

Received February 9, 2020, accepted February 21, 2020, date of publication February 27, 2020, date of current version March 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2976847

Developing a Semantic-Driven Hybrid Segmentation Method for Point Clouds of 3D Shapes

XIAOWEN YANG¹, XIE HAN¹, QINGDE LI², LIGANG HE³, (Member, IEEE),
MIN PANG¹, AND CAIQIN JIA¹

¹School of Data Science and Technology, North University of China, Taiyuan 030051, China

²Department of Computer Science, University of Hull, Hull HU6 7RX, U.K.

³Department of Computer Science, University of Warwick, Coventry CV4 7AL, U.K.

Corresponding author: Xie Han (hanxie@nuc.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (61672473), in part by the Shanxi Provincial Key Research and Development Project (201803D121081 and 201903D121147), and in part by the Natural Science Foundation of Shanxi Province of China (201901D111150).

ABSTRACT With the rapid development of point cloud processing technologies and the availability of a wide range of 3D capturing devices, a geometric object from the real world can be directly represented digitally as a dense and fine point cloud. Decomposing a 3D shape represented in point cloud into meaningful parts has very important practical implications in the fields of computer graphics, virtual reality and mixed reality. In this paper, a semantic-driven automated hybrid segmentation method is proposed for 3D point cloud shapes. Our method consists of three stages: semantic clustering, variational merging, and region remerging. In the first stage, a new feature of point cloud, called Local Concave-Convex Histogram, is introduced to first extract saddle regions complying with the semantic boundary feature. All other types of regions are then aggregated according to this extracted feature. This stage often leads to multiple over-segmentation convex regions, which are then remerged by a variational method established based on the narrow-band theory. Finally, in order to recombine the regions with the approximate shapes, order relation is introduced to improve the weighting forms in calculating the conventional Shape Diameter Function. We have conducted extensive experiments with the Princeton Dataset. The results show that the proposed algorithm outperforms the state-of-the-art algorithms in this area. We have also applied the proposed algorithm to process the point cloud data acquired directly from the real 3D objects. It achieves excellent results too. These results demonstrate that the method proposed in this paper is effective and universal.

INDEX TERMS Semantic-driven, local concave-convex histogram, variational method, shape diameter function.

I. INTRODUCTION

3D shape segmentation is a fundamental task in geometric information processing, geometric object recognition and reconstruction, and computer graphics. Depending on the way in which a 3D shape is represented, its segmentation can be represented as a set of the topological adjacent points, edges or faces. The main purpose of the 3D shape segmentation is to subdivide a given 3D shape into simple surface patches or meaningful parts [1]. By 3D shape segmentation, texture mapping for a complex geometry, which is a very difficult task, can be reduced to a set of simpler tasks of mapping

textures on these simple geometric patches, although these patches do not have any practical meaning. More importantly, after segmenting a 3D shape into parts, it becomes easy to perform local shape deformation and editing as well as shape reconstruction, retrieval and analysis.

With the ever increasing development of the devices for 3D data acquisition and other related technologies, it becomes increasingly easy to capture the 3D points. Thanks to these developments, the 3D point cloud is able to contain rich information nowadays. There are now a wide range of the applications which make use of 3D point cloud data [2], [3].

A highly dense 3D point cloud is a basic representation of a 3D shape. It can collectively provide detailed and relatively complete geometric information of the original 3D shape.

The associate editor coordinating the review of this manuscript and approving it for publication was Jiachen Yang¹.

However, the point cloud is a data-intensive geometric representation. For instance, this form of point cloud has to be very big to provide the relatively accurate geometry representation for a geometric object.

Since a 3D point cloud is simply a collection of discrete points, its irregular and disordered features make it difficult to segment a point cloud into meaningful parts directly. Therefore, a point cloud can be pre-processed in practice, for instance, by converting a point cloud into a solid shape through the process of the voxelization. However, this traditional process often leads to the change in the original detailed geometric features captured in the point cloud, and consequently the unnecessary loss of detailed data and space consumption, which will add extra workload in later processing stages [4].

In recent years, some researchers have adopted different voxel resolutions [5]–[8] to reduce the loss of detailed data. However, the techniques proposed in these works require a large dataset to train the segmentation process. The methods have also been proposed to optimize the voxelization method for segmenting objects [9], [10]. However, their methods are not designed for segmenting a single object, but segmenting objects from a scene, which manifests different boundary feature (i.e., normal vector) from segmenting a single object (i.e., use saddle characteristics as the boundary feature). We will review these studies in more details in the related work.

By using the Bearing Angle image (BA image) [11]–[13] or the Optimal Bearing Angle image (OBA image) [14], a 3D laser point cloud can be transformed directly to a 2D image to effectively characterize the depth discontinuities and the direction changes of the edges and contours for the objects from the indoor or outdoor scenes. Then the superpixel and K-Means methods can be used to cluster the feature of the 2D image to segment the objects from the scene. The experimental results show the validity of the BA image and the OBA image and their superior performance in object detection, recognition and understanding from the scenes. The BA image and the OBA image need to be computed by the 2D regular structure of a 3D laser point cloud. However, the point cloud from a single 3D shape is irregularly arranged and disordered. Therefore, the BA image or the OBA image is not suitable to be applied directly to the segmentation of 3D shapes. The voxelization process divides a 3D shape into a fixed number of voxels in each direction, which may form the regular structure of the point cloud. However, due to the limitation of the regular structure, the details of the boundary regions with the saddle characteristic will be lost. The point cloud of a single 3D shape with the continuity does not manifest the significant depth discontinuities and direction changes, and the voxelization further weakens the depth and direction difference between the neighbors. Hence, the boundary regions between the parts in the point cloud do not show obvious BA image feature or OBA image feature to segment a single 3D shape effectively. Therefore, Bearing Angle image and Optimal Bearing Angle image are not

suitable for the 3D shape segmentation in this paper. The image segmentation algorithms based on superpixel for the BA image or the OBA image are not suitable for segmenting a single 3D shape.

In many fields such as digital heritage protection, virtual manufacturing, building design, digital entertainment, medical imaging, robot navigation, smart city[4], the processing of the point clouds, which includes the point cloud segmentation, object understanding and recognition, and shape reconstruction, has become a popular field of research. As the point cloud segmentation is the most fundamental step for many point cloud-related applications, this paper focuses on the segmentation of the point cloud-based 3D shapes. The segmentation of a single 3D shape is mainly applied to the applications such as part-based 3D shape synthesis, part-based 3D shape editing, part-based 3D shape recognition, part-based 3D shape retrieval and so on. In these applications, the selection or matching is performed from the database of 3D parts that are obtained offline. These applications emphasize the segmentation quality, not the segmentation speed, unlike other applications such as object detection and recognition by service robots. Therefore, there is no need to obtain the segmentation results in real time for these applications.

Based on whether the data have to be labelled, the 3D shape segmentation algorithms can be divided into two groups: supervised and unsupervised 3D shape segmentation. The state-of-the-art supervised 3D shape segmentation algorithms can achieve the result close to that by human vision. However, such supervised algorithms require that the input data are labelled, which involves a very labor-intensive and time-consuming process, and a large amount of data have to be used for the purpose of the training. Thus, the efficiency of this kind of algorithms is not high in general. In practice, the point cloud acquired from a scanner is in general unlabelled. In order to directly segment the 3D point cloud effectively, it is necessary to develop the unsupervised 3D shape segmentation algorithms. The unsupervised algorithms mainly involve the techniques including region growing [15]–[17], hierarchical clustering [18], spectral clustering [19], [20], iterative clustering [21] and boundary segmentation [22], [23].

The boundary segmentation algorithm is developed based on the Hoffman's minima rule[24]. It can generate the accurate boundary in line with the semantic boundary selection principle of human vision. The problem of the boundary 3D shape segmentation algorithm is that it is very sensitive to noise and non-uniform points. On the other hand, the algorithms regarding region growing, hierarchical clustering, spectral clustering and iterative clustering do not take Hoffman's minima rule into account, and often lead to the under-segmentation of the boundary. However, the region growing technique has the advantage of being less sensitive to noise and non-uniform points, which can be used to compensate the disadvantage of the boundary segmentation algorithm.

Therefore, in this paper a semantic-driven hybrid segmentation scheme for 3D shapes based on point cloud is proposed. In particular, a bottom-up hierarchical segmentation method is proposed. The features of the Local Concave-Convex Histogram (LCCH) of the 3D point cloud are extracted, and the boundary region is then clustered based on the LCCH. Also, other feature regions are clustered and the unallocated points are clustered by the *KNN* algorithm to achieve semantic clustering. The similarity among the semantic clustering regions is calculated using the variational method and the similar regions are merged. The regions are then recombined using the improved Shape Diameter Function (SDF).

The rest of this paper is organized as follows: In Section 2, the related work is reviewed. In Section 3, a semantic-driven hybrid segmentation scheme for 3D shapes is presented in detail. The experimental results are presented in Section 4 to demonstrate the effectiveness and efficiency of the proposed algorithm. The limitations of the proposed approach are discussed in Section 5. Section 6 draws the conclusions of this paper.

II. RELATED WORK

The general purpose of 3D shape segmentation is to subdivide a given 3D shape into meaningful parts according to the features of the concavity and convexity as well as the similarity. The concavity and convexity are the main geometric features used by the boundary-based 3D shape segmentation algorithms to subdivide a 3D shape into different regions. On the other hand, the similarity is used as the means to measure the feature similarity between different regions of a geometric object. It can be used to build up the regions with the similar features, and then the region-based 3D shape segmentation algorithms can be performed. Performing hierarchical region clustering by jointly considering the boundary features is very useful for merging similar regions.

A. REGION-BASED CLUSTERING ALGORITHM

The region growing is the most popular region clustering algorithm, which works by first selecting a point of point cloud as a seed, and then letting it grow until the defined stop condition is satisfied. This process is then repeated by selecting a new seed point until the entire point cloud is divided into different regions. It should be noted that the region growing algorithm always stops at the boundary where the curvature changes significantly, thus, it may lead to the problem of the over-segmentation when it is applied to, for instance, an ellipsoidal shape.

Chazelle *et al.* [25] proposed a flooding-heuristics algorithm for decomposing a polyhedral surface into convex regions. The basic idea of this algorithm was established on the use of the dual graph of the given 3D mesh: starting from a node and traversing the graph and collecting facets along the way when the convex conditions were satisfied. The problem of this algorithm was that it could produce some fragmented regions.

The algorithm proposed by Zhang *et al.* [26] used a low-pass filter to denoise the mesh to remove some outlier points, and estimated the curvature of each vertex. Then, the region growing method was adopted, starting from the vertex with the largest curvature value, for segmenting the regions according to the similarity of the curvature of the vertices. After the vertices were grouped, the minimum cut algorithm was used at the last stage to adjust the boundary of the regions. However, whether the segmented results meet the minima rule had not been considered.

Xiao *et al.* [27] proposed a mesh segmentation algorithm for CAD models. A clustering algorithm was used in this method to divide the regions into two types: the dense regions and the sparse regions, where the sparse regions included large, elongated triangles, while the dense regions included small, relatively regular triangles. Then the sparse triangles were decomposed into sub-regions of planar surfaces, cylinders and cones by using Gaussian map and the random Hough transform, while the dense regions were segmented by the mean shift operation in the mean curvature field. The segmentation result obtained by this algorithm was patch-based, which was not suitable for processing the semantic segmentation of 3D shapes.

In Wang's method [28], the planar segmentation was achieved by using the region growing algorithm based on the angle between the normal vectors, the quadric surface segmentation was achieved based on the Gaussian mapping, and the free-form surface segmentation by using the bicubic B-spline. It was then followed by a process of feature reconstruction, together with a Boolean operation for combining the fitted surfaces to form the final model. One issue for this algorithm was that some details might be lost during the process.

Fan *et al.* [29] proposed an improved region growing algorithm. It automatically selected the seed point based on the features of the extracted neighborhood density and used the improved region growing algorithm to segment the points. The method could be applied to the point clouds of both outdoor and indoor scenes, as well as a single object. But this algorithm was not sensitive to the perception of the boundaries of the objects with the similar surface structures, and might not always create a satisfactory segmentation result.

The segmentation results obtained with the above-mentioned region growing algorithms do not in general conform to the semantic segmentation, as it often leads to the loss of the detailed information such as boundaries. To make segmentation results generated from the region growing method consistent with the semantic segmentation rules from the perspective of the cognitive science theory, the boundary feature that conforms to the semantic relation needs to be considered.

B. VARIATIONAL METHOD

The variational method is a branch of the mathematical analysis concerned with the variation of the functional, as opposed to the ordinary calculus which is concerned with the variation of the variable. It is developed initially to solve some

extreme value problems in the practical problems by looking for the maximum value or the minimum value of the functionals. In general, the points of a 3D shape surface could be divided into the planar region and various concave and convex regions, such as the umbilical concavity, umbilical convexity, hyperbolic concavity, hyperbolic convexity, hyperbolic saddle, parabolic concavity, parabolic convexity, ellipsoidal concavity and ellipsoidal convexity [30], [31]. Analyzing the feature of the surface types of 3D shape, by computing the error extreme values between adjacent regions of the same surface types using variational method, regions with the few errors can be merged together.

Cohen-Steiner *et al.* [32] proposed a distortion-minimizing flooding algorithm based on the normal vector of the meshes. It used the error-driven energy minimization to reduce the approximate error. But this algorithm might sometime produce too many fragments of the planar polygons when it was used to fit the planar surface. To solve the problem, high-order or special types of surfaces were introduced to expand the segmentation of the fitted primitives. For example, Simari and Singh [33] took the ellipsoidal surface as the only primitive type in their solution. By minimizing a combined energy function to approximate a given proxy, the segmented boundary was smoothed by the constraint condition relaxation of the boundary points. In [34], Attente *et al.* treated each triangle as a cluster, and merged neighboring triangles according to the minimization of the fitting function error. The primitives considered in this algorithm were planar surface, cylinder and sphere. The disadvantage was that the cost function needed to be recalculated every time the merging operation was performed. Kobbelt [35] introduced the sphere, cylinder and more complex rolling-ball to the flooding algorithm, which improved the effect of 3D shape segmentation. However, due to the limitation of the category of the primitives, it still showed some disadvantages when it was used to CAD and free-form surface models. Yan *et al.* [36] applied a variational segmentation framework to segment the tree point cloud obtained from a laser-scanning device into cylindrical parts, from which tree branches were then constructed. Yan *et al.* [21] proposed a 3D shape segmentation method by variational quadric approximation. Each part from the segmentation was replaced by a common quadric surface, which was a natural extension of the planar surface. The new energy functions L^2 and $L^{2,1}$ were introduced to measure the error between the surface and its proxy. To achieve the best segmentation effect, the Lloyd iteration was performed repeatedly between the mesh segmentation and the fitted quadric surface to minimize the energy function. However, the disadvantages were that the method was more sensitive to the noise and the surface types had to be fixed. By extending the Mumford-Shah model, Zhang *et al.* [37] proposed a technique that could decompose a mesh into a prescribed number of the segments by minimizing the internal variation of the part and the length of the boundaries between the parts. The disadvantage was that the 3D shape (such as the fish) lacking

obvious geometric boundaries needed higher-order eigenvectors of the Laplacian matrix to complete the segmentation.

Due to the fact that the types of the surfaces considered in above variational methods are very limited, they often lead to the problem of the over-segmentation, which significantly limits their applications to the segmentation of the CAD model and the free-form surface model.

C. SHAPE DIAMETER FUNCTION

The Shape Diameter Function (SDF) is derived from the medial axis transformation (MAT) [38], [39]. While MAT represents geometric information using the distance from the point p on the surface to the medial axis, SDF characterizes geometric features using the penetration distance of the point p along the internal direction of the normal vector. Gal and Shapira *et al.* [40], [41] proposed a method using SDF for calculating the local thickness for mesh surfaces. More specifically, it calculates the distance from a point on the mesh to the opposed region. SDF could be thought as the thickness of the bounding volume of a mesh. As a 3D geometric feature, SDF can be seen as the diameter of the shape for each point on the surface corresponding to the region where it is located. So it could be used to distinguish the regions with the different width or thickness. Its initial application was to segment a 3D shape based on the similarity of the SDF values. When a 3D shape is processed and transformed using translation, rotation, simplification, etc., the 3D shape always tends to maintain the original shape features. To calculate the SDF values for a 3D shape, a cone is specified for each point p on the surface of the shape with the cone vertex being at p and the cone axis is aligned with the surface normal at p inside the shape. Then, several rays within the cone range are generated from the cone vertex. Only the rays which have an angle less than 90° with the normal of the intersection are kept, and the rays that may intersect the mesh in a wrong manner are discarded. Finally, a weighted sum of all the lengths of the rays is calculated to obtain the SDF value of point p .

Therefore, when SDF is used as the feature of a 3D shape, the SDF will not be greatly affected by the posture change to the shape. Thus, SDF has very good robustness in this case. However, the calculation of the SDF is in general very expensive. Martin Huska and Morigi [42] proposed a method for calculating the SDF value of the point cloud on a closed two-dimensional manifold, where the point cloud was regarded as a movable particle flow, the particle state was judged by a simple collision test, and the SDF of the colliding particles was defined as the distance covered by mutual movement. This method extended the scalar function defined on the mesh to the point cloud, but the offset direction was greatly affected by the accuracy of the normal vector estimation. Chen *et al.* [43] proposed an SDF calculation method based on the offset surface, which used the space spheres to realize offset surface of the point cloud. It simplified the process of searching the penetrating rays by arranging the space spheres through OBB tree. This algorithm significantly

improved the computational efficiency of the SDF, but the offset distance needed to be determined by empirical values, which could not be adaptively selected, and some detailed information could be lost.

D. THE VOXELIZATION METHOD

The point cloud for a single 3D shape is usually transformed from 3D CAD model by the sampling method on each mesh, or obtained from multiple scans of real-world object by using the traditional methods such as SfM [44] or other methods extended from KinectFusion [45] or ElasticFusion [46], etc. The 3D points from both cases hardly follow the grid type and are irregularly arranged. Therefore, BA and OBA are not suitable for the direct application in this paper. The supervoxel method may be able to overcome the irregularity problem in the point cloud. This subsection reviews the voxelization methods in the literature.

References [5]–[8] studied the voxelization methods with different resolutions. In [5], Wu *et. al* adopted a voxel size of $30 \times 30 \times 30$ to represent a geometric 3D shape as a probability distribution of binary variables on a 3D voxel grid using a Convolutional Deep Belief Network. Their method naturally supports object recognition and shape completion from 2.5D depth maps. However, it needs a training dataset such as ModelNet [5], a large-scale 3D CAD model dataset. In [6], Wu *et. al* used another voxel size of $64 \times 64 \times 64$ for 3D object generation based on the 3D Generative Adversarial Network (GAN) from a probabilistic space by leveraging recent advances in volumetric convolutional networks and GAN. The generative objects have fine details and variations. However, again this method needs the training data (ModelNet). In [7], Yu and Lee i) performed the voxelization on the single-view depth data, ii) approximated the true observation model of 3D objects to the tractable distributions, iii) formulated the true generative model for 3D object with the Bayesian networks, and iv) performed the classification through the maximum likelihood estimation and shape retrieval. The method in [7] is based on the training dataset ModelNet10 and ModelNet40. In [8], Akai *et. al* adopted 1m as a voxel unit to construct a normal distribution map offline and combined it with a road-marker matching approach using a particle-filtering algorithm to localize the autonomous vehicles in the environment based on point cloud matching. It also requires an offline dataset in the scene.

According to the discussions above, although references [5]–[8] have improved the detailed data through the difference in voxel resolution, it requires either the large dataset (ModelNet) [5]–[7] to conduct the training, or the offline dataset in the scene [8] to perform mapping. However, in this paper we do not have a large dataset as the data source to perform the analysis, but only a single 3D shape to be segmented.

References [9] and [10] optimize the voxelization method to segment the objects in the scene. Lin *et. al* developed a heuristic algorithm in [9] that utilized local information on the basis of the traditional K voxels to optimize the voxelization

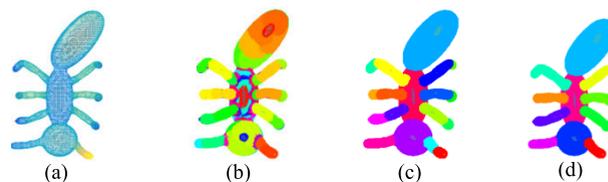


FIGURE 1. The framework of the segmentation algorithm in this paper: (a) Input: the 3D shape to be segmented, (b) Semantic Clustering: the result of the semantic clustering of the 3D shape based on the features of the LCCH, (c) Variational Merging: the result achieved by merging the over-segmented regions of the semantic clustering result according to the variational method, and (d) Final Result: the final segmentation result of the 3D shape by merging the approximate regions according to the improved SDF.

and efficiently solved the subset selection problem. However, the method is only suitable for the scenarios where the subset boundaries in the scene have the obvious difference in normal vector. In [10], Landrieu and Boussaha adopted a supervised method, combined the graph structure comparison with a cross-region weighting strategy to generate a high contrast at the boundaries. The segmentation result is significantly better than the unsupervised method. But this method requires a large number of labeled datasets to conduct supervised training in order to achieve better segmentation of objects in the scene.

In summary, the voxelization methods proposed by the literature improve the resolution of the voxels. However, the methods proposed in these references either require a large dataset to conduct the training in order to conduct effective segmentation of an object, or are for different purposes, i.e., segmenting objects from the scene, which manifests different boundary feature (i.e., normal vector) from segmenting a single object (i.e., use saddle characteristics as the boundary feature).

III. METHODOLOGY

Figure 1 illustrates the main steps of our bottom-up hierarchical segmentation algorithm. Our method uses the feature of the Local Concave-Convex Histogram (LCCH). Based on the feature, the boundary regions of the given shape are first extracted. A semantic clustering result can then be obtained by applying the region clustering algorithm to the 3D shape corresponding to the planar surface, convex surface and weak concave surface (Fig. 1b). The adjacent regions are then merged based on their similarity calculated by applying the variational method (Fig. 1c). Finally, the segmentation regions are merged again at a higher level (Fig. 1d) according to the improved SDF values of the regions, which are then further optimized using the graph-cut algorithm [47].

A. SEMANTIC CLUSTERING

1) DEFINITION OF THE LCCH

Most 3D shape segmentation algorithms are developed based on certain geometric features [48], which vary from the selected features and the adopted strategies. Different algorithms may have different computational efficiency and robustness, and produce different segmentation results.

The geometric features used in a segmentation algorithm can be generally divided into local features and global features: the former describes the features associated with the local regions of an object, such as the curvature, concavity and convexity, and SDF; while the latter characterizes the global geometric features of an object, such as Viewpoint Feature Histogram and Clustered Viewpoint Feature Histogram. The global features can fully reflect the structural information of a geometric shape, but their calculation can be very costly in terms of computing and memory resources. In contrast, local features can accurately reflect the locally specific features of an object, and their calculations are relatively simple and therefore less expensive. One issue associated with local features is that they are likely to be affected by local geometric details or noises.

In the cognitive science, it is widely acknowledged that the meaningful decomposition of an object is the main index used by the human perception system in the shape recognition process [49]. The minima rule [24], [50] proposed by Hoffmann and Richards has been commonly accepted as an important theory in the field of human visual perception. Based on this theory, human visual perception system interprets a region along the minimum value of the negative principal curvature as a boundary that conforms to semantic features. Therefore, the concave-convex features are crucial for the boundary segmentation of a 3D shape. Generally, the boundary features are determined by the angles between normal vectors. However, there can be many neighboring points around a point and consequently, there can be many such angles. It will be error-prone inevitably if only one angle is used to determine whether it belongs to a boundary or not.

The segmentation of a 3D shape can be carried out according to the visual cognition principle of the minima rule. Generally, a visually consistent region has a consistent concavity, convexity or curvature. That is, the surface of the 3D shape can be defined as a collection of the planar, concave and convex regions. Also, all those concave regions can be considered as the boundary of 3D shape. In other words, from the perspective of the cognitive science, the concave features are more important than the convex features for the 3D shape segmentation. Therefore, the performance of the algorithm could be effectively improved by accurately identifying uniform concavity regions. In order to quickly identify the local concave-convex features, a concept of the LCCH is introduced in this paper, based on which the segmentation complying with the minima rule is then achieved.

To segment a 3D point cloud with our method, the LCCH feature needs to be built. First, for each point in the point cloud, its k neighboring points are calculated. It then calculates a distance vector from the point to each of its neighboring points, as well as two included angles formed between the distance vector and the normal vectors at the two endpoints. The concave-convex relation is then determined according to the calculated angles.

Suppose p_1 is a point on the surface of a point cloud with the position vector \vec{c}_1 and the normal vector \vec{n}_1 . Let p_2 be

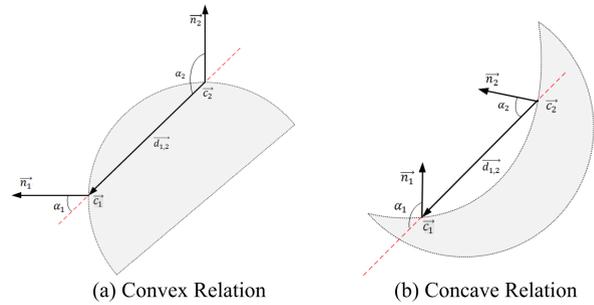


FIGURE 2. Judgment on the convexity and concavity between the points.

a neighboring point of p_1 with the position vector \vec{c}_2 and the normal vector \vec{n}_2 . The unit connection distance vector from the neighboring point p_2 to the point p_1 is defined as $\vec{d}_{1,2} = (\vec{c}_1 - \vec{c}_2) / |\vec{c}_1 - \vec{c}_2|$. It can be seen that the cosine of the included angle α_1 between the normal vector \vec{n}_1 and the unit connection distance vector $\vec{d}_{1,2}$ can be directly obtained from the dot product $\vec{n}_1 \cdot \vec{d}_{1,2}$. Similarly, from $\vec{n}_2 \cdot \vec{d}_{1,2}$, the cosine of the included angle α_2 formed between the normal vector \vec{n}_2 and the unit connection distance vector $\vec{d}_{1,2}$ can be calculated.

As shown in Fig. 2a, when $\alpha_1 < \alpha_2$, the connection line of the two points on the surface of the point cloud passes through the interior of the 3D shape, and the two points are convexly connected. Since the included angle is between 0 degree and 180 degree, the following formula can be derived to determine the convexity between the points.

$$\begin{aligned} \alpha_1 < \alpha_2 &\Leftrightarrow \cos \alpha_1 - \cos \alpha_2 > 0 \\ &\Leftrightarrow \vec{n}_1 \cdot \vec{d}_{1,2} - \vec{n}_2 \cdot \vec{d}_{1,2} > 0 \\ &\Leftrightarrow (\vec{n}_1 - \vec{n}_2) \cdot \vec{d}_{1,2} > 0 \end{aligned} \quad (1)$$

Equation (1) shows that two points are convexly connected if the dot product $(\vec{n}_1 - \vec{n}_2) \cdot \vec{d}_{1,2}$ is positive.

Similarly, as shown in Fig. 2b when $\alpha_1 > \alpha_2$, the connection line of the two points on the surface of the point cloud is outside of the 3D shape, and the two points are concavely connected, which can be represented in the following formula.

$$\alpha_1 > \alpha_2 \Leftrightarrow \cos \alpha_1 - \cos \alpha_2 < 0 \Leftrightarrow (\vec{n}_1 - \vec{n}_2) \cdot \vec{d}_{1,2} < 0 \quad (2)$$

If two points are in a planar region, the included angles of the connection line and the two normal vectors are both equal to 90 degrees.

For a point p_i on the surface of the given shape, the concave-convex relation between p_i and its neighboring point p_j can be represented by c_{ij} . A vector $cv_i = \{c_{i1}, c_{i2}, \dots, c_{in}\}$ is defined for p_i to represent its concave-convex relations with its n neighboring points. In order to fully describe the point, local concave-convex histogram statistics on this vector can be calculated, which provides the percentage of the planar relations, the weak convex relations, the convex relations, the weak concave relations and the concave relations in the n elements of the vector cv_i .

Bearing in mind that there may be the noises in the point cloud in reality, we have to take into account the possible

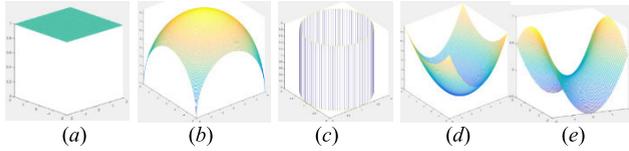


FIGURE 3. Typical surfaces structure in a 3D shape: (a) Planar surface (b) Ellipsoidal surface (c) Cylindrical surface (d) Concave surface (e) Saddle surface.

deviations when calculating the angles. Let $\beta = (\vec{n}_1 - \vec{n}_2) \cdot \vec{d}_{1,2}$, and β_{th} be the threshold of the difference of the cosine in the concave-convex relation between the two points. To avoid merging the regions one of which is the weak concave and the other is the strong concave, the threshold vector $\beta_{th} = \{\beta_1, \beta_2, \beta_3\}$ is set, which is used to determine the concave-convex relation of two points as follows:

- If $|\beta| < \beta_1$, the two points are of a planar relation;
- If $\beta > \beta_1$ and $\beta < \beta_2$, the two points are of a weak convex relation;
- If $\beta > \beta_2$, the two points are of a strong convex relation;
- If $\beta < -\beta_1$ and $\beta > \beta_3$, the two points are of a weak concave relation;
- If $\beta < \beta_3$, the two points are of a strong concave relation.

2) SEMANTIC CLUSTERING BASED ON LCCH

In order to ensure that the concave boundary regions in the 3D shape are not over-merged, our method first clusters the boundary regions, and then the flat region, convex region, weak concave region and regional competition.

Region clustering of classified point clouds requires the LCCH analysis for different types of surfaces to determine the threshold. The surfaces in reality can be comprised by five basic surfaces: planar, ellipsoidal, cylindrical, concave and saddle surfaces. Therefore, we conduct the LCCH analysis for these five surfaces.

These five surfaces can be modelled by (3)-(7), respectively, where $x = -n1 + n2 \cdot rand(p_n, 1)$, $y = -n1 + n2 \cdot rand(p_n, 1)$, R is the radius for the ellipsoid or cylinder, where p_n is the number of the randomly generated points, $n1$ and $n2$ are two natural numbers. Note that for the cylindrical surface, similar to the variable x described above, z is randomly generated. Figure 3 illustrates local surfaces of these five surfaces as the meshes. It randomly selects N feature points from the given surface, and then calculates the local concave-convex vectors described above for these N points.

$$z = 1 \tag{3}$$

$$z = (R^2 - p \cdot x^2 - q \cdot y^2)^{\frac{1}{2}} \tag{4}$$

$$y = 1 - (q \cdot (R^2 - (x - 1)^2 / p))^{\frac{1}{2}} \tag{5}$$

$$z = \frac{x^2}{p} + \frac{y^2}{q} \tag{6}$$

$$z = \frac{x^2}{p} - \frac{y^2}{q} \tag{7}$$

The steps of the semantic clustering based on LCCH are described as follows.



FIGURE 4. Concave feature points of the cup.

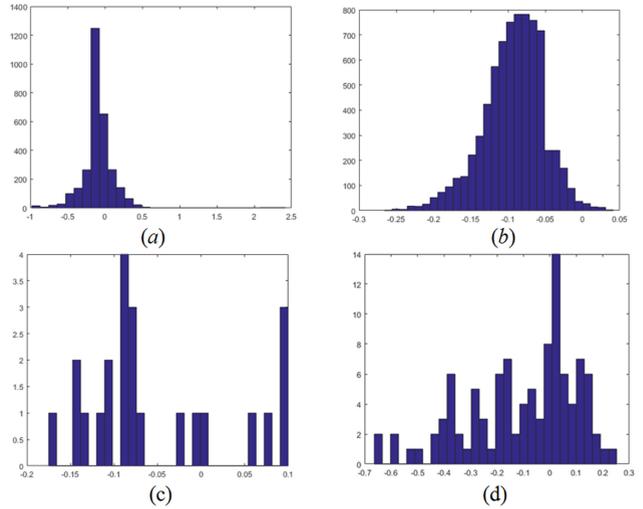


FIGURE 5. Histograms of the LCCVs of the boundary points and non-boundary points of the 3D cup model and the saddle surface: (a) the histogram of the LCCVs of the feature points with the boundary features from the cup model; (b) the histogram of the LCCVs of the feature points of the non-boundary features from the cup model, (c) the histogram of the LCCV of a particular feature point on the saddle surface, and (d) the histogram of the LCCVs of the feature point set on the saddle surface.

a: BOUNDARY CLUSTERING

In order to obtain the boundary region in the process of the 3D shape segmentation, the regions with the concave feature are first extracted (the points with the concave feature are identified based on the LCCH features introduced in this paper.). The concave region is then clustered by keeping the points which are consistent with the regional concave feature and by eliminating the isolated points.

There may also be some concave regions which are not associated with any boundary, but instead are some non-boundary inner regions. For instance, the interior of a cup is a typical concave, but it is not a boundary region (see Fig. 4). It is observed that the boundary region which has concave features is generally located among multiple convex regions or planar regions, so that between the feature points of the boundary region and their neighborhoods there are multiple concave-convex features.

In order to eliminate such concave non-boundary points, the characteristics of the neighborhood of the feature points need to be analyzed. Figure 5a plots the Local Concave-Convex Vectors (LCCVs) statistics of the feature points for the boundary, Fig. 5b is the LCCVs statistics of non-boundary points, Fig. 5c is the LCCV statistics of a particular feature point on the saddle surface, and Fig. 5d is



FIGURE 6. Feature points on the surface of a cup which are concave and planar with their neighborhoods.

the LCCVs statistics of the feature point set on the saddle surface. It can be seen from Fig. 5 that the LCCV statistics of the feature points as the segmentation boundary conforms to the histogram trend of the feature points on the saddle surface, whereas that of the neighborhood is of the varying features. Some with the convex feature are greater than 0, and some with the concave features smaller than 0, while others with the planar features close to 0. The non-boundary region only includes the concave features which are less than 0 and the planar features which are close to 0. Therefore, the concave non-boundary points can be identified and eliminated based on the distribution of the concave-convex features among the point and its neighborhood. The points used for identifying the concave and planar features from the LCCV are plotted in Fig. 6. From the comparison of Fig. 4 and Fig. 6 and the results of the statistical analysis in the form of the histogram shown in Fig. 5, it can be seen that the threshold obtained by analyzing the statistical results in this paper is valid.

From Fig. 5c and 5d, it can be seen that the feature points on the saddle surface include the concavity, convexity and planarity. It shows clearly that the histogram of the LCCVs of the boundary points conform to the histogram of the LCCVs on the saddle surface. So we can apply the histogram of the LCCVs on the saddle surface to identify the boundary points to cluster the semantic boundary.

The clustering algorithm for the boundary is shown in Algorithm 1.

Algorithm 1:

1. Extracting the feature points of the boundary region as the seed points and pushing them into the seed point queue;
2. *While* the seed point queue is not empty
 - a. Popping out a seed point;
 - b. Clustering the point as a boundary region;
 - c. Setting this point as a candidate feature point;
 - d. *While* the candidate feature points are not empty;
 - i. Taking the neighboring points with the saddle feature of the candidate points;
 - ii. Clustering the saddle neighboring points according to connectivity;
 - iii. Using the saddle neighboring points as candidate points;
 - e. end
 - f. Popping out a new point from the seed point queue as a seed point, starting a clustering of the new boundary region;
3. end
4. Removing the points with fewer connected regions.

b: PLANAR CLUSTERING

The fundamental idea of this technique is to find such a point that it matches the highest point of the histogram of the planar relation from those that meet the plane attribute threshold based on the LCCH, and then set it as the seed point. The planar region clustering process is then expanded from its 1-ring neighborhood. These regions are generally the planar surface regions of the models such as the tabletop, the chair surface, and the like.

c: CONVEX CLUSTERING

This algorithm first finds out those points that conform to the convex feature in the unallocated point cloud, it is then refined to find the first highest point of the histogram in line with the convexity condition and set it as the first seed point. The convex clustering continues along the 1-ring neighborhood. The convex regions are mainly quadric surfaces, such as the cylinder or the cylinder-like regions (part of a cylinder), and spherical surface regions.

d: WEAK CONCAVE CLUSTERING

Similar to the process of the boundary clustering, weak concave clustering is to identify those regions that are mainly located in the areas where different types of the regions are connected with the planar surfaces.

e: REGIONAL COMPETITION

Due to the thresholds set in the above four clustering processes, it may generate some isolated points or the points that are not completely connected. Therefore, there may exist some points on the surface of a 3D shape which are not within any of the above threshold ranges and thus have not been assigned to any clusters after the above clustering processes. Since many neighboring points of these unallocated points have already been classified into a certain cluster, the *KNN* algorithm can be used to assign these points to the proper clusters. This can be done by calculating the most likely cluster around the point. Once an unallocated point has been assigned to a cluster, the state of the point is updated as the neighborhood. The process iterates until all unallocated points have been allocated.

For the 3D shape with the strong convex and concave consistency and low noise, the above method is able to achieve excellent results fast. However, the processing is at the expense of the possible over-segmentation for the large blocks of the ellipsoids. In order to address the over-segmentation problem and generate more meaningful and accurate segmentation regions, a new method is developed to merge similar regions again, which is presented next.

B. VARIATIONAL MERGING

It can be seen from the semantic clustering results in Fig. 1b that the surface of the point cloud can be divided into many convex regions, concave regions or planar regions, and that the convex regions similar to the ellipsoid are

clearly over-segmented. To obtain the convex regions conforming to the semantics, it is necessary to measure the similarity between adjacent regions and merge these regions. In this paper, a variational method based on the narrow-band theory is proposed to perform another round of merging to combine the small convex regions generated by the semantic clustering.

The traditional variational method [21] starts from finding a proxy of a triangle on the mesh surface, and then searches the mesh surface closest to the proxy for merging. The search is conducted globally and may require an iterative process to obtain the final results. The computational cost of this method is very high. So the merging optimization process is necessary to improve the efficiency. In this paper, the variational method is developed to merge the regions to obtain larger regions. These larger regions are easily used to merge regions for similar SDF.

There are basically two methods for constructing a surface from a discrete point cloud: surface interpolation and surface fitting. Interpolating a noisy point cloud is not only very computationally expensive, but more importantly practically inappropriate. In fact, it is impossible to accurately reconstruct the surface represented by a point cloud through surface interpolation when the point cloud data are relatively noisy. On the other hand, while the surface obtained from a fitting algorithm may not interpolate all the points, it can represent the geometry represented by a point cloud more faithfully in general than that obtained from an interpolation method. In addition, due to its approximate nature, surface fitting can filter the noise in the point cloud effectively. According to the work presented in [51], most of the artificial objects are composed of quadric surfaces and planar surfaces, where quadric surfaces typically consist of ellipsoidal surface, cylindrical surface, concave surface and saddle surface. Since the planar surface can be considered as a special quadric surface, quadric fitting is sufficient for our task. Therefore, the quadric surface fitting method is presented in this paper.

A quadric surface can be represented implicitly as the level set of a function $f(X): f(X) = 0$, where $X = \{x, y, z\}^T \in \mathbb{R}^3$ is the coordinate of a point. More specifically, $f(X)$ can be written as:

$$\begin{aligned} f(X) &= A^T \cdot F \\ &= a_0x^2 + a_1y^2 + a_2z^2 + a_3xy + a_4xz \\ &\quad + a_5yz + a_6x + a_7y + a_8z + a_9 \end{aligned} \quad (8)$$

where A is the column vector of the unknown coefficient of the quadric implicit representation, and F is a vector defined as $F = \{x^2, y^2, z^2, xy, xz, yz, x, y, z, 1\}^T$.

To improve the efficiency of the algorithm, we draw on the narrow-band idea proposed in the reference[52] for the region selection. Based on this idea, the adjacent regions are selected according to where the surface proxy is located. Then the errors of the fitted surface proxy to the corresponding region and its neighbors are calculated to reduce the amount of the error calculation. More specifically, a surface proxy is

fitted first to each region, and then the error of each region to the fitted surface proxy and the errors from the surface proxy to its neighboring regions are calculated. Secondly, a global priority queue Qu is constructed according to the fitting error of the region and the corresponding surface proxy and the errors from the neighboring regions to the surface proxy. Finally, the *MacQueen* algorithm is applied to determine the pair of adjacent regions (R_i, R_j) which have the smallest error in the global queue. Then the two selected regions (R_i, R_j) are merged as a new region using the distortion-minimizing flooding algorithm [32] if the error of the two selected regions is less than the given threshold. This new region is then fitted with a surface proxy; the errors of the proxy to the new region and its adjacent regions are calculated and the global priority queue Qu is updated. This process is recursively applied to update the merging of other regions until there is not a region pair in Qu that has an error less than the threshold.

The formula for calculating the error between the region and the surface proxy is shown in (9), where R_i is the point cloud region after the semantic clustering, Q_i is the surface proxy fitted from a region R_i of point cloud, $d(R_i^k, Q_i)$ is the error between point R_i^k on the region R_i and the surface proxy Q_i , and $E(R_i, Q_i)$ is the error between a point cloud region R_i and the surface proxy Q_i .

$$E(R_i, Q_i) = \frac{1}{ni} \sum_{k=1}^{ni} d(R_i^k, Q_i) \quad (9)$$

According to the principle of least squares, $d(R_i^k, Q_i)$ is defined as the mean square error from a point to its surface proxy. Since the collection of the regions from the semantic clustering process is obtained according to the LCCBs, the directional difference between the points of the planar region to this region is 0 approximately and there must be a certain amount of the deviation between the normal directions of each point in the quadric surface and the normal direction of the fitted surface. Therefore, equation (9) only uses the error in terms of the distance difference. It is unnecessary to introduce the error in terms of the direction difference.

C. REGION MERGING BASED ON THE IMPROVED SDF

It can be seen from Fig.1 that after variational merging there still exist some over-segmentation regions that appear to have similar shapes. To merge these regions further, the order relation based on the traditional SDF computation is introduced to improve the computation of the SDF, and the histogram of the SDFs of the regions is computed. With the proposed method, the *Earth Mover's Distance* (EMD) [53] is used to calculate the similarity of the histograms between two adjacent regions. When the similarity is greater than the threshold Th , the adjacent regions are merged. This process repeats iteratively until the condition is not satisfied.

1) ORDER RELATION

The order relation analysis method [54] is a mathematical method for weight distribution. Firstly, the evaluation indexes

TABLE 1. Reference table for γ_k value assignment.

γ_k	Description
1.0	The index X_k is as important as the index X_{k-1}
1.2	The index X_k is slightly more important than the index X_{k-1}
1.4	The index X_k is obviously more important than the index X_{k-1}
1.6	The index X_k is strongly more important than the index X_{k-1}
1.8	The index X_k is extremely more important than the index X_{k-1}

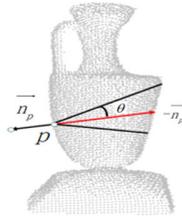


FIGURE 7. Schematic diagram of the shape diameter.

are ranked according to their importance, and the corresponding weight value of each evaluation index is calculated quantitatively. The following definitions are introduced in order to help to present the order relation analysis.

Definition 1: Let X_i and X_j be two evaluation indexes, and the expression $X_i \preceq X_j$ indicates that the importance of the index X_i is less than or equal to X_j ;

Definition 2: If for a certain evaluation criterion, there exists a relation “ \preceq ” such that $X_1^* \preceq X_2^* \preceq \dots \preceq X_m^*$ among a set of evaluation indexes X_1, X_2, \dots, X_m (X_i^* indicates the i -th evaluation index ($i = 1, 2, \dots, m$) when $\{X_i\}$ is sorted by the relation “ \preceq ”), then, it is said that this group of evaluation indexes X_1, X_2, \dots, X_m has an order relation “ \preceq ”.

For any pair of adjacent evaluation indexes X_{k-1} and X_k , the ratio of importance between them is expressed as ω_k/ω_{k-1} , shown in (10).

$$\gamma_k = \omega_k/\omega_{k-1} \quad (k = m, m - 1, \dots, 2) \quad (10)$$

The reference value for the value of γ_k is shown in Table 1.

First, γ_2 is selected to calculate ω_2 by (11) and then, (10) is used to iteratively calculate other weights.

$$\omega_m = (1 + \sum_{k=2}^m \prod_{i=k}^m \gamma_i)^{-1} \quad (k = m, m - 1, \dots, 2) \quad (11)$$

2) IMPROVED SDF

Let M be point cloud, p be an arbitrary point on its surface, q be the intersection of point p on the opposite surface in line with the internal normal direction, and $d(p, q)$ be the SDF for the point p . As shown in Fig.7, to calculate the SDF, a cone with an apex angle in line with the internal normal as the center line at the point p is constructed. The point p is used as the starting point to find several line segments intersecting

the surface of the shape inside the cone. These line segments must satisfy the condition that their two endpoints are visible to each other. When searching for the penetrating line segment, the number of the selected penetrating line segments is reduced in this paper by selecting self region or the multi-ring neighboring regions. The weight value of each line segment is the cosine of the included angle between the internal normal vector at the point p and the line segment. The computation of the traditional SDF at the point p is set as the average penetration distance of all line segments according to (12).

$$SDF_p = \frac{1}{m} \sum_{q=1}^m \cos \theta_{pq} d_{pq}, \quad (12)$$

where,

$$\cos \theta_{pq} = \frac{-\vec{n}_p \cdot \vec{q} - p}{|-\vec{n}_p| |q - p|},$$

$$d_{pq} = \sqrt{\sum_{k=1}^3 (p o_{pk} - p o_{qk})^2}.$$

According to the SDF calculated above, it can be seen that when the included angles between the internal normal vector and all the effective penetrating line segments in the cone become larger, the distances of the penetrating line segments projected on the internal normal direction deviate more from its actual SDF value for non-cylindrical shapes. Also, the smaller the included angle, the smaller the gap. For the like-cylindrical shapes, when the direction of the central axis forms a certain angle with the distance vector that is composed of the intersection points of the internal normal vector and the effective penetrating line segments on the surface, it is the same as the situation of non-cylindrical shapes. Otherwise the gap is very small or even close to 0. Therefore, if the cosine of the included angle and the mean value are used, the error of calculating the SDF value of the point will be retained.

Our solution to this problem is to introduce the order relation analysis method to sort the cosines of the included angles between the penetrating rays and the internal normal vector in the order of their importance. It assigns the corresponding weight value to each effective penetrating line segment according to the importance. This makes the weight value smaller for the penetration line segment with the larger included angle. Otherwise, the smaller included angle the larger weight. So the combination of these weights for the order relation and the cosines of the angles reduces the error of SDF calculation.

3) REGION MERGING BASED ON THE IMPROVED SDF

As the similarity of the adjacent regions can be found directly from the improved SDF, regions can be merged according to the similarity. Specifically, the merging is conducted as follows:

1. Generating the region set $R = \{R_1, R_2, \dots, R_m\}$ by the variational method;

2. Sampling a point p from the region R_i ; taking p as the vertex, specifying a cone with an apex angle of $\frac{2\pi}{18}$ and the cone axis along the internal normal direction $-\vec{n}_p$, and then creating a set of penetrating line segments inside the cone;
3. Forming a point set L from the intersection points on the surface inside the cone that are visible from point p , so that the line segment between point p and any point q from the point set L is a valid segment;
4. For each point q in the point set L , calculating the cosine of the included angle between the line segment pq and the internal normal vector at the point p , and sorting all the values in the ascending order;
5. Calculating the weight value corresponding to each penetration distance in the ascending sequence by using the (10) and (11) with $\gamma_2 = 1.2$ (refer to Table 1);
6. Calculating the SDF value at point p by computing a weighted sum on the penetration distances of all the penetrating line segments starting from the point p according to (13), where ω_{pq} is the unit weight value corresponding to the penetration line segment pq , θ_{pq} is the included angle between the internal normal vector and the line segment pq , and d_{pq} is the Euclidean distance from the point p to the point q ;

$$SDF_p^* = \sum_{q=1}^m \omega_{pq} \cdot \cos \theta_{pq} \cdot d_{pq}, \quad (13)$$

7. Calculating the SDF values of all sampling points iteratively on the region R_i ;
8. Normalizing the SDF value at point p using (14), where $SDF_{i_{min}}^*$ is the minimal SDF value on the region R_i , $SDF_{i_{max}}^*$ is the maximal SDF value, $SDF_{i_p}^*$ is the original SDF value at point p on the region R_i , $SDF_{i_p}^\#$ is the normalized SDF value at point p , and constructing a histogram h_i of the SDF values of the region R_i ;

$$SDF_{i_p}^\# = \frac{SDF_{i_p}^* - SDF_{i_{min}}^*}{SDF_{i_{max}}^* - SDF_{i_{min}}^*} \quad (14)$$

9. Similarly, constructing the histograms for other regions;
10. Selecting two adjacent regions R_i and R_j ; letting the histogram of the SDF for the region R_j be h_j ; calculating the similarity of the histograms in terms of the *Earth Mover's Distance* by (15). If the similarity is greater than the threshold Th , merging the adjacent regions R_i and R_j ;

$$\text{dist}(R_i, R_j) = \text{EMD}(h_i, h_j) \quad (15)$$

11. Go back to Step 10. to merge other similar adjacent regions until the condition is not met, namely, the similarity is not greater than the threshold Th .

D. ANALYSIS OF THE TIME COMPLEXITY

As presented in previous subsections, our algorithm processes the following stages in sequence: semantic clustering, variational merging, and region remerging. Suppose the number of points in a point cloud is pn . The time complexity of our method is analyzed as follows.

The semantic clustering stage firstly calculates the neighbors of each point, the time complexity of which is $O(pn^2)$. Next it calculates the LCCHs between each of all points and its neighbors, the time complexity of which is $O(pn \cdot \text{neigh})$, where neigh is the number of neighbors of a point. Multiple region-growing processes are performed according to the LCCHs, the time complexity of which is $O(pr \cdot \text{neigh})$, where pr is the number of the points in some particular regions ($pr < pn$). So the time complexity of the semantic clustering stage is $O(pn^2 + pn \cdot \text{neigh} + pr \cdot \text{neigh}) = O(pn^2)$.

The variational merging stage includes the process of calculating the errors between the regions and their neighbors and the process of regional merging. Let r_num be the number of the regions, which is typically much less than pn . The time complexity of calculating the errors for all regions is $O(r_num^2)$, and the time complexity of the regional merging is $O(r_num)$. So the time complexity of the traditional method is $O(r_num^2 + r_num) = O(r_num^2)$. In this paper, in order to reduce the number of error calculations, we use the regions as the unit of a neighbor in the error calculation method. The time complexity of the improved method is $O(r_num \cdot r_neigh)$, where r_neigh is the number of the neighboring regions. Because r_neigh is usually less than half of r_num , the running time of the improved method is reduced by half compared with that of the traditional method. The time complexity of this stage is $O(r_num \cdot r_neigh + r_num) = O(r_num \cdot r_neigh)$.

The region remerging introduces the SDF. In order to improve the accuracy of the SDF calculation, we introduce the order relation, which does not change the time complexity of this stage. The time complexity of this stage is determined by that of the SDF calculation, which is of $O(r_num2 \cdot p_pa \cdot p_neigh)$ and that of the merging and updating of the regions, which is $O(r_num2 \cdot r_neigh2)$. r_num2 is the number of the regions ($\leq r_num2r_num$), p_pa is the mean of the sampled points in every region, p_neigh is the number of the points in the region for the intersection ($p_neigh < pn$), and r_neigh2 is the number of the neighboring regions, which has the same order of magnitude as r_neigh . Since $r_num2 \cdot p_pa$ is less than pn in the average case, $O(r_num2 \cdot p_pa \cdot p_neigh) < O(pn^2)$. In order to optimize the boundaries of the segmentation we introduce the graph-cut method. Its time complexity is $O(e \cdot pn^2 \cdot |C|)$ [47], where e is the number of the edges of the graph, $|C|$ is the number of minimum cut set. Although the time complexity of the graph-cut is relatively high, we found that it can achieve better optimization results. So the time complexity of this stage is $O(r_num2 \cdot p_pa \cdot p_neigh + r_num2 \cdot r_neigh2 + e \cdot pn^2 \cdot |C|) = O(e \cdot pn^2 \cdot |C|)$.

Therefore, the time complexity of our algorithm is $(pn^2 + r_num \cdot r_neigh + e \cdot pn^2 \cdot |C|) = O(e \cdot pn^2 \cdot |C|)$.

IV. EXPERIMENTS

The hierarchical semantic-driven point cloud segmentation algorithm proposed in this paper can be used in a wide range of the applications, such as the accurate collision detection, shape editing, shape deformation, shape reconstruction.

The proposed method is evaluated and the experimental results are analyzed in this section.

A. EXPERIMENT AND RESULTS

The experimental results of our algorithm are presented in this section. The algorithm proposed in this paper takes the point clouds as input. These point clouds are usually obtained from the 3D CAD models through the sampling method on each mesh, or from scanning the real-world objects by using the traditional methods such as SfM [44] or other methods extended from KinectFusion [45] or ElasticFusion [46], etc.

The smoothness of the point cloud sampled from the 3D CAD model is determined by the grid surfaces of the CAD model. There are generally few noisy points. The KinectFusion technique [45] uses the following steps to scan a real object and generate the point cloud. First, the multi-view scanning method is used to scan the depth images from different views. Second, the filter method is typically performed to reduce the noise, and compute the coordinates and normal vectors of the points. After this step, the noise is significantly reduced compared with that in the raw data after the first step. Finally, the current camera position is calculated according to the point clouds of the current frame and the previous frame to fuse two frames of depth data until the complete data representation of a 3D shape is obtained. The point cloud obtained by scanning real object typically has a finer data representation than a point cloud by sampling a 3D CAD model, and consequently the surface details (such as engraved patterns on the object surface) of the real-world object may be captured by the point cloud. So the point cloud generated from scanning a real object demonstrates more convex and concave features on the surface due to the finer data representation.

All experiments were carried out on a computer with an Intel Xeon E5-2620 2.10 GHz CPU and 16 GB of memory. The 3D shapes of the Princeton object segmentation dataset [55] were used for the experimental testing. The dataset consisted of 380 objects in 19 types and included the results of various 3D shape segmentation algorithms. To use the mesh data, the points were sampled randomly with uniform density to obtain the points from the mesh shapes. The normal vector at each sampled vertex was calculated using the technique of the fitting from its neighboring points. Our algorithm works well on a 3D shape with the obvious concave-convex boundary features. The experimental results achieved with our algorithm on some triangle mesh objects are shown in Fig 8. Our method can also be applied to the point cloud datasets directly acquired from the real world 3D objects. The experimental results are shown in Fig 9.

B. ANALYSIS OF THE EXPERIMENTAL RESULTS

The method proposed in this paper is an unsupervised algorithm. The users do not need to set the different parameters for different shapes, and these built-in parameters obtained by the statistical analysis can segment a given 3D shape automatically, which is the strength of our technique. There are



FIGURE 8. The segmentation results of the Princeton dataset, all results are automatically generated by our algorithm. It can be seen that the segmentation boundaries have saddle geometry feature.



FIGURE 9. The segmentation results generated by our algorithm for some 3D point cloud datasets acquired from the real world environment.

different evaluation indexes for 3D shape segmentation, including *Cut Discrepancy*, *Consistency Error*, *Hamming Distance*, and *Rand Index* [55], [56].

1) CUT DISCREPANCY

Cut Discrepancy is a boundary-based method for measuring the similarity between two segmentation methods. It is based on the summation of the distances between the points along the boundary of the segmentation result under the evaluation and the boundary of the closest cuts of the benchmark segmentation.

2) CONSISTENCY ERROR

Consistency Error can determine the similarity and differences of the two segmentation results. By considering different hierarchical granularities, the quality of the segmentation can be assessed based on the Global Consistency Error(GCE) and Local Consistency Error(LCE).

3) HAMMING DISTANCE

Hamming Distance measures the discrepancy between the result of a segmentation algorithm and the benchmark segmentation result by finding the difference between their best segmentation. If a segmentation result is the benchmark segmentation result, such measurement will result in a missing rate or a false alarm rate, then Hamming Distance will be defined as the average of the two values.

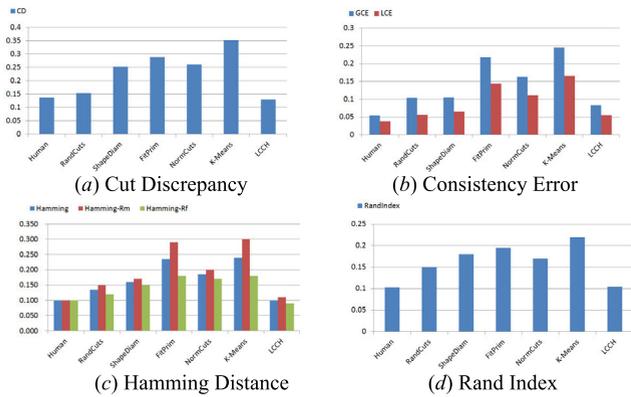


FIGURE 10. Comparison of the segmentation algorithms with four evaluation metrics.

4) RAND INDEX

Rand Index assesses the similarity between the results from the proposed segmentation technique and the benchmark segmentation by considering whether a pair of faces is in the same region or in two different regions for both segmentations. It is defined as the ratio of the number of pairs that are either in the same region or in the different regions for both segmentations to the total number of pairs.

With these four metrics, the segmentation results of the algorithm can be evaluated to examine whether they match the semantic results of the artificial segmentation. When the segmentation results of the algorithm are close to the benchmark segmentation results in terms of these four metrics, it indicates that the segmentation result of the algorithm is close to how human brains partition an object’s surface into meaningful parts. The closer the results are, the more effective the algorithm is.

To evaluate the effectiveness of the method proposed in this paper, the semantic-driven point cloud segmentation method is compared with a number of classical algorithms in terms of the above four metrics. These algorithms include Human [55], Rand Cuts [57], SDF [41], FitPrim [34], Norm-Cuts [55], K-means [55], SB19 [58], SB6 [58], M-S [37], SSF [19], and Heterogeneous [20], where SB19 and SB6 are supervised segmentation algorithms. The quantification histograms for the evaluation results in terms of the four evaluation metrics are shown in Fig. 10. Figures 10a, 10b, 10c, and 10d show the comparison results in terms of Cut Discrepancy, Consistency Error, Hamming Distance, and Rand Index, respectively. As can be seen from Fig. 10, the overall performance of the algorithm proposed in this paper is better. Our algorithm outperforms the unsupervised segmentation algorithms in Fig. 10, and is close to the benchmark results.

Table 2 shows the detailed comparison between our algorithm and the state-of-the-art algorithms in terms of Rand Index (RI). According to the results in Table 2, the algorithm proposed in this paper works well in most shape categories, especially for the 3D shapes with obvious convex features, such as ant, fish, table, octopus, teddy, glasses and vase. As can be seen from Table 2, our algorithm can achieve a

TABLE 2. The comparison of the segmentation results in terms of RI obtained by our algorithm and the existing algorithms.

Algorithms	Human	Rand Cuts	SDF	SB19	SB6	M-S	SSF	Heterogeneous	LCCH
Human	13.5	15.8	17.9	11.9	14.3	11.1	12.8	12.4	10.7
Cup	13.6	22.4	35.8	9.9	10	20.4	14.6	11	9.5
Glasses	10.1	9.7	20.4	13.6	14.1	9.4	11.3	9.9	9.1
Airplane	9.2	11.5	9.2	7.9	8	11.1	13.2	11.2	8.2
Ant	3	2.5	2.2	1.9	2.3	2.2	2.8	2	1.4
Chair	8.9	18.9	11.1	5.4	6.1	10.9	8.4	7.4	8.9
Octopus	2.4	6.7	4.5	1.8	2.2	2.5	2.6	2.6	2.3
Table	9.3	37.4	18.4	6.2	6.4	10.3	6.1	7	2.4
Teddy	4.9	4.5	5.7	3.1	5.3	3.2	3.6	3.9	4.2
Hand	9.1	9.7	20.2	10.4	13.9	7.9	11	10.7	10.5
Plier	7.1	10.9	37.5	5.4	10	8.9	8.5	5.7	8.0
Fish	15.5	29.7	24.8	12.9	14.2	29.6	21.5	18.6	8.7
Bird	6.2	11.4	11.5	10.4	14.8	9.4	7.8	7.8	12.8
Armadillo	8.3	8.1	9	9	8.4	8.7	9.1	10.3	14.4
Bust	22	25.1	29.8	21.4	33.4	25.1	28.6	25.8	26.6
Mech	13.1	28.3	23.8	10	12.7	13.1	12.6	10.5	18.0
Bearing	10.4	12.9	11.9	9.7	21.7	16.6	14.8	9.5	10.1
Vase	14.4	16	23.9	16	19.9	12.5	15.4	12.1	10.3
Fourleg	14.9	17.7	16.1	13.3	14.7	14.4	16.5	15.7	23.3
Average	10.3	15.7	17.6	9.4	12.2	12	11.6	10.2	10.5

meaningful segmentation closer to that by human, which is especially the case for ant, table and octopus. Our algorithm is also better than other unsupervised algorithms. It can achieve the same or better performance for shapes such as fish than Heterogeneous graphs [20].

We have also compared our method with the supervised mesh segmentation method. We note that when the training set of each classification exceeds 90%, the learning-based method can achieve a better accuracy for most types (see Table 2, in the column for SB19). However, when the training set is reduced to 30%, our method outperforms it (see SB6 in Table 2). In practice, it is difficult to collect a large number of the real-world 3D shapes. In many situations, there may be only a small number (even only one) of 3D shape datasets available. Due to this reason, the learning-based methods have certain limitations in their applications. In contrast, our algorithm does not need to use a large number of 3D shapes. Therefore, it is more suitable and applicable for many applications.

V. DISCUSSION

Based on the features of the LCCH, a semantic-driven 3D shape segmentation algorithm is proposed in this paper. This algorithm constructs the boundary regions that follow the semantic features, and clusters other types of regions based on preserving boundary regions. To reduce the over-segmented regions, a variational method based on the narrow-band idea is introduced to merge similar regions. A re-merging process is further applied using the improved SDF to create the final segmentation.

A. UNDER-SEGMENTATION OF THE SEMANTIC CLUSTERING

The algorithm proposed in this paper first classifies and merges similar regions according to the features of the LCCH.

TABLE 3. Mean and variance of local concave and convex features of the surfaces formed by different parameters according to (3)-(6).

		Plane	Ellipsoid	Cylindrical surface	Concave surface
$p=1$ $q=1$	Mean	0.0001	0.0655	0.1072	-0.3212
	Variance	3.58e-39	5.05e-4	0.0035	0.022
$p=1$ $q=2$	Mean	0.0001	0.0948	0.1125	-0.437
	Variance	5.47e-39	0.0016	0.0035	0.0021
$p=1$ $q=3$	Mean	0.0001	0.1119	0.1421	-0.503
	Variance	5.58e-39	0.0032	0.0064	0.0289
$p=1$ $q=4$	Mean	0.0001	0.1215	0.1239	-0.5319
	Variance	5.48e-39	0.0066	0.0051	0.0129
$p=1$ $q=5$	Mean	0.0001	0.1321	0.1208	-0.5292
	Variance	3.71e-39	0.0099	0.0048	0.0145
$p=2$ $q=3$	Mean	0.0001	0.1335	0.0816	-0.4033
	Variance	5.65e-39	0.0053	0.002	0.015
$p=2$ $q=5$	Mean	0.0001	0.149	0.0927	-0.4375
	Variance	3.79e-39	0.0099	0.0032	0.0119
$p=2$ $q=7$	Mean	0.0001	0.1481	0.1085	-0.4119
	Variance	5.48e-39	0.0092	0.0038	0.0096

When conducting the semantic region clustering in the concave region, multiple boundary regions are plotted out one region according to the connectivity, which in general cannot directly produce semantically correct segmentation. To improve the segmentation, the connected adjacent regions generated with the LCCH are analyzed. When there are multiple neighboring regions with similar *SDFs*, to avoid the under-segmentation of the subsequent segmentation results, we determine which region a point should be clustered according to the distances from the centroids of the similar regions.

In addition, our 3D shape segmentation method is fully automatic, although it still requires to specify some thresholds statistically, which can be done offline. These thresholds are required when performing semantic clustering operations relating to point feature classification and clustering the points into regions. Analyzing the effect to the surface features from the surface parameters, the following parameters are selected for typical feature analysis. Table 3 shows the mean and variance of the local concave-convex features for typical feature regions corresponding to the planar surface, the ellipsoid, the cylindrical surface, and the concave surface. The numerical values of the concave and convex features of the local neighborhood on the planar surface should be close to 0, and the numerical value of the features on the ellipsoid should all be greater than 0. We know from (5) that if the connection direction between the point on cylindrical surface and its neighbor is parallel to the z-axis, its feature value is close to 0; if not, the feature value is greater than 0. The numerical value of the features on the concave surface may be all less than or equal to 0. According to the analysis on the results in Table 3, the absolute value of the difference of the planar angle cosine should be less than 0.05 and to obtain the obvious concavity or convexity the thresholds is set as $\{-0.15, 0.15\}$. Therefore, the threshold β_{th} is set to be $\{0.05, 0.15, -0.15\}$, by which it can achieve the reasonable point feature classification result.

TABLE 4. The proportion of the feature points on the certain saddle surfaces with the different parameters and type thresholds.

Values of parameters p and q in (7)	Type thresholds	
	0.05	0.15
$p=1$ $q=5$	0.6667	0.8571
$p=1$ $q=4$	0.7098	0.8535
$p=1$ $q=3$	0.4722	0.625
$p=1$ $q=2$	0.6364	0.7273
$p=1$ $q=1$	0.5625	0.5833
$p=2$ $q=3$	0.6818	0.9091
$p=2$ $q=5$	0.8182	0.9091
$p=2$ $q=7$	0.7591	1
$p=5$ $q=1$	0.4765	0.5556
$p=4$ $q=1$	0.4516	0.6724
$p=3$ $q=1$	0.3709	0.641
$p=2$ $q=1$	0.5714	0.75

As for the saddle surface, it can be seen from Fig. 5 that the LCCH features formed by the feature points and their neighborhood points with the saddle features should be a mixture of the concave, convex or planar features. That is, some feature values are greater than 0, and others are less than or equal to 0. Thus, it is not helpful to calculate the mean or variance of the feature values of the saddle surface. Therefore, relevant statistics for the saddle surface features are not calculated in Table 3. However, the regions with this kind of features are typically located at the boundary, which will be used as an important feature for the saddle surface analysis.

Table 4 shows the proportion of the feature points on saddle surfaces with the different parameters, where the type threshold is to set the range of concave relations of the feature points. It can be seen from Table 4 that the appropriate proportion can be obtained which abides by the rule that type threshold is smaller than 0.05. Although the maximum of the type threshold used in Table 4 is 0.15, the proportion of the extracted feature points is higher. More non-boundary feature points may be introduced in this calculation process, which can increase the workload of eliminating the non-boundary feature points in the subsequent stages. We randomly generate some new saddle surfaces, set the type threshold to 0.05, and obtain new results similar to Table 4. The clustering threshold value of 0.48 is obtained by the average value of all data about the feature point proportions subtracting the standard deviation. The clustering threshold of 0.48 divides the feature points from the neighbors. Our experimental results show that by setting the clustering threshold, we can achieve better extraction of saddle feature points and also better clustering effect. This is also why 0.05 is used in this paper for selecting adequate feature points on the saddle surface. When clustering the saddle regions with the clustering threshold being set to 0.48, the feature points that are more than the clustering threshold are clustered. Similarly, it can be seen that, when determining the growth of the saddle, the planar relation between the point and the neighbor should exist and the proportion values be small. The planar relation proportion is deemed to be smaller than 0.15. Therefore, the threshold

of clustering the saddle feature points is set as $\{0.48, 0.15\}$, and the surface containing such feature points should have weak convex and convex attributes. The competition allocation in the first stage also requires to be performed for the unallocated points. In order to avoid the increasing workload of eliminating the wrong type of points, more demanding conditions can be imposed when extracting certain types of the feature points.

B. OVER-SEGMENTATION OF THE VARIATIONAL MERGING

In the process of region merging based on the variational method, it is necessary to calculate the error between the proxy and the corresponding region, and the errors between this proxy and the corresponding neighboring regions. If a fixed threshold is used for these different shapes, it may not be possible to get a satisfactory segmentation; if the value is set too big, some regions may be over-merged; if it is too small, some shapes may be under-merged. If the error distribution between the regions is not uniform, the threshold may not be suitable for the error calculation of those regions. An adaptive threshold method is introduced in our algorithm, which can eliminate some of the obvious errors according to the error values between the regions. The remaining errors are sorted to find out a proper threshold. Through the statistical analysis, it is found that it is most suitable to set the threshold as the value which is at the lowest 20% of all non-zero error values. Therefore, it is essential to optimize the threshold obtained above.

C. SEGMENTING EFFICIENCY FOR LARGE-SCALE SHAPES

In the experiments presented in Section IV, the numbers of points in the point clouds of some 3D shapes are less than a few thousands, and some are more than 50,000, which demonstrate different scales, and result in different running time ranging from tens of seconds to about 70 minutes. We have analyzed the time complexity of our method systematically in Section III.D. In this section, we examine the efficiency for segmenting the shape with the largest scale in the Princeton object segmentation dataset, which are the armadillo shapes with the scale of about 50,000 points.

Figure 11a shows the segmentation results of five models of the armadillo shape. As can be seen from Fig. 11a, our method can generate the finer-grained semantic segmentation results for the shape. For example, in the first column of Fig. 11a different semantic parts of the left leg, such as thigh, shin and calf, are effectively captured by our method and segmented into different regions.

The segmenting time for these five models is about 70 mins. The segmenting time for the large-scale shapes is relatively long, which can be reduced by reducing the shape scale, i.e., the number of points in the shape. We introduce a simplification method to reduce the shape scale, in which a shape is discretized into a grid of voxels, and only the point closest to the center of a voxel is used to represent all points in the voxel. Then the ratio of the difference between the number

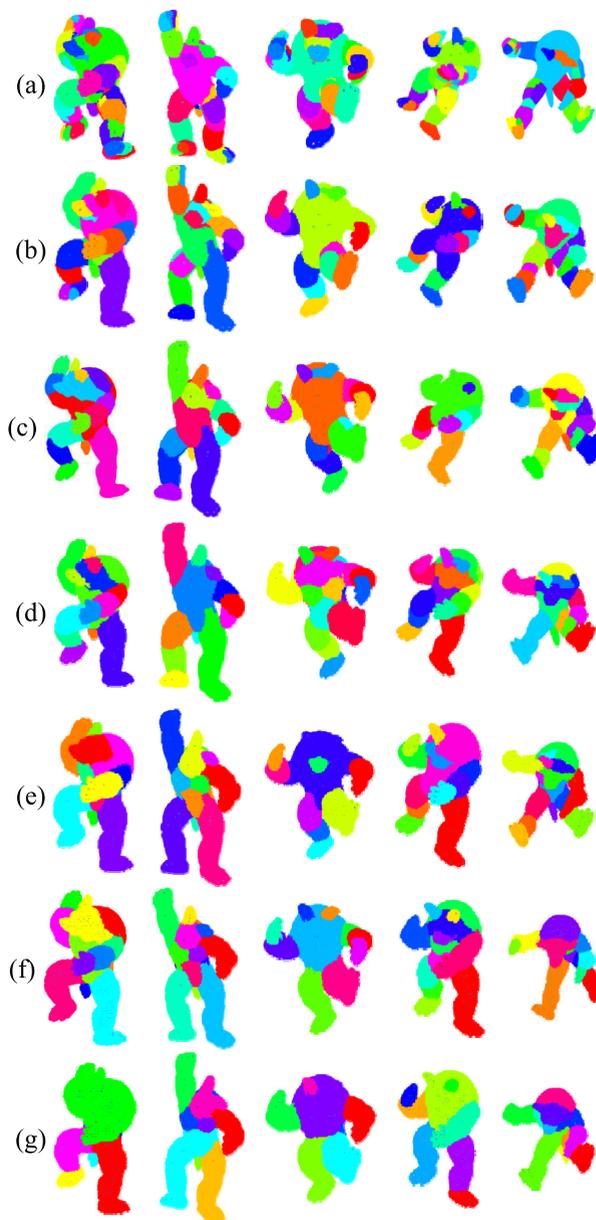


FIGURE 11. Segmentation results of the large-scale shapes. (a) The original point cloud. (b) 50% simplification rate. (c) 60% simplification rate. (d) 70% simplification rate. (e) 80% simplification rate. (f) 85% simplification rate. (g) 90% simplification rate.

of points in the original point cloud and the total number of voxels to the number of points in the original point cloud is called the simplification rate.

We further evaluate the segmentation effectiveness and the efficiency with different simplification rates. Figures 11b-11g show the segmentation results with the simplification rates from 50% to 90%. When the simplification rate is 50% (i.e. the number of points is about 25000), our method still generates good segmentation results (Fig. 11b), although fine-grained segmentation is missing in some shapes. For example, in the first column of Fig. 11b, the left leg of the shape is only segmented as a single region rather than into different regions as shown in Fig. 11a. This is

TABLE 5. Clustering results for the sampled point clouds in terms of different thresholds.

Classification Thresholds	Thresholds of clustering the saddle feature points	Clustering results of saddle feature points				
{0.02,0.10,-0.10}	{0.30,0.15}					
	{0.48,0.15}					
	{0.80,0.15}					
{0.05,0.15,-0.15}	{0.30,0.15}					
	{0.48,0.15}					
	{0.80,0.15}					
{0.15,0.30,-0.30}	{0.30,0.15}					
	{0.48,0.15}					
	{0.80,0.15}					

because when we simplify the original point cloud, many joints of the armadillo shape form the boundaries only among the major parts, causing our method to segment a major part of the shape as a single region.

However, although the fine-grained semantic segmentation is missing, the segmenting time is reduced significantly from ~70 mins to ~6 mins when the simplification rate is 50%. As we further increase the simplification rate to 90% (Fig. 11c to 11g), the segmenting time further decreases to 30-40 seconds for these five models, but the segmentation results become less effective. We found that when the simplification rate is over 60%, the segmentation attains too few parts and the under-segmentation occurs. This is because when the simplification rate is too high, the boundary details among the parts are lost.

D. LIMITATIONS

There are some limitations in our method. Firstly, for the 3D point cloud with the strong consistency of the concave



FIGURE 12. Over-segmentation results.

and convex features and little noise, the algorithm proposed in this paper obtains excellent segmentation results; but for the shapes with the rich set of the concave and convex regions and high level of noise, it can be shown from Fig. 12, the face, hat and body from the bust consist of some patches. The issue of over-segmentation will occur, which is inevitable. If the parameter values set in our algorithm for local concave-convex feature is tight, it will lead to excessive over-segmentation.

On the other hand, if the parameter values are set too loose, it will cause the boundary region to be too large, leading to

TABLE 6. Clustering results for the scanned point clouds in terms of different thresholds.

Classification Thresholds	Thresholds of clustering the saddle feature points	Clustering results of saddle feature points			
{0.02,0.10,-0.10}	{0.30,0.15}				
	{0.48,0.15}				
	{0.80,0.15}				
{0.05,0.15,-0.15}	{0.30,0.15}				
	{0.48,0.15}				
	{0.80,0.15}				
{0.15,0.30,-0.30}	{0.30,0.15}				
	{0.48,0.15}				
	{0.80,0.15}				

the situation where the boundary is far away from the region with the most prominent saddle feature, and consequently the merging workload will increase. As shown in Table 5, when the tight classification threshold of the saddle feature is used, such as -0.3 , or the threshold of the clustering feature points is selected as 0.8 , fewer feature points or feature regions are obtained. Instead the classification threshold is selected as -0.1 or the clustering threshold is selected as 0.3 , more feature points will be obtained, some of which are redundant, increasing the workload in the later processing stages.

We also carried out the experiments with the point clouds generated from scanning real objects under varying parameter values (similar to Table 5). The results (the clustering of the feature points) are shown in Table 6. Similarly, when the classification threshold or clustering threshold is set tight, a fewer number of feature points are obtained. In contrast, more feature points are obtained as the threshold setting becomes looser.

In addition, because the point cloud generated from scanning a real object has a finer data representation, the surface details of the real-world object may be captured by the point cloud, and consequently the point cloud generated from scanning demonstrates more convex and concave features on the surface. In Table 6, the 3D shape in the second column has more surface details. It can be seen from this table that more feature points are extracted for this 3D shape when the thresholds are $\{0.05, 0.15, -0.15\}$ and $\{0.48, 0.15\}$, which can be used to differentiate valid boundary regions in later stages.

In this paper, the fixed threshold is determined according to the parameters of various shapes. However, for the 3D shape with more concave-convex features (such as bust), the segmentation results may not be satisfactory. In the future work, we first plan to develop an adaptive approach to setting parameter, which is able to automate the 3D shape segmentation, and achieve even better segmentation result.

Secondly, the proposed algorithm can still be sensitive around the saddle region. In this paper, in order to avoid the problem of the under-segmentation of the saddle region in the process of the semantic clustering, the first step in our method is to extract the region with the saddle features. As can be seen from our experimental results, the extracted boundary is not accurate enough. To address this issue, it is required to set the refined threshold parameters for the saddle surface. However, it is a very challenging task to determine a set of parameters suitable for all 3D shapes. In order to obtain the boundary region more accurately and suitable for multiple shapes, we plan to carry out the multi-granularity segmentation on the boundary region in our future work, which can reduce the workload of the subsequent region clustering and improve the efficiency of 3D shape segmentation.

Finally, in the experiments for the large-scale shapes, we evaluated the segmentation effectiveness and efficiency when the large-scale shapes are simplified, for which we introduce an intuitive simplification method. This prompts us a new research direction towards developing a more

sophisticated simplification method. In the future, we plan to incorporate the difference in boundary feature into the simplification method, aiming to reduce the loss of the boundary details.

VI. CONCLUSION

An efficient semantic-driven 3D point cloud hybrid segmentation algorithm is proposed in this paper. Firstly, the normal vectors and the LCCH features of the 3D point cloud are calculated. Based on the LCCH, the boundary regions which meet the semantic-driven requirements are obtained. This is accompanied with several other type-specific region clustering operations to achieve the semantic clustering of the 3D point cloud. Since the saddle regions with the boundary features are first extracted for region clustering, the under-segmentation of the saddle regions is avoided, which overcomes the shortcomings encountered for some commonly used region clustering operations, as they do not in general consider the semantic properties of the boundary region. The effective boundary regions can be obtained by the semantic clustering based on the LCCH for the 3D shape which are not overly simplified. However, the issue of the over-segmentation may occur in the process of semantic clustering for some regions such as ellipsoids. The variational method based on the narrow-band theory is introduced in our algorithm to merge the adjacent regions with the approximation of the regions. Finally, a re-merging process is applied based on the improved SDF values by using the order relation to combine similar adjacent regions, which makes the segmentation result more consistent with the semantic features. The experimental results from the Princeton Dataset show that the algorithm proposed in this paper outperforms the existing classical automatic segmentation algorithms in terms of the four metrics. The algorithm proposed in this paper can also be applied to the point cloud data acquired directly from real-world objects with digital scanning devices, which shows that our algorithm has a good universality.

In the future, we intend to further improve our method from the following three aspects: 1) introducing an adaptive parameters setting scheme by additional consideration of theoretical and mathematical aspects of the algorithm, 2) carrying out the multi-granularity segmentation on the boundary region, and 3) developing a simplification method that takes the boundary features into account. We also plan to extend our method to other 3D applications by considering local concave-convex histogram features.

ACKNOWLEDGMENT

The models and segmentation benchmark used in this paper are courtesy of the Princeton Benchmark. The authors are grateful to the authors who have provided their experimental results. The authors would also like to thank the reviewers for their detailed comments and suggestions which greatly improved the integrity of the manuscript.

REFERENCES

- [1] R. S. V. Rodrigues, J. F. M. Morgado, and A. J. P. Gomes, "Part-based mesh segmentation: A survey," *Comput. Graph. Forum*, vol. 37, no. 6, pp. 235–274, Feb. 2018.
- [2] Y. Ben-Shabat, T. Avraham, M. Lindenbaum, and A. Fischer, "Graph based over-segmentation methods for 3D point clouds," *Comput. Vis. Image Understand.*, vol. 174, pp. 12–23, Sep. 2018.
- [3] M. Wang, M. Ju, Y. Fan, S. Guo, M. Liao, H. Yang, D. He, and T. Komura, "3D incomplete point cloud surfaces reconstruction with segmentation and feature-enhancement," *IEEE Access*, vol. 7, pp. 15272–15281, 2019.
- [4] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 77–85.
- [5] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1912–1920.
- [6] J. Wu, "Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling," in *Proc. Conf. NIPS*, Barcelona, Spain, 2016, pp. 1–9.
- [7] H. W. Yu and B. H. Lee, "A variational feature encoding method of 3D object for probabilistic semantic SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Madrid, Spain, Oct. 2018, pp. 1–8.
- [8] N. Akai, L. Y. Morales, E. Takeuchi, Y. Yoshihara, and Y. Ninomiya, "Robust localization using 3D NDT scan matching with experimentally determined uncertainty and road marker matching," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Redondo Beach, CA, USA, Jun. 2017, pp. 1356–1363.
- [9] Y. Lin, C. Wang, D. Zhai, W. Li, and J. Li, "Toward better boundary preserved supervoxel segmentation for 3D point clouds," *ISPRS J. Photogram. Remote Sens.*, vol. 143, no. 2018, pp. 39–47, Sep. 2018.
- [10] L. Landrieu and M. Boussaha, "Point cloud oversegmentation with graph-structured deep metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Los Angeles, CA, USA, Jun. 2019, pp. 1–15.
- [11] Y. Zhuang, G. He, H. Hu, and Z. Wu, "A novel outdoor scene-understanding framework for unmanned ground vehicles with 3D laser scanners," *Trans. Inst. Meas. Control*, vol. 37, no. 4, pp. 435–445, Jul. 2014.
- [12] Y. Zhuang, X. Lin, H. Hu, and G. Guo, "Using scale coordination and semantic information for robust 3-D object recognition by a service robot," *IEEE Sensors J.*, vol. 15, no. 1, pp. 37–47, Jan. 2015.
- [13] X. Zhang, Y. Zhuang, H. Hu, and W. Wang, "3-D laser-based multiclass and multiview object detection in cluttered indoor scenes," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 1, pp. 177–190, Jan. 2017.
- [14] Y. Zhuang, Y. Liu, G. He, and W. Wang, "Contextual classification of 3D laser points with conditional random fields in urban environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Hamburg, Germany, Sep. 2015, pp. 3908–3913.
- [15] A. F. Koschan, "Perception-based 3D triangle mesh segmentation using fast marching watersheds," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Madison, WI, USA, vol. 2, Jun. 2003, pp. 27–32.
- [16] G. Lavoué, F. Dupont, and A. Baskurt, "A new CAD mesh segmentation method, based on curvature tensor analysis," *Comput.-Aided Des.*, vol. 37, no. 10, pp. 975–987, Sep. 2005.
- [17] U. Imdad, M. Asif, M. Ahmad, O. Sohaib, M. Hanif, and M. Chaudary, "Three dimensional point cloud compression and decompression using polynomials of degree one," *Symmetry*, vol. 11, no. 2, p. 209, Feb. 2019.
- [18] Y. Xu, W. Yao, S. Tattas, L. Hoegner, and U. Stilla, "Unsupervised segmentation of point clouds from buildings using hierarchical clustering based on gestalt principles," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 11, pp. 4270–4286, Nov. 2018.
- [19] H. Wang, T. Lu, O. K.-C. Au, and C.-L. Tai, "Spectral 3D mesh segmentation with a novel single segmentation field," *Graph. Models*, vol. 76, no. 5, pp. 440–456, Sep. 2014.
- [20] P. Theologou, I. Pratikakis, and T. Theoharis, "Unsupervised spectral mesh segmentation driven by heterogeneous graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 2, pp. 397–410, Feb. 2017.
- [21] D.-M. Yan, W. Wang, Y. Liu, and Z. Yang, "Variational mesh segmentation via quadric surface fitting," *Comput.-Aided Des.*, vol. 44, no. 11, pp. 1072–1082, Nov. 2012.
- [22] Y. Zheng and C.-L. Tai, "Mesh decomposition with cross-boundary brushes," *Comput. Graph. Forum*, vol. 29, no. 2, pp. 527–535, Jun. 2010.
- [23] O. Kin-Chung Au, Y. Zheng, M. Chen, P. Xu, and C.-L. Tai, "Mesh segmentation with concavity-aware fields," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 7, pp. 1125–1134, Jul. 2012.
- [24] D. D. Hoffman and W. A. Richards, "Parts of recognition," *Cognition*, vol. 18, nos. 1–3, pp. 65–96, 1984.
- [25] B. Chazelle, "Strategies for polyhedral surface decomposition," *Comput. Geom.*, vol. 7, nos. 5–6, pp. 327–342, 1997.
- [26] X. Zhang, "3D mesh segmentation using mean-shifted curvature," in *Proc. Int. Conf. Advan. Geo. Modeling Process.*, Hangzhou, China, 2008, pp. 465–474.
- [27] D. Xiao, H. Lin, C. Xian, and S. Gao, "CAD mesh model segmentation by clustering," *Comput. Graph.*, vol. 35, no. 3, pp. 685–691, Jun. 2011.
- [28] J. Wang, D. Gu, Z. Yu, C. Tan, and L. Zhou, "A framework for 3D model reconstruction in reverse engineering," *Comput. Ind. Eng.*, vol. 63, no. 4, pp. 1189–1200, Dec. 2012.
- [29] Y. Fan, M. Wang, N. Geng, D. He, J. Chang, and J. J. Zhang, "A self-adaptive segmentation method for a point cloud," *Vis. Comput.*, vol. 34, no. 5, pp. 659–673, 2018.
- [30] L. Di Angelo and P. D. Stefano, "Geometric segmentation of 3D scanned surfaces," *Comput.-Aided Des.*, vol. 62, pp. 44–56, May 2015.
- [31] M. Gu, L. Duan, M. Wang, Y. Bai, H. Shao, H. Wang, and F. Liu, "An improved approach of mesh segmentation to extract feature regions," *PLoS ONE*, vol. 10, no. 10, Oct. 2015, Art. no. e0139488.
- [32] D. Cohen-Steiner, P. Alliez, and M. Desbrun, "Variational shape approximation," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 905–914, Aug. 2004.
- [33] P. D. Simari and K. Singh, "Extraction and remeshing of ellipsoidal representations from mesh data," in *Proc. Graph Interf.*, Victoria, BC, Canada, 2005, pp. 161–168.
- [34] M. Attene, B. Falcidieno, and M. Spagnuolo, "Hierarchical mesh segmentation based on fitting primitives," *Vis. Comput.*, vol. 22, no. 3, pp. 181–193, Feb. 2006.
- [35] J. W. L. Kobbelt, "Structure recovery via hybrid variational surface approximation," *Comput. Graph. Forum*, vol. 24, no. 3, pp. 277–284, Oct. 2005.
- [36] D.-M. Yan, J. Wintz, B. Mourrain, W. Wang, F. Boudon, and C. Godin, "Efficient and robust reconstruction of botanical branching structure from laser scanned points," in *Proc. 11th IEEE Int. Conf. Comput.-Aided Des. Comput. Graph.*, Huangshan, China, Aug. 2009, pp. 572–575.
- [37] J. Zhang, J. Zheng, C. Wu, and J. Cai, "Variational mesh decomposition," *ACM Trans. Graph.*, vol. 31, no. 3, pp. 1–14, May 2012.
- [38] M. Ramanathan and B. Gurumoorthy, "Interior medial axis transform computation of 3D objects bound by free-form surfaces," *Comput.-Aided Des.*, vol. 42, no. 12, pp. 1217–1231, Dec. 2010.
- [39] Y. Yan, D. Letscher, and T. Ju, "Voxel cores: Efficient, robust, and provably good approximation of 3D medial axes," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–13, Jul. 2018.
- [40] R. Gal, A. Shamir, and D. Cohen-Or, "Pose-oblivious shape signature," *IEEE Trans. Vis. Comput. Graphics*, vol. 13, no. 2, pp. 261–271, Mar. 2007.
- [41] L. Shapira, A. Shamir, and D. Cohen-Or, "Consistent mesh partitioning and skeletonisation using the shape diameter function," *Vis. Comput.*, vol. 24, no. 4, pp. 249–259, Jan. 2008.
- [42] M. Huska and S. Morigi, "A meshless strategy for shape diameter analysis," *Vis. Comput.*, vol. 33, no. 3, pp. 303–315, 2017.
- [43] S. Chen, T. Liu, Z. Shu, S. Xin, Y. He, and C. Tu, "Fast and robust shape diameter function," *PLoS ONE*, vol. 13, no. 1, Jan. 2018, Art. no. e0190666.
- [44] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof, "From structure-from-motion point clouds to fast location recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 2599–2606.
- [45] R. A. Newcombe, A. Fitzgibbon, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, and S. Hodges, "KinectFusion: Real-time dense surface mapping and tracking," in *Proc. 10th IEEE Int. Symp. Mixed Augmented Reality*, Basel, Switzerland, Oct. 2011, pp. 127–136.
- [46] T. Whelan, S. Leutenegger, R. Salas Moreno, B. Glocker, and A. Davison, "ElasticFusion: Dense SLAM without a pose graph," in *Proc. Robot., Sci. Syst. XI*, Rome, Italy, Jul. 2015, pp. 1–9.
- [47] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004.

- [48] P. Theologou, I. Pratikakis, and T. Theoharis, "A comprehensive overview of methodologies and performance evaluation frameworks in 3D mesh segmentation," *Comput. Vis. Image Understand.*, vol. 135, pp. 49–82, Jun. 2015.
- [49] I. Biederman, "Recognition-by-components: A theory of human image understanding," *Psychol. Rev.*, vol. 94, no. 2, pp. 47–115, 1987.
- [50] D. D. Hoffman and M. Singh, "Saliency of visual parts," *Cognition*, vol. 63, no. 1, pp. 29–78, Apr. 1997.
- [51] Y. H. Chen and C. Y. Liu, "Quadric surface extraction using genetic algorithms," *Comput.-Aided Des.*, vol. 31, no. 2, pp. 101–110, Feb. 1999.
- [52] M. Soleimani, O. Dorn, and W. R. B. Lionheart, "A narrow-band level set method applied to EIT in brain for cryosurgery monitoring," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 11, pp. 2257–2264, Nov. 2006.
- [53] M. Zhang, J. Peng, and X. Liu, "Sparse coding with earth mover's distance for multi-instance histogram representation," *Neural Comput. Appl.*, vol. 28, no. 12, pp. 3697–3708, 2017.
- [54] L. Garmendia, R. González del Campo, and J. Recasens, "Partial orderings for hesitant fuzzy sets," *Int. J. Approx. Reasoning*, vol. 84, pp. 159–167, May 2017.
- [55] X. Chen, A. Golovinskiy, and T. Funkhouser, "A benchmark for 3D mesh segmentation," *ACM Trans. Graph.*, vol. 28, no. 3, p. 1, Jul. 2009.
- [56] M. Schoeler, J. Papon, and F. Worgotter, "Constrained planar cuts—object partitioning for point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 5207–5215.
- [57] A. Golovinskiy and T. Funkhouser, "Randomized cuts for 3D mesh analysis," *ACM Trans. Graph.*, vol. 27, no. 5, pp. 1–12, Dec. 2008.
- [58] E. Kalogerakis, A. Hertzmann, and K. Singh, "Learning 3D mesh segmentation and labeling," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 1–12, Jul. 2010.



QINGDE LI received the Ph.D. degree in computer science from the University of Hull, U.K. He is currently a Lecturer with the Department of Computer Science, University of Hull. His current research interests include computer graphics, medical image processing, and geometric modeling.



LIGANG HE (Member, IEEE) received the Ph.D. degree in computer science from the University of Warwick, U.K. He worked as a Postdoctoral Researcher at the University of Cambridge, U.K. Since 2006, he has been with the Department of Computer Science, University of Warwick, as an Assistant Professor and then an Associate Professor. He has published more than 100 articles in international conferences and journals, such as the *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS*, *IPDPS*, *CCGrid*, and *MASCOTS*. His research interests focus on parallel and distributed processing, cluster, grid, and cloud computing. He has been a Co-Chair or a member of the program committee for a number of international conferences, and has been the Reviewer for many international journals, including the *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS* and the *IEEE TRANSACTIONS ON COMPUTERS*.



XIAOWEN YANG received the master's degree in computer application technology from the North University of China, China, in 2005. She is currently pursuing the Ph.D. degree in communication and information system. She is also an Associate Professor with the School of Data Science and Technology, North University of China (NUC). Her research interests focus on computer graphics, 3D model editing, and virtual reality.



MIN PANG received the master's degree in computer application technology from the North University of China, China, in 2009, where she is currently pursuing the Ph.D. degree in system simulation and modeling. She is also a Lecturer with the School of Data Science and Technology, North University of China (NUC). Her current research interests include computer graphics, 3D shape deformation, and virtual reality.



XIE HAN received the master's degree in computer science and technology from the North China Institute of Technology, China, and the Ph.D. degree from the Institute of Information Engineering, University of Science and Technology Beijing, China, in 2002. She is currently a Professor with the School of Data Science and Technology, North University of China. Her current research interests include computer vision, simulation, and visualization.



CAIQIN JIA is currently pursuing the Ph.D. degree in system simulation and modeling with the School of Data Science and Technology, North University of China (NUC), China. Her current research interests include computer graphics, 3D fragment reassembly, and virtual reality.

...