearly seven times than that of parent level because of the characteristic of the octree. Linear coding, instead of an octree-linked list, can make use of less memory. The octree-node objects in the program are generated from the code of nodes only when we need to perform intersection detection. After that

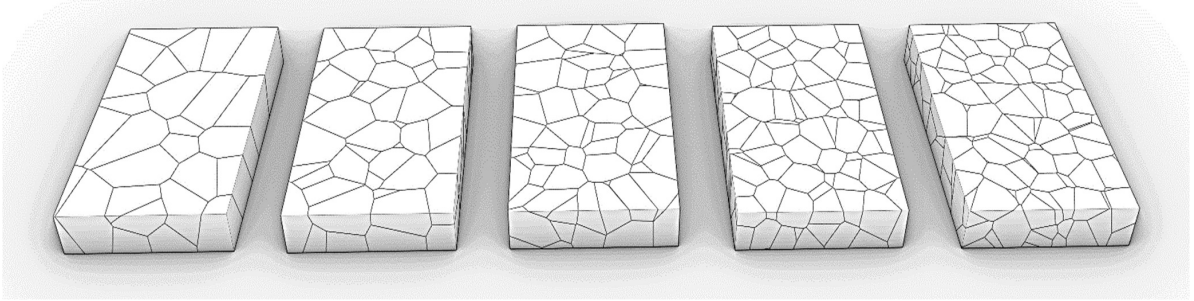


FIGURE 4. Five Voronoi models with different blocks.

FIGURE 5. Yu Qingcheng Art Museum.

they would be destroyed immediately. Furthermore, nodes which don't intersect with models will not be subdivided.

Artificial models are different from natural objects because they have no infinite fractal structures when they are subdivided to a small size. Therefore, the fractal-dimension measurement of a 3D model is affected by the actual modeling precision. The box-counting dimension of a model without self-similarity usually converges quickly to 2. Within a certain subdivision range, the model with self-similarity shows a stable box-counting dimension interval, which is called the scale-free interval [23]. The box-counting dimension of the 3D model invariably reflects the complexity or space-filling ability of the model.

It should be noted that 3D-WBD is not Hausdorff–Besicovitch dimension. The calculated value of 3D-WBD is related to parameter $\boldsymbol{W}$. According to formula (4), the value of 3D-WBD is between 1 and 3. From Experiment C, it can be seen that, unlike 3D-BCD, when less-child nodes are ignored, the calculation results show a linear correlation with the change of surface curvature. This provides a new reference for the design of building models. In fact, we measure a cube and a sphere respectively by 3D-WBD (where $\boldsymbol{W} = [0,1,1,1,1,1,1,1]$). The final result of cube is 2.00, while the one of sphere is 2.16. Compared with the Gaussian curvature, which can only be used to calculate the complexity of continuous surfaces, the 3D-WBD algorithm can be employed to calculate intersecting or discontinuous triangular

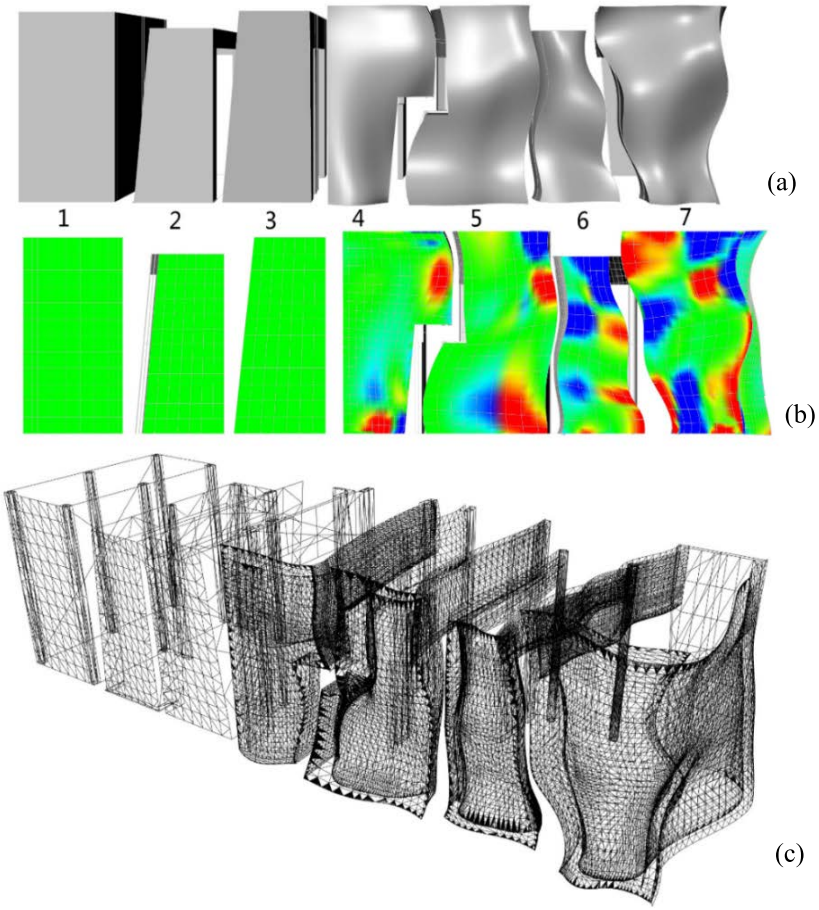


FIGURE 6. Simplified shape model of Yu Qingcheng Art Museum and homotopy change of its surface. (a) Block number of the simplified model of Yuqingcheng Art Museum. (b) Gaussian curvature analysis of the blocks. (c) Triangulation of the space surface of the art gallery.

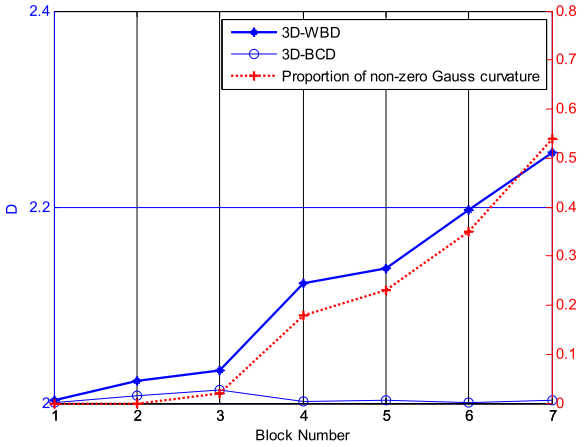


FIGURE 7. Comparison of 3D-WBD and 3D-BCD.

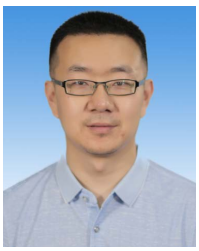
meshes. In additional, 3D-WBD is reduced to 3D-BCD where $\boldsymbol{W} = [1,1,1,1,1,1,1,1]$.

VI. CONCLUSION

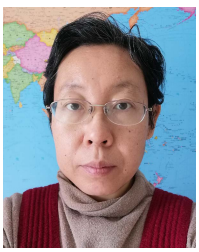
In this study, we propose an improved box dimension calculation method for a 3D model based on collision detection optimized. Key technologies such as octree linear coding, intersection detection of cubic cells and triangles, and the weighted box-counting dimension algorithm were introduced in detail. Several experiments about regular models and typical fractals prove that the improved 3D-BCD method is reliability and accuracy. 3D-WBD method could be employed to calculate the complexity of surface curvature which provides a new reference for the design of building models. The influence of different components of the box dimension, which are classified by the number of children cells owned, can be redistributed. In the future, we will further study the effect of the weighed vector on global or partial box-counting dimensions in practical engineering applications.

REFERENCES

[1] B. Mandelbrot, *The Fractal Geometry of Nature*. New York, NY, USA: Freeman, 1982.
[2] M. V. Berry and Z. V. Lewis, "On the Weierstrass–Mandelbrot fractal function," *Proc. Roy. Soc. London A, Math. Phys. Sci.*, vol. 370, no. 1743, pp. 459–484, 1980.
[3] D. A. Russell, J. D. Hanson, and E. Ott, "Dimension of strange attractors," *Phys. Rev. Lett.*, vol. 45, no. 14, pp. 1175–1178, Oct. 1980.
[4] B. B. Chaudhuri and N. Sarkar, "Texture segmentation using fractal dimension," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 1, pp. 72–77, Jan. 1995.
[5] S. R. Nayak and J. Mishra, "Fractal dimension-based generalized box-counting technique with application to grayscale images," *Fractals*, vol. 29, no. 3, May 2021, Art. no. 2150055.
[6] M. C. Shelberg, N. Lam, and H. Moellering, "Measuring the fractal dimensions of surfaces," Defense Mapping Agency Aerospace Center, St. Louis, MO, USA, Tech. Rep. ADA129664, 1983.
[7] G. Zhou and N. S.-N. Lam, "A comparison of fractal dimension estimators based on multiple surface generation algorithms," *Comput. Geosci.*, vol. 31, no. 10, pp. 1260–1269, Dec. 2005.
[8] A. Napryushkin, V. Kibitkin, and V. Pleshanov, "Linear transformation based error correction algorithm for fractal dimension estimation of images," *Signal Process.*, vol. 90, no. 6, pp. 2094–2101, Jun. 2010.
[9] Q. Xia, T. Li, L. Kang, S. Leng, and X. Wang, "Study on the PTX parameters and fractal characteristics of ore-forming fluids in the east ore section of the pulang copper deposit, southwest China," *J. Earth Sci.*, vol. 32, no. 2, pp. 390–407, Apr. 2021.
[10] F. Martinez, H. Manriquez, A. Ojeda, and G. Olea, "Organization patterns of complex river networks in Chile: A fractal morphology," *Mathematics*, vol. 10, no. 11, pp. 1–23, 2022.
[11] H. Ni, L. Zhou, X. Ning, and L. Wang, "Exploring multifractal-based features for mild Alzheimer's disease classification," *Magn. Reson. Med.*, vol. 76, no. 1, pp. 259–269, Jul. 2016.
[12] H. Pezeshki, M. Rastgarpour, A. Sharifi, and S. Yazdani, "Mass classification of mammograms using fractal dimensions and statistical features," *Multidimensional Syst. Signal Process.*, vol. 32, pp. 573–605, Jan. 2021.
[13] I. Gościniak, K. Gdawiec, K. Woźniak, and M. Machoy, "An approach to determine the features of dental X-ray images based on the fractal dimension," *Proc. Comput. Sci.*, vol. 192, pp. 1856–1865, Jan. 2021.
[14] A. Sangeetha and R. Rajakumari, "Classification of building images using fractal features," *Int. J. Recent Technol. Eng.*, vol. 9, no. 5, pp. 183–185, Jan. 2021.
[15] L. Ma, H. Zhang, and M. Lu, "Building's fractal dimension trend and its application in visual complexity map," *Building Environ.*, vol. 178, Jul. 2020, Art. no. 106925, doi: 10.1016/j.buildenv.2020.106925.
[16] K. Im, J.-M. Lee, U. Yoon, Y.-W. Shin, S. B. Hong, I. Y. Kim, J. S. Kwon, and S. I. Kim, "Fractal dimension in human cortical surface: Multiple regression analysis with cortical thickness, sulcal depth, and folding area," *Human Brain Mapping*, vol. 27, no. 12, pp. 994–1003, Dec. 2006.
[17] K.-K. Shyu, Y.-T. Wu, T.-R. Chen, H.-Y. Chen, H.-H. Hu, and W.-Y. Guo, "Measuring complexity of fetal cortical surface from MR images using 3-D modified box-counting method," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 2, pp. 522–531, Feb. 2011.
[18] W. An, C. Wang, H. Zhang, and Z. Bi, "Measuring the formal complexity of architectural curved surfaces based on 3D box-counting dimension," *Nexus Netw. J.*, vol. 23, pp. 1–14, Nov. 2021.
[19] K. Falconer, *Fractal Geometry: Mathematical Foundations and Applications*. U.K.: Univ. of St Andrews, 1990.
[20] I. Gargantini, "Linear octtrees for fast processing of three-dimensional objects," *Comput. Graph. Image Process.*, vol. 20, no. 4, pp. 365–374, Dec. 1982.
[21] T. Möller, "A fast triangle-triangle intersection test," *J. Graph. Tools*, vol. 2, no. 2, pp. 25–30, Jan. 1997.
[22] L. Wu and Z. Hua, "The application of fractal language in architecture: The design of Yu Qingcheng art museum," *New Archit.*, vol. 37, no. 1, 2019.
[23] H. Qin, K. Xu, and L. Jiang, "Fractal scaleless band automatic identification for fractaltheory application," *Chin. J. Mech. Eng.*, vol. 42, no. 12, p. 106, 2006.



CHONG WANG received the master's degree in software engineering and the Ph.D. degree in mining engineering from the Shandong University of Science and Technology, in 2007 and 2016, respectively. He is currently working as a Postdoctoral Fellow in civil engineering with the School of Civil Engineering and Architecture, Shandong University of Science and Technology. His primary research interests include computer graphics and computational mechanics.



WEIQIANG AN received the master's degree in mathematics and architecture from Qingdao Technological University, in 2009, and the Ph.D. degree in architecture from Tianjin University, in 2022. She is currently working with the School of Civil Engineering and Architecture, Shandong University of Science and Technology. Her primary research interests include parametric design and fractal architecture.