

## Research Article

# Dominant Symmetry Plane Detection for Point-Based 3D Models

Chen He <sup>1</sup>, Lei Wang,<sup>2</sup> Yonghui Zhang,<sup>2</sup> and Chunmeng Wang <sup>3</sup>

<sup>1</sup>Media and Communication College, Weifang University, Weifang, China

<sup>2</sup>Computer Engineering College, Weifang University, Weifang, China

<sup>3</sup>Computer Engineering College, Jinling Institute of Technology, Nanjing, China

Correspondence should be addressed to Chen He; [imhechen@163.com](mailto:imhechen@163.com) and Chunmeng Wang; [wchm87@jit.edu.cn](mailto:wchm87@jit.edu.cn)

Received 19 August 2020; Revised 28 September 2020; Accepted 16 October 2020; Published 28 October 2020

Academic Editor: Patrick Seeling

Copyright © 2020 Chen He et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, a symmetry detection algorithm for three-dimensional point cloud model based on weighted principal component analysis (PCA) is proposed. The proposed algorithm works as follows: first, using the point element's area as the initial weight, a weighted PCA is performed and a plane is selected as the initial symmetry plane; and then an iterative method is used to adjust the approximate symmetry plane step by step to make it tend to perfect symmetry plane (dominant symmetry plane). In each iteration, we first update the weight of each point based on a distance metric and then use the new weights to perform a weighted PCA to determine a new symmetry plane. If the current plane of symmetry is close enough to the plane of symmetry in the previous iteration or if the number of iterations exceeds a given threshold, the iteration terminates. After the iteration is terminated, the plane of symmetry in the last iteration is taken as the dominant symmetry plane of the model. As shown in experimental results, the proposed algorithm can find the dominant symmetry plane for symmetric models and it also works well for nonperfectly symmetric models.

## 1. Introduction

Symmetry exists widespread in nature and various man-made objects. In recent years, symmetry detection has drawn increasingly more attention in research areas of computer graphics and computer vision. Symmetry detection is an important problem in 3D shape analysis [1]. First, the essence of symmetry detection is to compute a nontrivial correspondence of the model (or its subpart) to itself. Correspondence analysis is a basic problem in shape analysis. Second, symmetry is a bridge between low-level geometry attributes and high-level shape information of the models: on one hand, symmetry is a geometric property that can be detected by measuring and analyzing the geometric properties of the model; on the other hand, symmetry describes the shape and structure of the model and even the semantic information. Especially in man-made objects, symmetry often reflects the creator's design intent and even suggests the functional components of the object.

The geometric symmetry of a three-dimensional object refers to the invariance of the shape as a whole or a part

under a certain transformation. According to the geometric properties that the symmetry depends on, the geometric symmetry can be divided into two types: extrinsic symmetry and intrinsic symmetry. The extrinsic symmetry is defined by the Euclidean distance, while the intrinsic symmetry is based on geodesic distance on the surface. Since the calculation of the intrinsic distance is very complicated, it is often more difficult to represent and calculate the intrinsic symmetry. Intrinsic symmetry is invariant to the isometric transformation of a surface. Besides, symmetry can be classified into global symmetry and local symmetry according to the distribution of the symmetrical region on the surface of the three-dimensional model.

The existing symmetric analysis of three-dimensional model mainly focuses on the detection of various types of geometric symmetry in the model and can be generally divided into four categories: global extrinsic symmetry, local extrinsic symmetry, global intrinsic symmetry, and local intrinsic symmetry detection. A brief introduction to some existing symmetry detection algorithms in related work is discussed below.

## 2. Related Work

The global extrinsic symmetry is the simplest form of symmetry, so the research work on it started earliest. Minovic et al. [2] proposed a three-dimensional model symmetry detection algorithm based on octree representation. They use PCA analysis to calculate the principal axes of the input objects, determine a local coordinate system, and construct the corresponding octree representation. In this way, the symmetry of the input object can be determined by calculating the “degree of symmetry” of the symmetrical cells in the octree. Zabrodsky and Weinshall [3] use the transformed symmetry metric to measure the distance from one shape to its nearest symmetric shape, achieving approximate symmetry detection. Kazhdan et al. [4] proposed a symmetric transformation that measures the degree of symmetry with respect to any straight line in the plane for a two-dimensional shape. This method was later extended to a three-dimensional geometric model by Podolak et al. [5] and proposed the concept of planar reflective symmetry transform (PRST). Rustamov [6] incorporated the asymmetric spatial distribution of objects into PRST, resulting in an enhanced PRST for efficient shape retrieval. Martinet et al. [7] defined a kind of generalized moments based on spherical harmonic coefficients, which can detect both discrete and continuous global symmetries simultaneously, and proposed a method that can recover the global symmetry of 3D models.

Relative to the global extrinsic symmetry, the detection of local extrinsic symmetry is more difficult. This is because local symmetry detection needs to search for symmetric subareas on the surface of the model, while the subset search space is very large. Thrun and Wegbreit [8] presented a hierarchical production test algorithm for the local symmetry detection of the model. Simari et al. [9] studied the local approximate planar reflection symmetry problem of the mesh model and defined a hierarchical representation of the three-dimensional shape based on the planar reflection symmetry. Mitra et al. [10] proposed an algorithm to detect partial symmetries based on the transformation space voting and clustering. They considered the invariance of the model in the conversion process such as reflection, translation, uniform scale, and rotation is symmetrical. Similarly, Loy and Eklundh [11] also studied the method of symmetry detection based on transformation space. Based on the detected local extrinsic symmetry, Mitra et al. [12] further studied the symmetrization of the three-dimensional models; that is, applying a deformation to a model makes it a global extrinsic symmetry one.

One of the problems associated with the detection of local extrinsic symmetry is repetitive pattern detection. In recent years, the detection of repetitive patterns in three-dimensional models has also received widespread attention. Repetitive patterns can be formally described by symmetric groups, typically expressed as multiple translations and/or rotations of the model’s subparts. Pauly et al. [13] first proposed the repetitive pattern detection on 3D models. This work uses the symmetric group to formally describe the repetitive patterns on the model. The symmetric group

parameters are obtained by clustering in the transformation space to realize the automatic detection of repetitive patterns. Bokeloh et al. [14] utilized the feature lines of the model to construct graph representations of repetitive patterns, which transforms the detection problems of repetitive patterns into subgraph matching problems. Liu et al. [15] studied the detection of periodic reliefs on 3D meshes. Yeh and M ech [16] studied repetitive pattern detection problems in 2D line art. Recently, Ben-Chen et al. [17] studied the calculation of the discrete Killing vector field (KVF) and applied it to the detection of the repeated structures on three-dimensional surfaces.

For the intrinsic symmetry detection of the model, most of the current methods only consider the global intrinsic symmetry. Raviv et al. [18] presented an algorithm for a set of sample points on manifold surfaces to find an isometric transformation from the set to itself. Global point signature (GPS) [19] can translate the global intrinsic symmetry of a model into the global extrinsic symmetry in high-dimensional space. Using GPS, Ovsjanikov et al. [20] proposed a pure algebraic algorithm to detect global intrinsic symmetry. This method can divide the symmetry into several discrete equivalence classes. Similarly, based on spectral analysis, Chertok and Keller [21] presented an intrinsic symmetry detection algorithm. Kim et al. [22] presented a global intrinsic symmetry detection method based on the Mobius transform.

As for local intrinsic symmetry detection, the existing work is still relatively few. Bronstein et al. [23] studied the local intrinsic similarity of nonrigid models. However, this method tends to find the maximum symmetry region and, therefore, not suitable for local intrinsic symmetry detection. Xu et al. [24] first solved the problem of local intrinsic symmetry detection in 3D models. They give the definition of local intrinsic reflection symmetry on a closed 2-manifold and propose a method based on voting strategy. In addition, they also realized the meaningful segmentation of the model based on local intrinsic reflection symmetry. Lipman et al. [25] presented an algorithm for detecting the orbits of symmetric groups of three-dimensional point clouds. The method first constructs a symmetry correspondence matrix for the input point cloud and then uses the spectral characteristics of the matrix to define a symmetry factored distance on the three-dimensional surface. This method can stably detect the local intrinsic symmetry existing in 3D point cloud data.

Recently, Li et al. [26, 27] present a simple and efficient view-based reflection symmetry detection method based on the viewpoint entropy features of a set of sample views of a 3D model. Sipiran et al. [28] present an algorithm to detect symmetry planes in objects with partial data. The method is based on the ability of finding good candidate features. For this reason, the method is not suitable for feature-less shapes. Ecins et al. [29] present an approach to detecting symmetric objects and extracting their symmetries from three-dimensional data. The method can be used to detect symmetric objects in scenes consisting of synthetic 3D models, as well as 3D scans of real environments. Li et al. [30] propose an FSM- (frequent subgraph mining-) based

symmetry detection approach for CAD models, but this approach mainly works for regular CAD models. Nagar and Raman [31] develop a generic framework to detect approximate reflection symmetry in a volumetric set of points in  $R^d$  using optimization on Riemannian manifold. They show the effectiveness of the method by applying it to the problem of 2D and 3D reflection symmetry detection.

### 3. Algorithm Description

The proposed algorithm is based on the point cloud model represented by surfel (surface element) set. For more information about this format model, see Pointshop 3D's homepage at <http://graphics.ethz.ch/pointshop3d/>. In this representation, each point is rendered as a small disc containing information such as position, normal vector, color, and radius. Simari et al. [9] have proposed an iterative reweighted least-squares algorithm for the detection of planar reflection symmetry of 3D mesh models. Different from Simari et al's algorithm, we propose an algorithm that can be used to detect the dominant symmetry plane for the point cloud models.

**3.1. Overview.** As we all know, the principal component analysis (PCA) method can generally be used to detect the symmetry plane for a perfectly symmetric model. The PCA method firstly calculates the model's covariance matrix and then calculates the eigenvectors of the matrix. The symmetry plane is perpendicular to one of the eigenvectors and passes through the centroid of the model. However, for a nonperfectly symmetric model, using only one PCA procedure cannot find the dominant symmetry plane of the model. For example, the symmetry plane detected by only one PCA process is obviously incorrect for the bunny model shown in Figure 1.

In this paper, an iterative, reweighted PCA algorithm is proposed to detect the symmetry of the point cloud model. The algorithm is implemented as follows: first, using the point element's area as the initial weight, a weighted PCA is performed and a plane is selected as the initial symmetry plane, and then an iterative method is used to adjust the approximate symmetry plane step by step to make it tend to be a perfect symmetry plane (dominant symmetry plane). In each iteration, we first update the weight of each point based on a distance measure and then use the new weights to perform a weighted PCA to determine a new symmetry plane. If the current plane of symmetry is close enough to the plane of symmetry in the previous iteration or if the number of iterations exceeds a given threshold, the iteration terminates. After the iteration is terminated, the plane of symmetry in the last iteration is taken as the dominant symmetry plane of the model. The algorithm flow chart is shown in Figure 2, and the details of the algorithm will be described in the following sections.

**3.2. Weighted Principal Component Analysis.** For  $n$  surfels in a given point cloud model, suppose the positions are  $\mathbf{p}_i$  ( $i = 1,$

$2, \dots, n$ ), and the corresponding weights for the surfels are  $w_i$  ( $i = 1, 2, \dots, n$ ), then the covariance matrix of the model is computed as follows:

$$C = \frac{1}{s} \sum_{i=1}^n w_i (\mathbf{p}_i - m)(\mathbf{p}_i - m)^T. \quad (1)$$

In this equation,  $m = (1/s) \sum_{i=1}^n w_i \mathbf{p}_i$  and  $s = \sum_{i=1}^n w_i$ . Having the matrix  $C$ , the eigenvectors of the matrix are computed, and these eigenvectors and the centroid  $m$  determine three mutually orthogonal planes,  $P_1, P_2,$  and  $P_3$ . For each plane, a value of symmetry degree is computed and the plane with the smallest symmetry degree is chosen as the principal symmetry plane. From this procedure, we can see that the key problem of weighted PCA is how to calculate the weight of the surfels and how to calculate the symmetry degree of a plane.

**3.3. Calculation of Weight and Symmetry Degree.** Before calculating the weight of a point, we first calculate a distance. When calculating the distance for the  $i$ -th surfel  $s_i$ , the reflection symmetry point of  $s_i$  with respect to current symmetry plane  $P$  is first calculated and this reflection point is marked as  $s_{ir}$ . Then, for all the surfels  $s_j$  which are located on the same side of plane  $P$  as the point  $s_{ir}$ , the Euclidean distance between  $s_{ir}$  and  $s_j$  is computed if the angle between the normal vectors of  $s_{ir}$  and  $s_j$  is less than a given threshold  $T_\alpha$  (30 degrees in our experiment). Finally, the minimum value of these distances is used as the distance metric for the  $i$ -th surfel, marked as  $d_i$ . The 2D illustration of  $d_i$  is shown in Figure 3. Intuitively speaking, this distance metric can be used to measure the symmetry degree of a point about the current plane  $P$ . The smaller the distance, the better the model's symmetry at this point. If the model is perfectly symmetric relative to the current plane  $P$  at this point,  $d_i$  should tend to zero. With this distance metric, the weight  $w_i$  can be calculated as follows:

$$w_i = \begin{cases} \frac{2\delta^2}{(\delta^2 + d_i^2)^2}, & \text{if } d_i < \delta, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

In this equation,  $\delta = c \times \text{median}(d_i)$ , where  $c$  is a constant defined by the user. In our experiment,  $c$  is set to 1.5. With this distance metric, the symmetry degree of plane  $P$  is defined as  $m_s = \sum_{i=1}^n d_i$ .

**3.4. Convergence Analysis.** In the proposed algorithm, convergence is detected by comparing the current symmetry plane and previous symmetry plane. Suppose the equation of current symmetry plane is  $A_1x + A_2y + A_3z + A_4 = 0$ , and the equation of previous symmetry plane is  $B_1x + B_2y + B_3z + B_4 = 0$ , then the proximity of this two planes can be measured by the following equation:

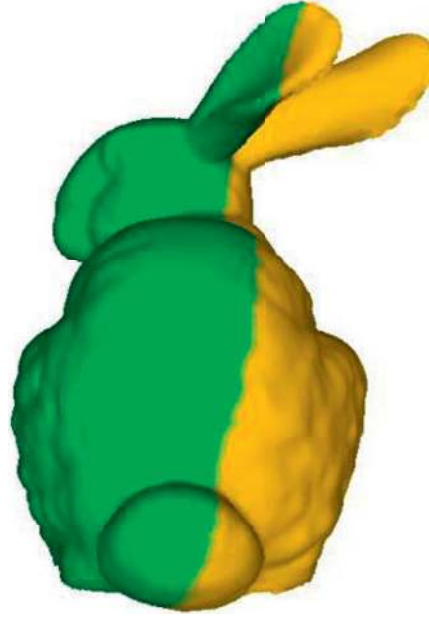


FIGURE 1: Symmetry plane obtained by one PCA process, taking bunny model for example.

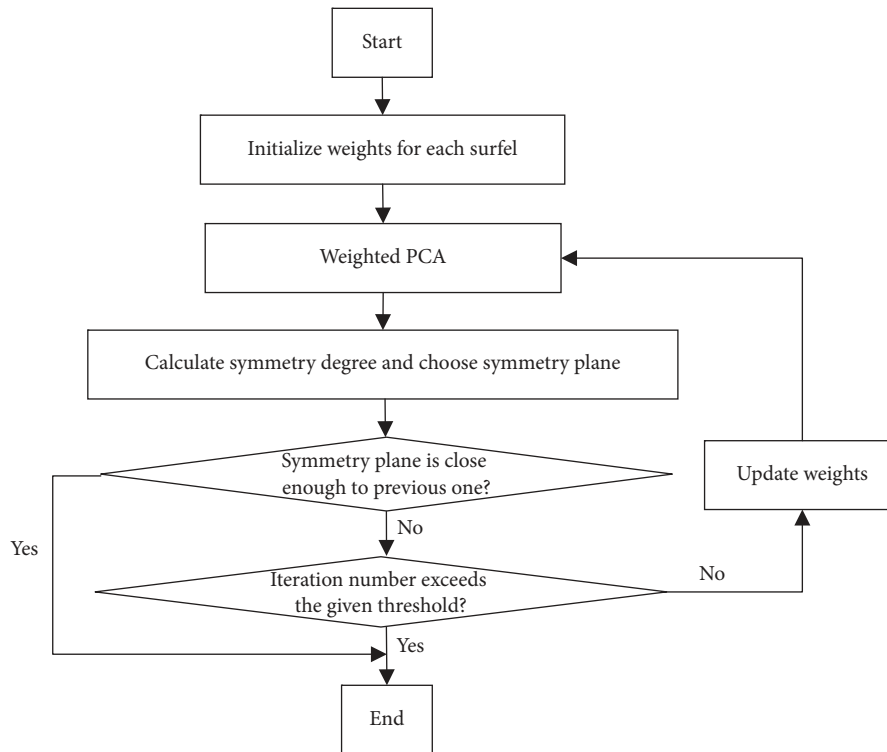


FIGURE 2: The flowchart of the proposed symmetry detection algorithm.

$$\varepsilon = \sqrt{\sum_{i=1}^4 (A_i - B_i)^2}. \quad (3)$$

In our experiments, the proposed algorithm exhibits good convergence behaviour on all the tested models. Figure 4 illustrates the convergence property of the proposed

algorithm, taking the bunny model for example. In Figure 4, the horizontal axis represents iteration numbers and the vertical axis represents the value of  $\varepsilon$ . The estimated symmetry planes of the bunny model are also shown in Figure 4. From left to right, the iteration numbers are 1, 2, and 7, respectively. After the experiment, we found that the two symmetry planes are close enough when  $\varepsilon$  is less than 0.1.

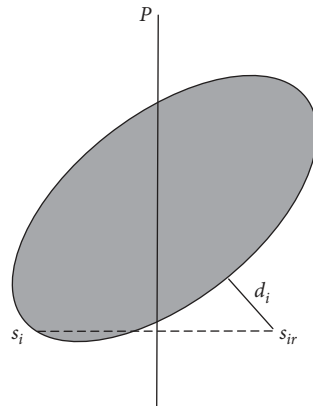
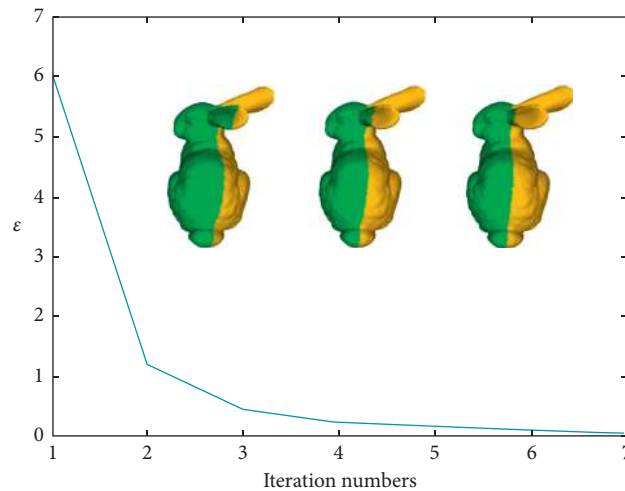
FIGURE 3: The 2D illustration of distance metric  $d_i$ .

FIGURE 4: The convergence property of the proposed algorithm.

#### 4. Results and Discussion

In the experiment, we use five models to test the effectiveness of our symmetry detection algorithm, which are shown in Figure 5. They are face by the courtesy of Pointshop 3D, bunny from the Stanford 3D scanning repository, horse from Aim@Shape, and banana and rabbit from the Internet. Under each model, the number of points is marked. It should be noted that we obtained the surfel set models for bunny, horse, banana, and rabbit from the original 3D meshes by subdividing each triangle into four subtriangles and placing a surfel at the center of each subtriangle.

Figure 6 shows some intermediate results of the algorithm during iteration, taking the rabbit model as an example. From this figure, we can see that starting from the

initial plane, after each iteration, the plane gradually approaches the dominant symmetry plane of the model, indicating that our algorithm is correct.

The result of the proposed algorithm running on several point-based models is shown in Figure 7. In this figure, the symmetry plane is shown by coloring the two sides of the plane with different colors. As shown in the figure, for perfectly symmetrical models such as the face model and the banana model, our algorithm can accurately detect the symmetry plane, and the algorithm also works well for nonperfectly symmetric models such as bunny, horse, and rabbit models.

We compare our method with the traditional PCA method on bunny, horse, and rabbit models, shown in Figure 8. We also show the timing statistics in Table 1. From

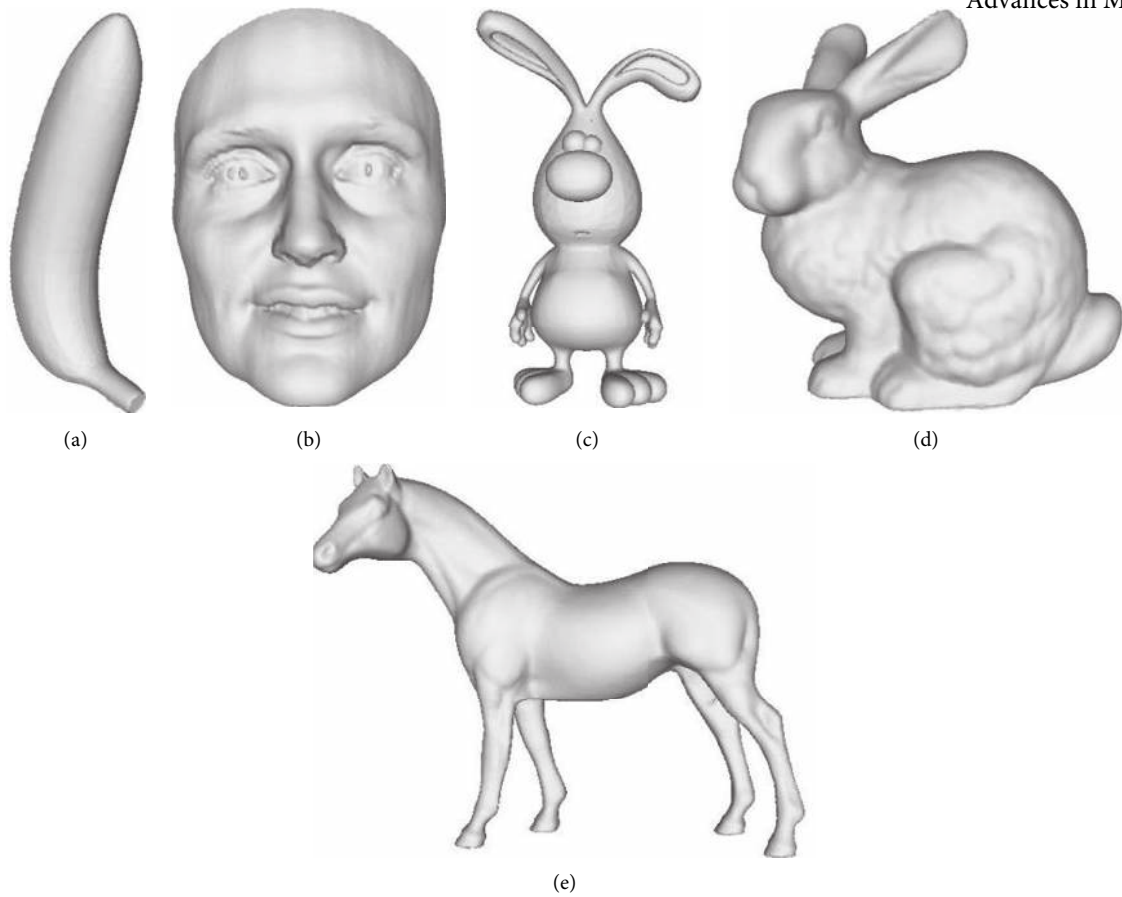


FIGURE 5: Models used in the experiment. For each model, the numbers in parentheses indicate the number of points. (a) Banana (25k). (b) Face (40k). (c) Rabbit (189k). (d) Bunny (20k). (e) Horse (158k).

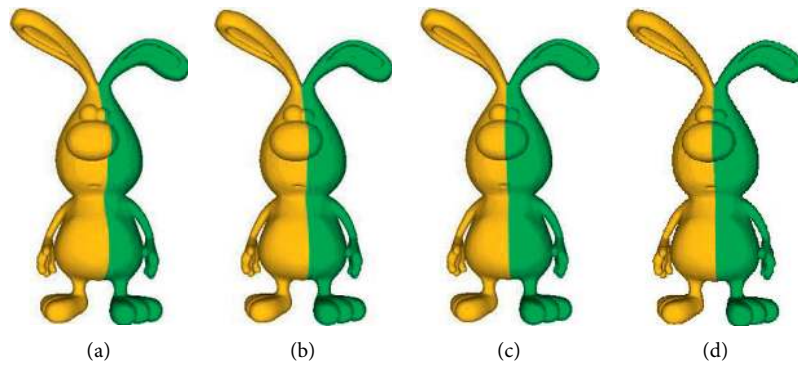


FIGURE 6: The iteration procedure of the proposed algorithm, taking rabbit model for example. (a)  $i=0$ . (b)  $i=1$ . (c)  $i=2$ . (d)  $i=3$ .

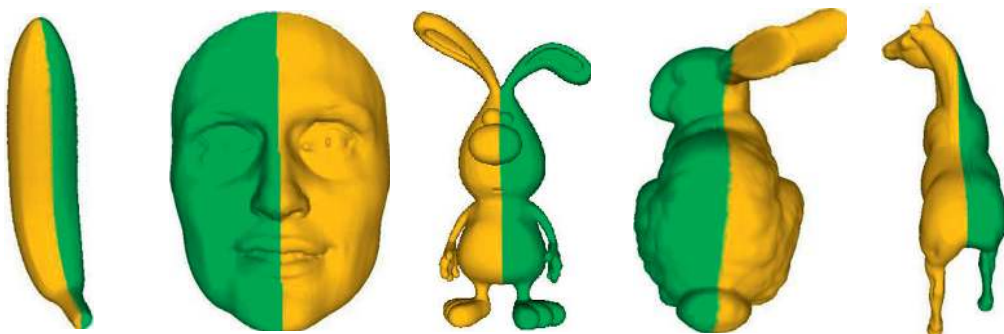


FIGURE 7: Experimental results for different point-based models.

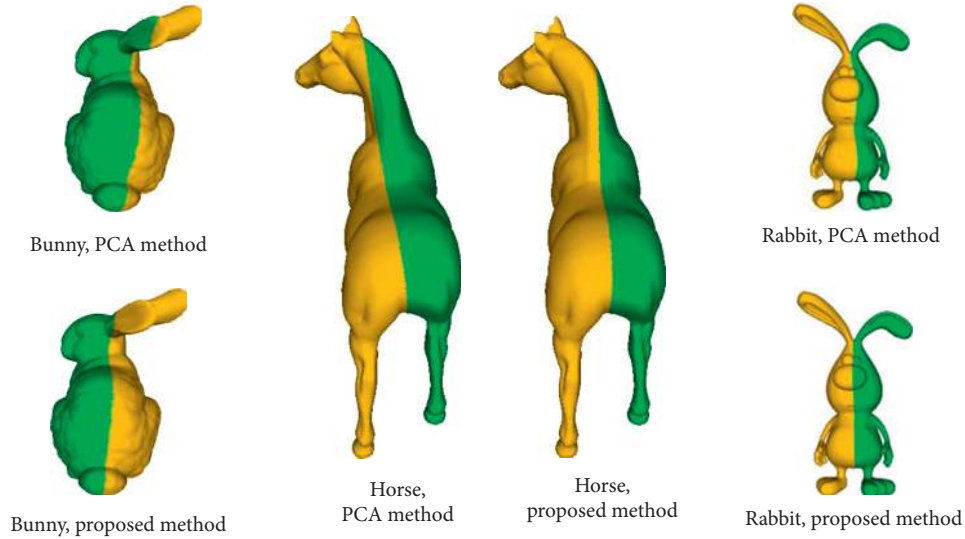


FIGURE 8: Comparison results of the proposed method and traditional PCA method for bunny, horse, and rabbit models.

TABLE 1: Running time of the proposed method and PCA method.

Models	Running time (s)/proposed method	Running time (s)/PCA method
Bunny	4.57	1.52
Horse	462.14	89.43
Rabbit	521.77	94.35

this comparison, we can see that although our method takes more time, it works well on detecting the dominant symmetry plane for nonperfectly symmetric models, while the traditional PCA method does not work.

## 5. Conclusion and Future Work

This article presented a simple yet efficient symmetry detection algorithm for three-dimensional point cloud models. Based on weighted principal component analysis, the proposed algorithm first uses the area of each point element as the weight, performs a weighted PCA to determine an initial symmetry plane, and then iteratively update the weights and adjust this approximate symmetry plane step by step to the perfect symmetry plane (dominant symmetry plane).

The future research can be carried out from the following two aspects: first, the detection of partial symmetry that exists in the models, for example, the two ears of the bunny model; second, the detection of other types of symmetries in 3D models, such as rotational symmetry, translational symmetry, and so on.

## Data Availability

The data used to support the findings of the study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Authors' Contributions

C.H. and C.W. conceived and designed the experiments. C.H. performed the experiments. C.H. and C.W. wrote the manuscript. L.W. and Y.Z. edited the manuscript.

## Acknowledgments

This work was supported by the Shandong Provincial Natural Science Foundation, China (Grant no. ZR2019PF023), the National Natural Science Foundation of China (Grant no. 31872847), the Project of Shandong Provincial Higher Educational Science and Technology Program (Grant nos. J17KB184 and J18KA130), the Science and Technology Development Plan Project of Weifang City (Grant nos. 2019GX005 and 2017GX006), the Doctoral Research Foundation of Weifang University (Grant no. 2015BS12), the General Program of Natural Science Foundation of the Jiangsu Higher Education Institutions of China (no. 19KJB520007), and the Project of High-level Talents Research Foundation of Jinling Institute of Technology (jit-b-201802).

## References

- [1] A. Berner, M. Wand, N. J. Mitra, D. Mewes, and H.-P. Seidel, "Shape analysis with subspace symmetries," *Computer Graphics Forum (Special Issue of Eurographics 2011)*, vol. 30, 2011.
- [2] P. Minovic, S. Ishikawa, and K. Kato, "Symmetry identification of a 3-D object represented by octree," *IEEE*

- Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 5, pp. 507–514, 1993.
- [3] H. Zabrodsky and D. Weinshall, “Using bilateral symmetry to improve 3D reconstruction from image sequences,” *Computer Vision and Image Understanding*, vol. 67, no. 1, p. 48, 1997.
  - [4] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, “Symmetry descriptors and 3D shape matching,” in *Proceedings of the 2004 Symposium on Geometry Processing*, pp. 115–123, Nice, France, 2004.
  - [5] J. Podolak, P. Shilane, A. Golovinskiy, S. Rusinkiewicz, and T. Funkhouser, “A planar-reflective symmetry transform for 3D shapes,” *ACM Transactions on Graphics*, vol. 25, no. 3, p. 549, 2006.
  - [6] R. M. Rustamov, “Augmented planar reflective symmetry transform,” *The Visual Computer*, vol. 24, no. 6, p. 423, 2008.
  - [7] A. Martinet, C. Soler, N. Holzschuch, and F. X. Sillion, “Accurate detection of symmetries in 3D shapes,” *ACM Transactions on Graphics*, vol. 25, no. 2, p. 439, 2006.
  - [8] S. Thrun and B. Wegbreit, “Shape from symmetry,” in *Proceedings of the 2005 International Conference on Computer Vision*, pp. 1824–1831, Las Vegas, NV, USA, 2005.
  - [9] P. Simari, E. Kalogerakis, and K. Singh, “Folding meshes: hierarchical mesh segmentation based on planar symmetry,” in *Proceedings of the 4th Eurographics Symposium on Geometry Processing*, pp. 111–119, Sardinia, Italy, 2006.
  - [10] N. J. Mitra, L. J. Guibas, and M. Pauly, “Partial and approximate symmetry detection for 3D geometry,” *ACM Transactions on Graphics*, vol. 25, no. 3, p. 560, 2006.
  - [11] G. Loy and J.-O. Eklundh, “Detecting symmetry and symmetric constellations of features,” in *Proceedings of the 2006 European Conference on Computer Vision*, pp. 508–521, Graz, Austria, 2006.
  - [12] N. J. Mitra, L. J. Guibas, and M. Pauly, “Symmetrization,” *ACM Transactions on Graphics*, vol. 26, no. 3, pp. 1–8, 2007.
  - [13] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. J. Guibas, “Discovering structural regularity in 3D geometry,” *ACM Transactions on Graphics*, vol. 27, no. 3, p. 1, 2008.
  - [14] M. Bokeloh, A. Berner, M. Wand, H.-P. Seidel, and A. Schilling, “Symmetry detection using feature lines,” *Computer Graphics Forum*, vol. 28, no. 2, p. 697, 2009.
  - [15] S. Liu, R. Martin, F. C. Langbein, and P. L. Rosin, “Segmenting periodic reliefs on triangle meshes,” in *Proceedings of the 12th IMA International Conference on Mathematics of Surfaces*, pp. 290–306, Sheffield, UK, 2007.
  - [16] Y.-T. Yeh and R. Mëch, “Detecting symmetries and curvilinear arrangements in vector art,” *Computer Graphics Forum*, vol. 28, no. 2, p. 707, 2009.
  - [17] M. Ben-Chen, A. Butscher, J. Solomon, and L. Guibas, “On discrete killing vector fields and patterns on surfaces,” *Computer Graphics Forum*, vol. 29, no. 5, p. 1701, 2010.
  - [18] D. Raviv, A. M. Bronstein, M. M. Bronstein, and R. Kimmel, “Symmetries of non-rigid shapes,” in *Proceedings of the 2007 International Conference on Computer Vision (ICCV’2007)*, pp. 1–7, Rio de Janeiro, Brazil, 2007.
  - [19] R. M. Rustamov, “Laplace-Beltrami eigenfunctions for deformation invariant shape representation,” in *Proceedings of the 2007 Eurographics Symposium on Geometry Processing (SGP’2007)*, pp. 225–233, Barcelona, Spain, 2007.
  - [20] M. Ovsjanikov, J. Sun, and L. Guibas, “Global intrinsic symmetries of shapes,” *Computer Graphics Forum*, vol. 27, no. 5, p. 1341, 2008.
  - [21] M. Chertok and Y. Keller, “Spectral symmetry analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, p. 1227, 2010.
  - [22] V. G. Kim, Y. Lipman, X. Chen, and T. Funkhouser, “Möbius transformations for global intrinsic symmetry analysis,” *Computer Graphics Forum*, vol. 29, no. 5, p. 1689, 2010.
  - [23] A. M. Bronstein, M. M. Bronstein, A. M. Bruckstein, and R. Kimmel, “Partial similarity of objects, or how to compare a centaur to a horse,” *International Journal of Computer Vision*, vol. 84, no. 2, p. 163, 2009.
  - [24] K. Xu, H. Zhang, A. Tagliasacchi et al., “Partial intrinsic reflectional symmetry of 3D shapes,” *ACM Transactions on Graphics (TOG)*, vol. 28, no. 5, pp. 1–10, 2009.
  - [25] Y. Lipman, X. Chen, I. Daubechies, and T. Funkhouser, “Symmetry factored embedding and distance,” *ACM Transactions on Graphics*, vol. 29, no. 4, p. 1, 2010.
  - [26] B. Li, H. Johan, Y. Ye et al., “Efficient view-based 3D reflection symmetry detection,” in *Proceedings of the 2014 International Conference on Computer Graphics and Interactive Techniques*, Vancouver, Canada, 2014.
  - [27] B. Li, H. Johan, Y. Ye, and Y. Lu, “Efficient 3D reflection symmetry detection: a view-based approach,” *Graphical Models*, vol. 83, pp. 2–14, 2016.
  - [28] I. Sipiran, R. Gregor, and T. Schreck, “Approximate symmetry detection in partial 3D meshes,” *Computer Graphics Forum*, vol. 33, no. 7, pp. 131–140, 2015.
  - [29] A. Ecins, C. Fermuller, and Y. Aloimonos, “Detecting reflectional symmetries in 3D data through symmetrical fitting,” in *Proceedings of the 2017 IEEE International Conference on Computer Vision Workshop (ICCV)*, Venice, Italy, 2017.
  - [30] C. Li, M. Li, and S. Gao, “Multi-scale symmetry detection of CAD models,” *Computer Aided Design & Applications*, vol. 16, no. 1, pp. 50–66, 2019.
  - [31] R. Nagar and S. Raman, “Detecting approximate reflection symmetry in a point set using optimization on manifold,” *IEEE Transactions on Signal Processing*, vol. 67, no. 6, pp. 1582–1595, 2019.