Optical Engineering

OpticalEngineering.SPIEDigitalLibrary.org

# Optimized shape semantic graph representation for object understanding and recognition in point clouds

Xiaojuan Ning
Yinghui Wang
Weiliang Meng
Xiaopeng Zhang

SPIE.

Xiaojuan Ning, Yinghui Wang, Weiliang Meng, Xiaopeng Zhang, "Optimized shape semantic graph representation for object understanding and recognition in point clouds," *Opt. Eng.* **55**(10), 103111 (2016), doi: 10.1117/1.OE.55.10.103111.

# Optimized shape semantic graph representation for object understanding and recognition in point clouds

**Xiaojuan Ning,[a,*] Yinghui Wang,[a] Weiliang Meng,[b] and Xiaopeng Zhang[b]**
[a]Xi'an University of Technology, Department of Computer Science and Engineering, No. 5 South Jinhua Road, Xi'an 710048, China
[b]Chinese Academy of Sciences, Institute of Automation, National Laboratory of Pattern Recognition, #95 East Zhongguancun Road, Beijing 100190, China

**Abstract.** To understand and recognize the three-dimensional (3-D) objects represented as point cloud data, we use an optimized shape semantic graph (SSG) to describe 3-D objects. Based on the decomposed components of an object, the boundary surface of different components and the topology of components, the SSG gives a semantic description that is consistent with human vision perception. The similarity measurement of the SSG for different objects is effective for distinguishing the type of object and finding the most similar one. Experiments using a shape database show that the SSG is valuable for capturing the components of the objects and the corresponding relations between them. The SSG is not only suitable for an object without any loops but also appropriate for an object with loops to represent the shape and the topology. Moreover, a two-step progressive similarity measurement strategy is proposed to effectively improve the recognition rate in the shape database containing point-sample data. [DOI: 10.1117/1.OE.55.10.103111]

Keywords: laser scanned object; shape decomposition; shape semantic graph; similarity measurement.

Paper 161043 received Jun. 29, 2016; accepted for publication Oct. 3, 2016; published online Oct. 25, 2016.

## 1 Introduction

Point cloud has become a popular representation for three-dimensional (3-D) models in recent years owing to the improvements in digital scanning technology. Understanding the shape of objects in point clouds is one of the most fundamental problems in shape processing, and can promote meaningful research, such as multidimensional media, dealing with semantic-based knowledge systems fields. The key challenge to shape understanding is to improve the structure and the topology representation, with the ultimate goal being to obtain an optimized shape representation model. Naturally, shape representation becomes more difficult with a large number of loops in an object.

In this study, the understanding and recognition of an object are related to the cognition and learning of geometric and topological properties of an object, and the purpose is to distinguish the object from others by determining the object type based on the geometric and semantic features. Many existing algorithms refer to semantic segmentation.[1–4] For a better understanding of 3-D shapes, some methods were used to analyze the object by utilizing its structure,[5–8] and others made use of the topology.[9–12] Ning et al.[13] introduced a shape decomposition and skeleton extraction method that could understand the shape of objects without constructing the mesh or any other surface representation, but it would fail when dealing with an object containing a loop.

To overcome this problem, we propose an improved algorithm, extracting junction points to handle the existence of a loop in objects and generating an improved shape semantic graph (SSG) representation of the objects. The SSG is defined by a set of detected components and their connections. Based on the SSG, our algorithm can not only process the simply connected object, but also deal with the nonsimple connected object. We validate our method by recognizing objects on various point-sampled models. We can also use our SSG to identify different objects from a point-sampled models database. In summary, our paper makes the following contributions:

1. An automatic extraction method of a decomposed skeleton that increases junction points aiming to better determine the complete topology.
2. An improved representation of an object shape called an SSG that can better describe object containing a loop.
3. An efficient recognition algorithm, based on a two-step similarity measurement of an improved SSG.

The rest of this paper is organized as follows. In Sec. 2, we review previous studies that are closely related to ours. In Sec. 3, we describe our process and give detailed steps of our proposed method with the terminology involved in the method. After a discussion of the decomposition method in Sec. 4, we present an SSG of the object, and analyze on its properties in Sec. 5. In Sec. 6, a two-step similarity measurement method is described that recognizes different objects based on the SSG. Finally, we discuss the efficiency and of our decomposition method and make comparisons by providing the results of the boundary extraction and SSG. We also demonstrate the application of object recognition based on the SSG in point model database.

## 2 Related Work

In recent years, researchers have proposed many shape understanding methods for planar shapes.[14–17] In 3-D shape

*Address all correspondence to: Xiaojuan Ning, E-mail: ningxiaojuan@xaut.edu.cn

understanding, Attene et al.[18] classified the methods of shape understanding into two categories from the computational perspective: the representation based on geometry and structure, and the representation based on topology.

### 2.1 *Geometry and Structure-Based Representation*

Geometry and structure-based representation involves the scope of the object (geometry representation), the object characteristics, and object decomposition components (structure representation). Generally, the definition of an ideal shape descriptor is required to capture and compute the main features of a surface, and extract the geometric shape that is invariant to rotation, translation, and scaling. This representation distinguishes the local and global features that could be combined and abstracted into a compact representation that is useful in various applications such as shape matching, shape retrieval, and shape comparison. The research institution, Institute of Applied Mathematics and Information Technologies—Genova (IMATI CNR), developed the project AIM@SHAPE. The goal was to develop a semantic-based shape representation, and design a semantic-oriented tool to obtain, construct, transfer, and deal with the shape using related knowledge.

Generally, the feature representation of a surface has two types: a local surface descriptor and a global surface descriptor. The global surface descriptor mainly describes the whole or one of the most significant features of the 3-D objects that are commonly used in 3-D object matching and classification. The local surface descriptor represents the geometrical features of a vertex's neighborhood on the surface, which can be used in object identification, model matching, and shape registration.

At present, there are numerous studies focused on using local surface descriptors to represent the shape of an object.[19–21] In the following, we will review several similar studies in shape understanding of the 3-D object.

De Figueiredo and Kehtarnavaz[5] assumed that the object was composed of some smooth fragmentation sets and denoted the object as an attribute graph in which the attributes of each node was defined by the Gaussian curvature of the corresponding fragmentation. Their object recognition was primarily based on the graph matching method. Stein and Medioni[6] focused on the intensive data, and they generally adopted two main characteristics that were coded to retrieve and match information from database quickly. Chua and Jarvis[7] presented a feature descriptor—point signature, then used it for recognition of an object based on the calculation of characteristics and voting mechanism. Johnson and Hebert[8] proposed a spin image representation method to determine the surface shape from the dense sample points. Sidi et al.[22] introduced an unsupervised cosegmentation method to reveal the semantic shape parts and established their correspondence across the set. Guo et al.[20] provided a guidance for the selection of an appropriate 3-D feature descriptor in different applications, and further they presented a comprehensive survey of existing local surface feature-based 3-D object recognition methods.[21]

### 2.2 *Topology-Based Representation*

Topology-based representation can capture and understand the shape of spatial objects by matching and distinguishing different objects through mathematical tools.[23] Classical tools such as Morse theory,[23,24] Reeb graph, homotype, and homology are suitable for dealing with several issues related to shape understanding. Morse theory sets the foundation for associating the topology of a given manifold with the critical points of a smooth function defined on the manifold. In recent years, Morse theory has been applied successfully to data visualization in the scalar field, and is often used to construct a multiresolution structure.

Many methods adopted the Reeb graph to analyze the topological structure of models and obtain a representation of the corresponding topology, further generating the semantic description of an object that can be used for shape understanding and recognition.

Hilaga et al.[9] made use of the surface geodesic distance as the Morse function and proposed a multiscale Reeb graph algorithm. Based on the Morse theory and Reeb graph, Biasotti[10] presented an extended Reeb graph which can be used to represent the relationship of points. This method could provide an effective way to classify, simplify, and store the model. Dey et al.[11] investigated the mesh segmentation using a smooth function defined on the discrete domain based on Morse theory. Xiao et al.[12] adopted a discrete Reeb graph approach to analyze the topological structure of the human model. In addition, Bespalov et al.[25] proposed a distance matrix of vertices, by which the model is decomposed to acquire the Reeb graph. Tung and Francis[26] provided an incremental Reeb graph algorithm that adopted the height function as the Morse function. Hui and Floriani[27] proposed a two-level topological decomposition method and the decomposition relationship between components to implement shape understanding. Schnabel et al.[28] described semantic entities as a constrained shape graph, and studied the shape understanding, including the shape problem of 3-D point cloud data. Floriani et al.[29] presented TopMesh, a tool for extracting topological information from nonmanifold, 3-D objects with parts of nonuniform dimensions.

Many works are proposed for extracting the skeleton from an object to represent the topology. Dey and Sun[30] introduced a mathematical definition of the curve skeleton. Mesh contraction using Laplacian smoothing was first employed by Au et al.[31,32] to find skeletons of mesh surfaces and was extended to handle point sets by Cao et al.[33] Tagliasacchi et al.[34] presented a mesh-based contraction algorithm by incorporating Voronoi pole structure into the mean curvature flow. Our work is related to the research by Ning et al.[13] that had errors when dealing with an object containing loops. Our work depends on the following crucial aspects:

(1) Our algorithm integrates the structure and topological characteristics and proposes an optimized SSG that could make shape representation more intuitive and robust and (2) our topological relation representation devises a boundary surface for different parts of the object, thus maintaining the topological structure information.

## 3 Algorithm Overview

In this paper, we propose an effective method to understand and recognize the object based on decomposition and topology relations. Decomposition means decomposing the object into components, and topology relations indicate the connection between these components. The characteristics of an object can be easily recognized using the structure and topology representation. Assume that the 3-D object is represented by $\Omega$ and the skeleton is denoted by $S$, we give the following definition:

Decomposition:
The original object $\Omega$ is comprised of a set of points $\{p_1, p_2, \dots, p_n\}$ and can be decomposed into different parts $\Omega_1, \Omega_2, \dots, \Omega_m$, in which $\Omega_i = \{p_1^i, p_2^i, \dots, p_j^i\}$. Each part has a relatively independent meaning for $\Omega$.

Topological relations:
We use a decomposed, centralized skeleton $S$ to describe the topological relations of $\Omega$. $S$ can provide the shape of $\Omega$ and has a one-to-one correspondence with $\Omega$, i.e., $S = \{S_1, S_2, \dots, S_m\}$, and $S_1$ corresponds to $\Omega_1$, and so on.

Feature points:
A set of critical points that could be considered as the representative points for different parts, denoted as $\mathrm{Fp} = \{\mathrm{fp}_1, \mathrm{fp}_2, \dots, \mathrm{fp}_m\}$. They conform to the norm of human perception.

Central point:
The center $O$ of the object $\Omega$ is defined as the point with the minimum average geodesic distance to all other points, especially the point that satisfies $\operatorname{argmin}[\sum_{p \in P} G^2(p, p_i)]$.

Junction points:
The intersection of two neighboring parts $\Omega_i$ and $\Omega_j$ denoted as $J_{ij} = \{\rho_1^{ij}, \rho_2^{ij}, \dots, \rho_{m-1}^{ij}\}$.

The feature points Fp are often located on the contour of objects. We first adopted the alpha-shape-based method to detect the contour points, and then obtained the optimal feature points by clustering those points and refining the points of local curvature maxima.[13] Fp, $O$, and $J$ are essential elements in topological relations representation. The details of our algorithm are given as follows:

1. The object $\Omega$ is decomposed into a few disjointed, meaningful sets based on the chosen feature points Fp.
2. The geodesic distance between feature points Fp and central point $O$ is calculated, and the points on the geodesic lines are labeled according to different ingredients and are called the initial surface skeleton IS. Moreover, the label variation of points on IS could be used to generate the junction points/boundary surface points $J_{ij}$.
3. IS is placed in the center of the object and is simplified to obtain the final surface skeleton $S$.
4. The SSG is constructed based on $\Omega$, $S$, $J$, Fp, and $O$.
5. A similarity measurement is designed using the SSG of $\Omega$ to recognize the object.

## 4 Decomposition and Topology Representation for the Object

### 4.1 *Decomposition of the Object*

Decomposition produces the object structure including the component information and can be used to guide the topological structure generation. A perception-based approach to decompose the object in a point cloud is presented by Ning et al.[13] that follows a rule that segments an object along lines of negative minimum curvature. To determine the number of patches, we calculate and select the critical feature points based on perception information to represent each patch. Taking the critical marker sets as a guide, each marker is spread to a meaningful region by curvature-based decomposition and further constraints are provided by the variation of normals. A brief introduction to its strong connection to our work is as follows.

1. The method first automatically extracts those feature points with local curvature maxima that appear on a convex hull of contour points.
2. Second, the feature points and the variance in the normal direction over the variance in the neighborhood are determined to weigh whether the region is smooth.
3. Next, taking the feature points as the seed points, our method identifies connected point sets by growing the seed points on the basis of the constraint that the points belonging to one patch have little variation of normal vectors.

Afterward, object $\Omega$ represented by point cloud is divided into $m$ parts ($m$ is not less than the number of critical points), thus, $\Omega = \bigcup_{i=1}^{m} \Omega_i$, $\Omega_i \cap \Omega_j = \varnothing$. Here, $\Omega_i$ is called a patch of $\Omega$. As such, the object $\Omega$ is comprised of $\Omega_1, \Omega_2, \dots, \Omega_m$. Moreover, if the object has too many points, we can simplify the data to save the computation time while keeping its characteristics using Morton ordering.[35]

### 4.2 *Topological Relations of Components*

#### 4.2.1 *Surface skeleton*

We use the method in Ning et al.[13] to handle the teapot data and the result is shown in Fig. 1. We found that the extracted surface skeleton misses the important loop information of the teapot handle that may have an impact on the complete shape understanding. Therefore, we propose an improved method based on the junction points of different components in object $\Omega$.

**Definition 1** The initial surface skeleton consists of a set of geodesic lines from the feature points to the center of the object that can be denoted as $\mathrm{IS} = \bigcup_{i=1}^{\zeta} \mathrm{IS}_i$, where $\zeta$ is the number of feature points.

**Definition 2** The decomposed skeleton is composed of points on $\mathrm{IS}_i$, with labels corresponding to different regions of $\Omega$.

**Definition 3** The junction points are those points whose $k$ neighbor vertices (based on $k$-d tree) exist label variation of initial surface skeleton.

**Definition 4** The final surface skeleton consists of a set of geodesic lines from the feature points $\mathrm{Fp} = \{\mathrm{fp}_1, \dots, \mathrm{fp}_m\}$ to the centroid of the boundary surface $J = \{\rho_1, \rho_2, \dots, \rho_{m-1}\}$ and from the centroid of the boundary surface to the central point $O$ of $\Omega$ that can be denoted as $S = \bigcup_{i=1}^{\zeta} S_i = \bigcup_{i=1}^{\zeta} \bigcup_{j=1}^{\tau} (\ell_{t_i \rho_j} \bigcup \ell_{\rho_j o}) = \bigcup_{i=1}^{\zeta} \{(\bigcup_{j=1}^{\tau} \eta_i^j)\}$. $\ell_{uv}$ denotes the shortest path between points $u$ and $v$.

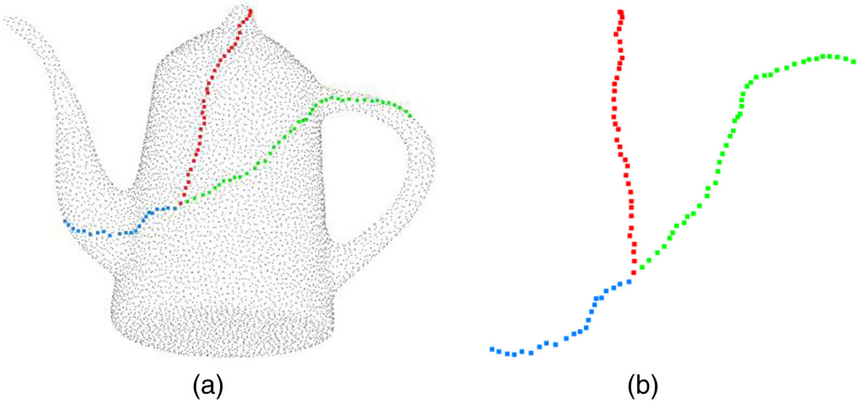


**Fig. 1** Surface skeleton of teapot. (a) Extracted skeleton and (b) enlarged part of the skeleton.

#### 4.2.2 *Junction points*

After decomposing and labeling each decomposed part of $\Omega$, the junction points are determined by several steps:

1. Detect the point whose label appears mutated in different regions such as $R_i$ and $R_j$ corresponding to label $i$ and $j$. These points are called junction points.
2. Guided by the contour points, we judge the label variations among the neighboring points and record the frequency with which each label appears. Then we select those points whose neighbors are only labeled $i$ and $j$ as the junction points. We repeat this step until all the points on the boundary surface are checked.

Figure 2(b) shows the junction points $J = \{\rho_1, \rho_2, \dots, \rho_{m-1}\}$ between decomposed parts. For example, for two decomposed parts $\Omega_1$ and $\Omega_2$, we detect one point $\rho$ as the junction point/boundary point first. Then we take $\rho$ as the seed point, find its $k$ neighboring points, and mark those who have the label of “1” and “2” as the junction points. The junction points are clustered according to the nearest neighbor points. For the object with loops, the junction points are clustered into two different parts leading to two different boundary surfaces (Fig. 3). Based on the final boundary surface of the teapot, the surface skeleton can be obtained by connecting the feature points Fp with the corresponding point $\rho$ on the boundary surface and $\rho$ with center $O$ of the teapot (Fig. 4).

### 4.3 *Final Skeleton*

Based on the surface skeleton extraction results, we should move it further into the interior of the object to centralize the skeleton.

Let $S = \{S_1, S_2, S_3, \dots, S_\zeta\}$ be the final surface skeleton set. The arbitrary point $\eta_\tau^i$ on $S$ is moved into the interior of the object in the reverse surface normal direction, with contracting procedure

$$\eta_\tau^{i\prime} = \eta_\tau^i + \text{normalize}[W_F(\eta_\tau^i)] * e, \tag{1}$$

where $e$ is a distance defined by the user, and $W_F$ is the repulsive force defined in Wu et al.[36] that is calculated by $W_F(x) = \sum_{q_i \in V(x)} F(\|q_i - x\|_2) \cdot (q_i - x)$, with the Newtonian potential function $F(r) = r^{-2}$. The $k$-nearest neighboring point set is $V(x) = \{q_1, q_2, \dots, q_k\}$. The iteration continues until $|W_F(\eta_\tau^{i\prime})| > |W_F(\eta_\tau^i)|$ and the final position of $\eta_\tau^i$ is recorded.

If the points on the final skeleton $S$ are dense, we can simplify and smooth the skeleton according to the label variation and the angle between two arbitrary conjoint segments (the segment is the line created by connecting two neighboring points). Finally, a smooth, simplified, and centralized skeleton is generated, which will be applied in the next section.

## 5 Shape Semantic Graph

As a shape representation, the SSG can describe the topology of objects efficiently and has wide applications in 3-D model retrieval. The SSG is unique and can capture the critical topology information for the object.

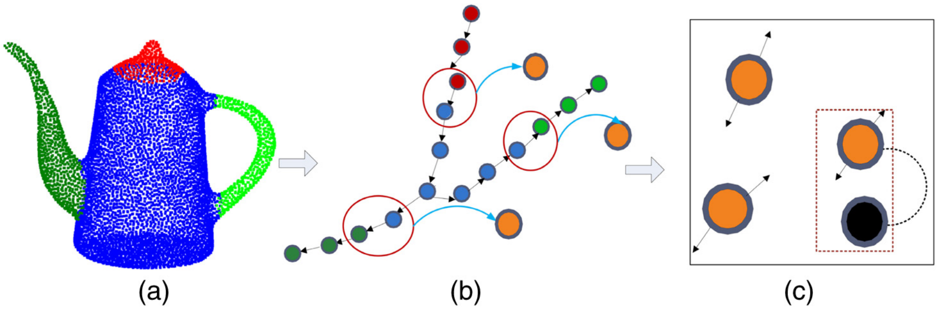


**Fig. 2** Decomposition and the boundary surface of the teapot. (a) Shape decomposition, in which one color represents one component, (b) points on the boundary of decomposed parts (in red circles), and (c) boundary surface detection.

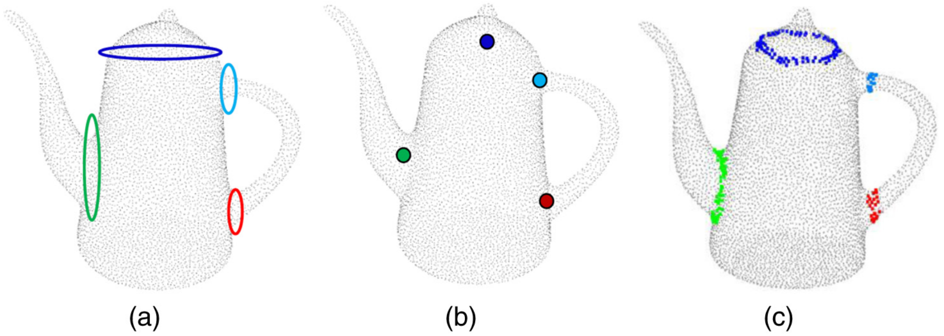


**Fig. 3** Final boundary surface of the teapot. (a) Diagram of the boundary surface, (b) variation of labels, and (c) boundary surface of the loop shape.

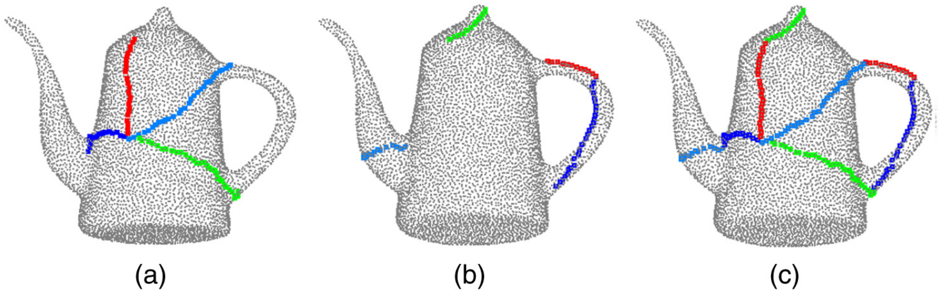


**Fig. 4** Surface skeleton extraction. (a) Geodesic lines from boundary surface points *J* to the center *O* of the object, (b) geodesic lines from feature points Fp to boundary surface points *J*, and (c) surface skeleton.

The SSG is defined by $\mathbb{G} =< V, E >$, where $V$ represents the decomposed subparts $\Omega_i$, $V = \{V_1, V_2, \dots, V_m\}$, and $E = E(\mathbb{G}) = \{(V_1, V_2), \dots, (V_i, V_j), \dots\}$, recording the topological relations between the two subparts when there is a joint that transitions from one of the labeled parts to the other connecting the skeleton points. After decomposition, we regard each part as a node [represented by the left one in Fig. 5(a)] and two nodes have an edge, if and only if they are adjacent. The adjacent relations can be evaluated by the skeletal structure [Fig. 5(b)].

**Definition 5** For the node in the SSG $\mathbb{G}$, if there is only one adjacent node, especially if the node degree is 1, the note is called the endpoint of the SSG and can be treated as a feature point of the object ("loop" structure is excluded). The junction points between the two decomposed components construct a triangle node [Fig. 5(b)].

**Property 1** The node $V_i$ is corresponding to part $i$ in the object (the node may belong to the decomposition part and also to the point on the boundary surface of two parts). Each $V_i$ has a tag to record the type of the node tag $\in \{0,1\}$. If tag $= 0$, then the node $V_i$ in SSG belongs to the junction points. If tag $= 1$, the node $V_i$ in SSG belongs to a decomposition part.

**Property 2** The central point of the object has the highest degree among the nodes in SSG.

**Property 3** The topological relationship of object components can be represented by $E = \{E_{12}, \dots, E_{ij}, \dots\}$, where $E$ is composed of $\{(V_1, V_2), \dots, (V_i, V_j), \dots\}$. $E$ records the neighborhood around nodes and has a length that is determined by the geodesic distances between two nodes, especially, $E_{ij} = (V_i, V_j)$. The length of $e_{ij}$ can be calculated by $l_{E_{ij}} = \mathcal{G}(V_i, V_j)$, and $\mathcal{G}(.)$ denotes the geodesic distance between two points of the object.

Figure 5 shows the SSG of the teapot, demonstrating that the SSG can effectively describe the shape of an object with

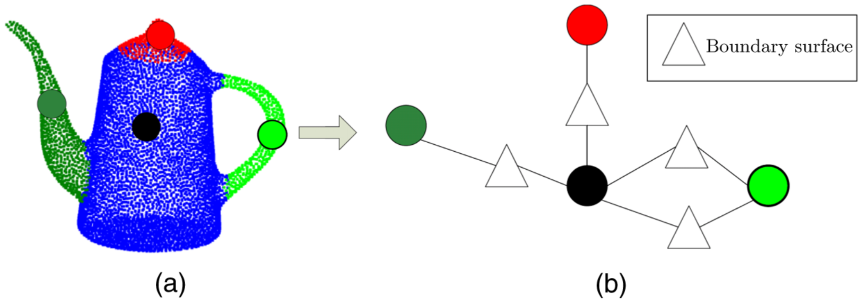


**Fig. 5** SSG for teapot. (a) Nodes after decomposition and (b) boundary surface between different parts.

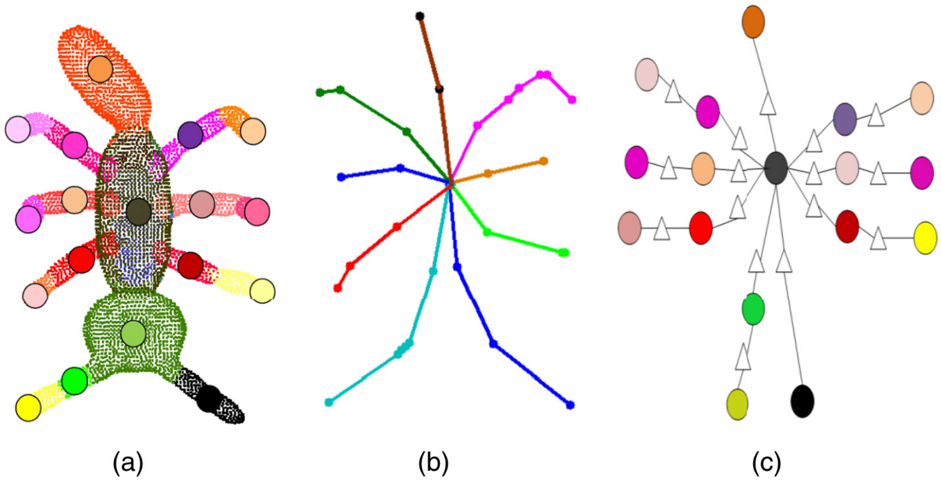


**Fig. 6** Generation process for SSG of an ant. (a) Ant decomposition, (b) corresponding skeleton, and (c) SSG of an ant.

loops or without loops. Figure 6 shows the SSG generation process of ant.

## 6 Object Recognition Based On Similarity Measurement

### 6.1 *Similarity Measurement*

**Definition 6** The similarity measurement between two models is calculated by

$$\begin{aligned}\Omega(\mathbb{G}_{M1},\mathbb{G}_{M2}) = {} & 0.25 * C(V_{\mathbb{G}_{M1}},U_{\mathbb{G}_{M2}}) \\ & + 0.25 * C[D(O_{\mathbb{G}_{M1}}),D(O_{\mathbb{G}_{M2}})] \\ & + 0.25 * C[V^1_{\mathbb{G}_{M1}},U^1_{\mathbb{G}_{M2}})] \\ & + 0.25 * C(V^2_{\mathbb{G}_{M1}},U^2_{\mathbb{G}_{M2}}), \end{aligned} \tag{2}$$

where $M_1$ and $M_2$ denote two models, and $C(.)$ represents the comparison of data. The value of $C(.)$ is generally 0 or 1, where 0 means the two models are different and 1 means the two models are similar. $V$ and $U$ are the number of vertices in the SSG of $M_1$ and $M_2$ respectively, $V = \{V_1, V_2, \dots, V_\nu\}$ and $U = \{U_1, U_2, \dots, U_\mu\}$. $C(V^1, U^1)$ compares the nodes whose degree is 1 in $\mathbb{M}_1$ and $\mathbb{M}_2$. $C(V^2, U^2)$ compares the nodes whose degree is 2 in $\mathbb{M}_1$ and $\mathbb{M}_2$. $C[D(O_{\mathbb{G}_{M1}}), D(O_{\mathbb{G}_{M2}})]$ compares the nodes that have the largest degree in $\mathbb{M}_1$ and $\mathbb{M}_2$.

The similarity measurement is used to select the most similar model in the database. Assuming that the input model is $M_1$, after choosing a model $M_2$ from the database, the comparison steps for the SSG $\mathbb{G}_{M1}$, $\mathbb{G}_{M2}$ of $M_1$ and $M_2$ are as follows:

1. Compare the number of nodes $V_{\mathbb{G}M_1}$ and $U_{\mathbb{G}M_2}$. If they are consistent, then $C(V_{\mathbb{G}M_1}, U_{\mathbb{G}M_2}) = 1$, and go to step (2). Else, $C(V_{\mathbb{G}M_1}, U_{\mathbb{G}M_2}) = 0$ and $M_2$ is not the model similar to $M_1$. Choose another model and continue (1).
2. Compare the number of nodes that have the largest degree in $\mathbb{G}_{M1}$ and $\mathbb{G}_{M2}$. If $D(O_{\mathbb{G}_{M1}}) = D(O_{\mathbb{G}_{M2}})$ then $C(D(O_{\mathbb{G}_{M1}}), D(O_{\mathbb{G}_{M2}}) = 1$ and go to (3). Else, $C(D(O_{\mathbb{G}_{M1}}), D(O_{\mathbb{G}_{M2}}) = 0$. Then choose another model and start from (1).
3. Compare the number of nodes whose degree is 1 in $\mathbb{G}_{M1}$ and $\mathbb{G}_{M2}$. If they are consistent, let $C(V^1_{\mathbb{G}_{M1}}, U^1_{\mathbb{G}_{M2}}) = 1$ and go to (4). Else, $C(V^1_{\mathbb{G}_{M1}}, U^1_{\mathbb{G}_{M2}}) = 0$. Then choose another model and start from (1).
4. Compare the number of nodes whose degree is 2 in $\mathbb{G}_{M1}$ and $\mathbb{G}_{M2}$. If they are consistent, let $C(V^2_{\mathbb{G}_{M1}}, U^2_{\mathbb{G}_{M2}}) = 1$ and insert the model into a model set that is empty initially. It is a matched model. Else $C(V^2_{\mathbb{G}_{M1}}, U^2_{\mathbb{G}_{M2}}) = 0$. Then choose another model and start from (1).

Based on (1) to (4), we can acquire a set of models $\{M_r\}$ that match the given model $M_1$. The type of $M_1$ must be consistent with that of one model in $\{M_r\}$. Thus, we analyze the types of the models in $\{M_r\}$ and choose one that appears frequently as the type of $M_1$. After the initial choice, we can rule out those data that do not match with $M_1$.

Figure 7 shows the initial choice from the database. Assume the given model $M_1$ is tippy [see Fig. 7(a)]. We can find a series of data from the database using the initial similarity matching, shown in Fig. 7(b). Since each object in the shape database has seven models under different poses, it could recognize the type of the object according to the frequency that the object appears in $M_\gamma$. Hence, the given model $M_1$ can be recognized as tippy in Fig. 7(b).

### 6.2 *Progressive Similarity Measurement*

Objects with different shapes can be distinguished by the initial similarity measurement, however, it is difficult to make a distinction between different poses of one object. For example, a four-leg table is consistent with four-leg tables in the database regardless of its model decomposition results, skeleton structure, or shape semantic maps. Then how do we acquire the consistent results corresponding to the initial input model? In addition, for a given model, one or more similar models could be detected (as shown in Fig. 7). However, how do we determine which one is the closest or most consistent with the given object? In order to solve these problems, we need an additional similarity detection called "progressive similarity comparison" after the initial choice.

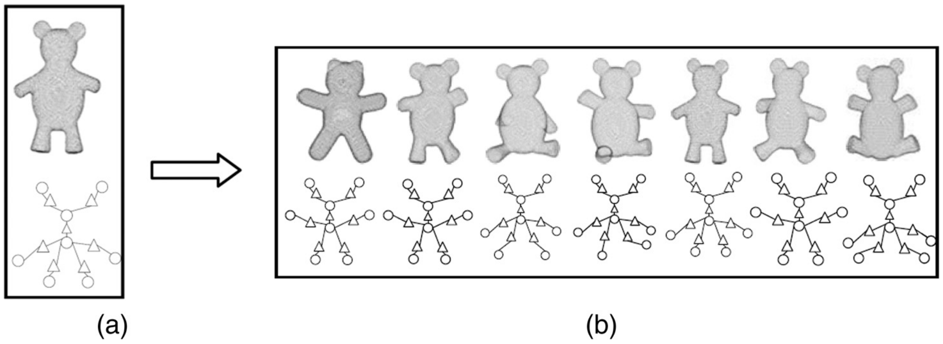


**Fig. 7** Retrieval of "tippy." (a) The given model "tippy" and its SSG and (b) the query result from the database.

**Definition 7** The progressive measurement of two models is defined by the histogram of the geodesic distance between the point with a degree of 1 and the point with the largest degree. The geodesic distance is divided into equal intervals by using the maximum and minimum geodesic distances.

Figures 8(a) and 8(b) show the geodesic distance histogram of tippy and the horse, respectively. The histogram comparison method[37] contains mainly $\chi^2$, Bhattacharyya distance, PDF$L_N$, CDF$L_N$

$$\chi^2 : D(f,g) = \int (f-g)^2/(f+g),$$

$$\text{Bhattacharyya} : D(f,g) = 1 - \int \text{sqrt}(fg),$$

$$\text{PDF}L_N : \text{Minkowski } L_N \text{ norm of the PDF} : D(f,g) = \left( \int |f-g|^N \right)^{1/N},$$

$$\text{CDF}L_N : \text{Minkowski } L_N \text{ norm of the CDF} : D(f,g) = \left( \int |\hat{f}-\hat{g}|^N \right)^{1/N}.$$

The Bhattacharyya distance is adopted for comparison in this paper. In our example, the geodesic distance histogram of tippy and the horse data are compared and the divergence value is found to be 0.9008. Generally, the smaller this value is, the smaller the difference is. When the Bhattacharyya distance is 0, the two objects are identical.

## 7 Experimental Results Analysis

All the tests in the paper are on a PC that has an Intel Core2 Duo 2.80 GHz CPU, together with an integrated graphics card, Intel G33/G31 express chipset, and 2G of RAM. All the data used in our experiments are from the Princeton segmentation benchmark.[38] We chose seven objects under seven different poses as testing data.

### 7.1 *Analysis on Decomposition Algorithm*

Figure 9(a) shows the time in each stage: $K$NN ($k$-nearest neighbor), Seg (segmentation process), Bou (boundary detection), Clu (clustering for further critical points selection), and Cri (final critical points determination) for different point cloud data. The computational complexities of the different stages are, respectively: $K$NN: $O[kn \log(n)]$, Bou: $O[n \log(n)]$, Clu: $O[n \log(n)]$, Cri: $O[\log(n)]$, and Seg: $O[n^2 \log(n)]$. $K$NN, boundary detection, clustering, and

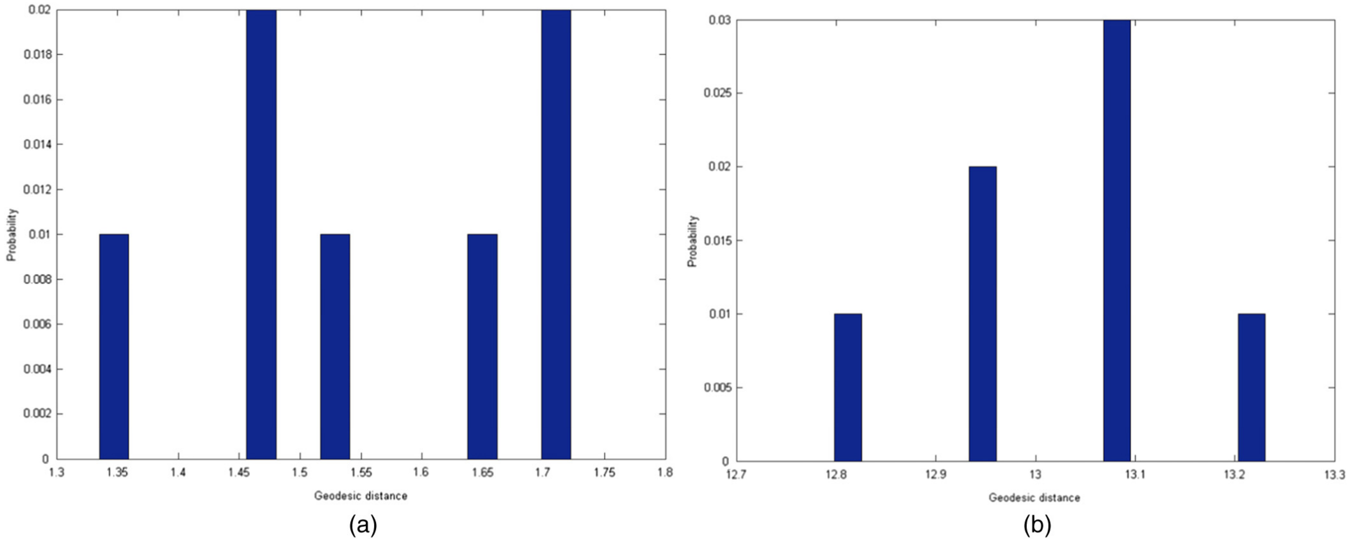
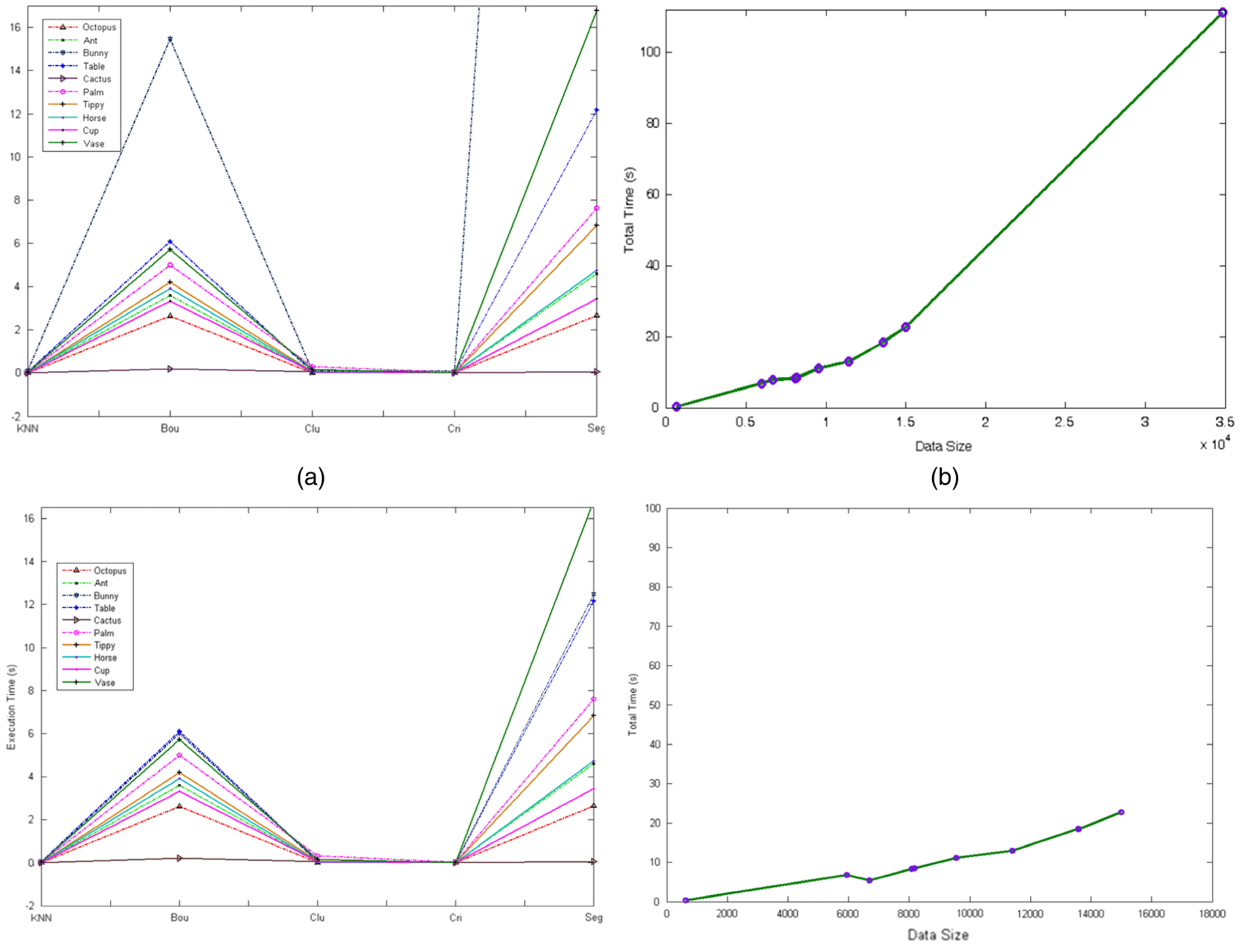


**Fig. 8** Geodesic distance histogram for (a) tippy model and (b) horse model.



(a)

(b)

(c)

(d)

**Fig. 9** Execution time of our decomposition algorithm. (a) Execution time for various stages of our algorithm with several different models. (b) Relations between the total time and data size. (c) and (d) Time after simplification compared with (a) and (b). All times are measured in seconds.

the selection of critical points are very fast and finish in a matter of a few seconds as opposed to minutes. Our segmentation process is also fast when the data size is <20,000 points. Figure 9(b) shows the relationship between the total running time and the data size. Compared with Figs. 9(a) and 9(b), the running time and the total time are improved after simplification shown in Fig. 9(c) and Fig. 9(d). We compared the time before and after data simplification for bunny. Detailed data are recorded and the execution times for both bunny and simplified bunny are compared in Table 1. Notice that the times in the table are used for preprocessing, which signifies that we only perform the computation once. However, the bigger data (such as bunny) can be simplified to improve the efficiency using Morton ordering. The results are shown in Fig. 10. Figures 10(b) and 10(c) show the data after using different sampling rates 50% and 33%, and the data after sampling contain 17,777 and 13,571 points, respectively. After simplification, the decomposition time could be improved effectively from 111.238 to 18.565 s. In addition, the simplification process is short and can process 2 million points of data in seconds. This could improve the execution time dramatically.

### 7.2 *Comparisons on Decomposition Algorithm*

Figure 11 shows the structure of four different objects (palm, octopus, tippy, and cactus). We also compared our results with related works (Ma et al.,[39] Richtsfeld and Vincze,[40] and Yamazaki et al.[41]) in Fig. 12. The result of Yamazaki et al.[41] is displayed in Fig. 12(a) [referred to as segmenting point sets (SPS)]. The results on the horse data using the Ma's method[39] [the segmentation with critical point and fuzzy clustering (SFC) method] are shown in Fig. 12(b). Richtsfeld and Vincze[40] obtained good results with a core part extraction that is based on the radical reflection [see Fig. 12(c) for segmentation based on radial reflection (SRR)]. Our method decomposes the object into meaningful components [Fig. 12(d)]. In view of the fact that our method can decompose the object into more detailed information, e.g., legs, ears, body, and horse faces, ours is useful for additional semantic labeling and recognition, especially in 3-D retrieval.

To run our algorithm, we first transform the meshes into point clouds, and then map the segmentation results back to the meshes, as described before. Figure 13 shows

**Table 1** Datasets and time analysis on each period in decomposition.

| | Data size | Run time (s) | | | | | |
|---|---|---|---|---|---|---|---|
| Dataset | $n$ | $k$NN | Bou | Clu | Cri | Seg | Total time |
| Octopus | 5944 | 0.01 | 2.625 | 0.015 | 2.609 | 2.656 | 7.915 |
| Ant | 8176 | 0.015 | 3.594 | 0.031 | 0.016 | 4.609 | 8.265 |
| Bunny | 34,835 | 0.11 | 15.460 | 0.094 | 0.090 | 95.484 | 111.238 |
| Simplified bunny | 13,571 | 0.041 | 6.000 | 0.064 | 0.020 | 12.44 | 18.565 |
| Table | 13,579 | 0.047 | 6.100 | 0.060 | 0.015 | 12.172 | 18.394 |
| Cactus | 620 | 0.001 | 0.200 | 0.047 | 0.002 | 0.047 | 0.294 |
| Palm | 11,413 | 0.020 | 5.000 | 0.310 | 0.032 | 7.625 | 12.987 |
| Tippy | 9548 | 0.010 | 4.200 | 0.070 | 0.010 | 6.840 | 11.130 |
| Horse | 8078 | 0.015 | 3.906 | 0.025 | 0.016 | 4.750 | 8.446 |
| Teapot | 6678 | 0.016 | 3.328 | 0.063 | 0.001 | 3.437 | 6.844 |
| Vase | 14,989 | 0.021 | 5.719 | 0.172 | 0.016 | 16.781 | 22.659 |

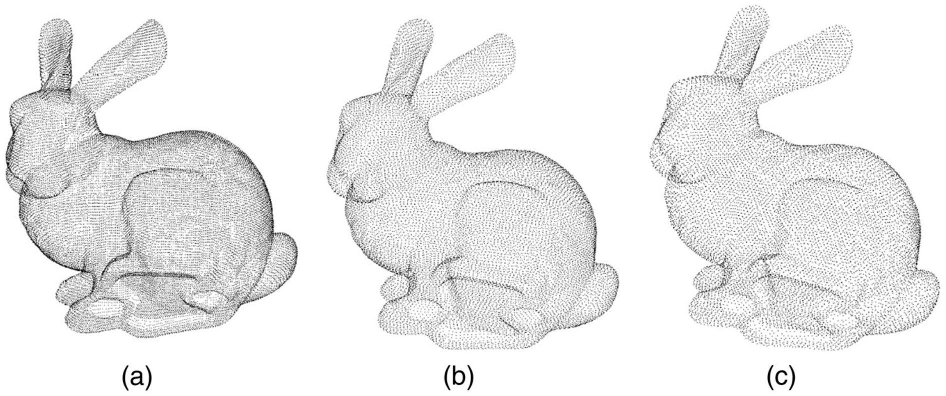


**Fig. 10** Data simplification. (a) Raw bunny data, (b) data with 50% sampling rate, and (c) data with 33% sampling rate.

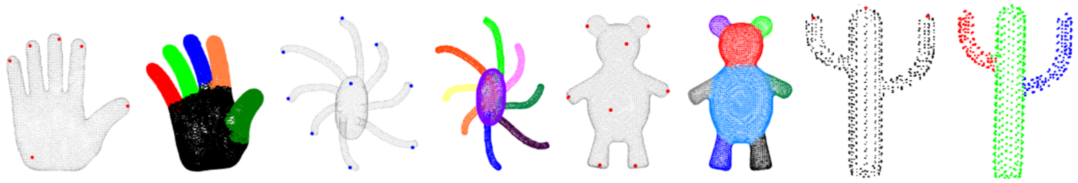


**Fig. 11** A gallery of decomposition results on different data.

a qualitative comparison on the tippy model and the cup model. We compared our segmentation method with the other six segmentation algorithms including deep learning,[42] randomized cut,[44] normalized cuts,[45] random walks,[46] $K$-means,[47] and approximate convexity analysis (WcSeg)[43] on the PSB database.

### 7.3 *Shape Semantic Graph Results*

It is necessary to extract the boundary and skeleton for additional processing of the SSG. Figures 14(a)–14(e) show the skeleton extraction results of several data (e.g., octopus, cactus, hand, horse, and teapot). Figure 15 shows the boundary extraction results on selected data of the object database from the Princeton segmentation benchmark.[38] It demonstrates that the boundary of each object can describe the shape contour of the object effectively from the optimal view.

The SSG of the shape database is also obtained [shown in Fig. 16(a)]. The similarity measurement and progressive similarity measurement are implemented on the SSG in Fig. 16(a). Afterward, the corresponding models are retrieved, as shown

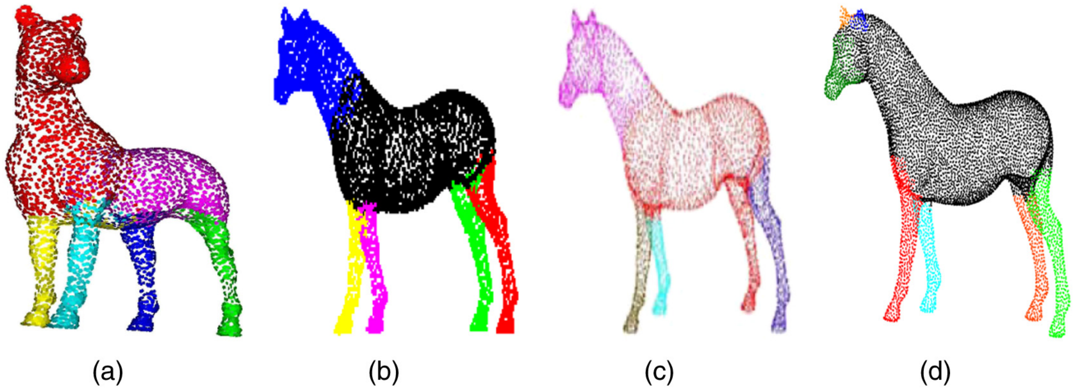


**Fig. 12** Segmentation comparison of the horse. (a) SPS, (b) SFC, (c) SRR, and (d) our method.

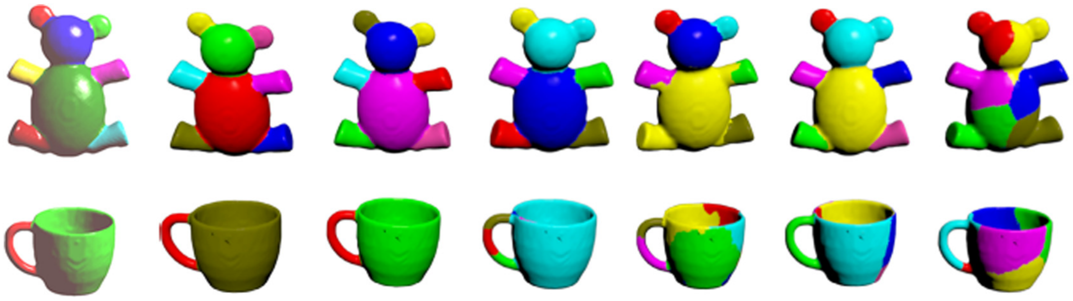


**Fig. 13** Comparison with the other six segmentation algorithms. The approaches used here include (from left to right): ours; deep learning;[42] WcSeg;[43] randomized cuts;[44] normalized cuts;[45] random walks;[46] and $K$-means.[47]

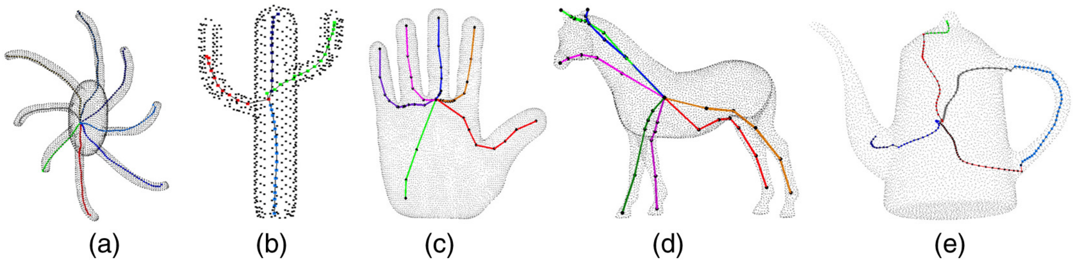


**Fig. 14** Skeleton extraction results. (a)–(e) The skeletons of an octopus, cactus, hand, horse, and teapot, respectively.

in Fig. 16(b). The first column in Fig. 16(b) indicates the query object. The objects with the dotted colored boxes belong to the same category as the query object by using the similarity measurement, and the final recognized result is represented by the dotted ellipses after the progressive similarity measurement.

The efficiency of the SSG when it is used to retrieve objects from the database is quite high. In our experiments, the database for objects in the point cloud is small, only containing 20 objects with 20 different postures (400 objects). In fact, the SSG, the degree of each vertex in the SSG, and the progressive measurement for 400 objects are obtained and recorded offline. For retrieval, we only need to compare the SSGs of two models, efficiently accomplished using the similarity measurement with the $O(n)$ complexity of the algorithm. Table 2 shows the execution time when using the similarity measurement of the SSGs for seven input models. Here, the execution time refers to online time not including the offline time. This demonstrates that our method is a fast and efficient way to implement object understanding and recognition.

We demonstrate the whole process of SSG generation for the object with loops. Figure 17 shows the whole process of shape decomposition [see in Figs. 17(a)–17(e)], skeleton extraction [from Figs. 17(f)–17(r)], and finally, SSG generation [Fig. 17(s)].

### 7.4 *Comparison*

Figures 18 and 19 compare the work of Ning et al.[13] with our algorithm on the handling of objects with loops. It is evident that our approach produces skeletons that capture the more

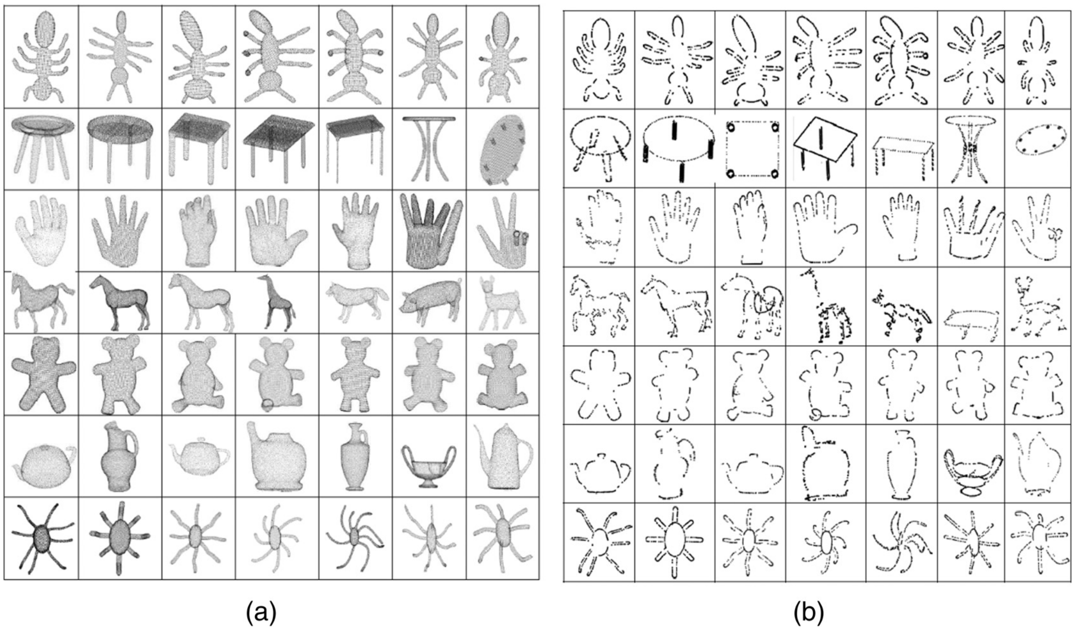


**Fig. 15** Boundary extraction. (a) Some data from the database and (b) corresponding boundary results.

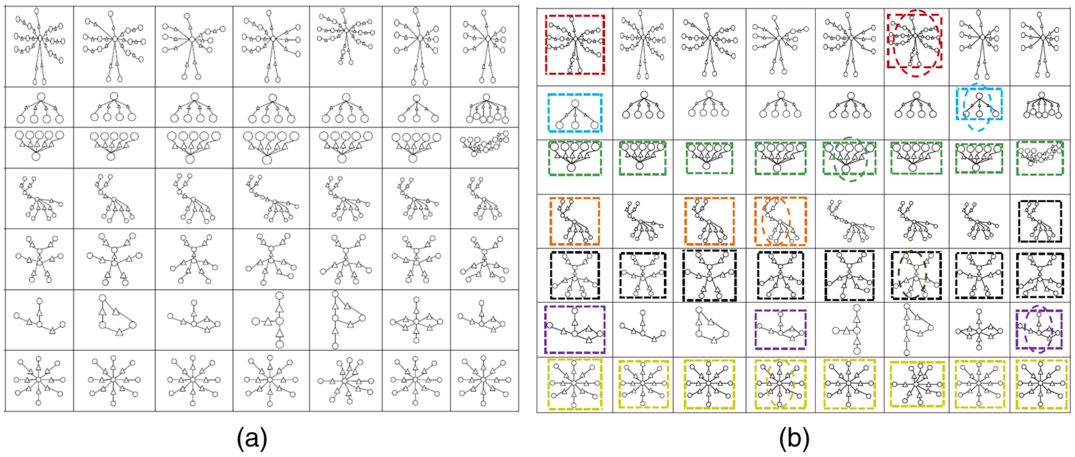


**Fig. 16** Object recognition based on SSG. (a) Corresponding SSG of objects and (b) recognition results.

**Table 2** Execution time for similarity measurement. $D_S$ is the size of dataset, $V_N$ denotes the number of vertices in SSG, $V_N^1$ is the number of vertices whose degree is 1, $V_N^2$ is the number of vertices whose degree is 2, $V_D^{\max}$ is the maximum degree in SSG, time refers to the consuming time for the online operation.

| Input | Ant | Table | Palm | Horse | Tippy | Teapot | Octopus |
|---|---|---|---|---|---|---|---|
| $D_S$ | 8176 | 13,579 | 11,413 | 8078 | 9548 | 6678 | 5944 |
| $V_N$ | 33 | 7 | 11 | 15 | 15 | 8 | 17 |
| $V_N^1$ | 9 | 3 | 5 | 6 | 6 | 2 | 8 |
| $V_N^2$ | 23 | 3 | 5 | 7 | 7 | 5 | 8 |
| $V_D^{\max}$ | 9 | 3 | 5 | 5 | 5 | 4 | 8 |
| Time (s) | 0.0166 | 0.0186 | 0.0323 | 0.0218 | 0.0333 | 0.0212 | 0.0317 |

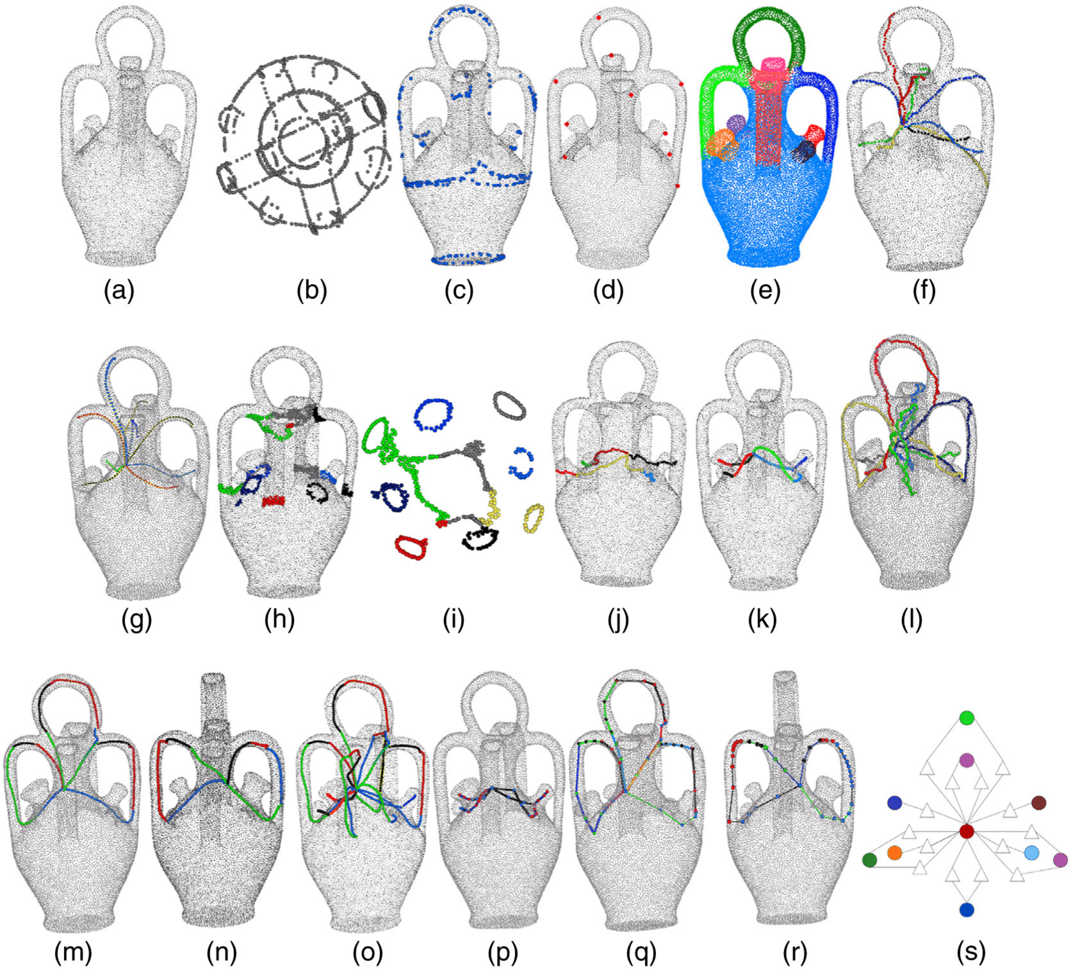


**Fig. 17** SSG generation for the vase model. (a) Original vase data, (b)–(e) part decomposition, (f) and (g) the surface skeletons, (h) and (i) the boundary surface of components, (j)–(o) the centralized skeletons, (p)–(r) the simplified skeletons, and (s) the SSG.

general geometry better, especially for the vase data in Fig. 19. The vase has more complex loops; however, the detailed shape is not presented in the corresponding SSG in Fig. 19(b). Compared with the skeletons shown in Ning et al.,[13] the optimized skeletons in our paper contain more geometrical and topological information.

If we take the teapot as query input, the SSG of the teapot in Ning et al.[13] is displayed in the first column in Fig. 20(a) and the retrieval results would be those with the dotted colored boxes belonging to the same category as the query object (teapot) after using the similarity measurement. Moreover, it is difficult to distinguish the data in the third and the fourth columns. Compared to our results in Fig. 20(b), after the similarity measurement it retrieves two similar results, respectively, in the fourth and last columns.

### 7.5 *Limitation*

Our method can decompose objects depending on the number of patches that is determined by the feature points. When dealing with incomplete point cloud data, our method occasionally selects incorrect feature points, thus leading to incorrect topology.

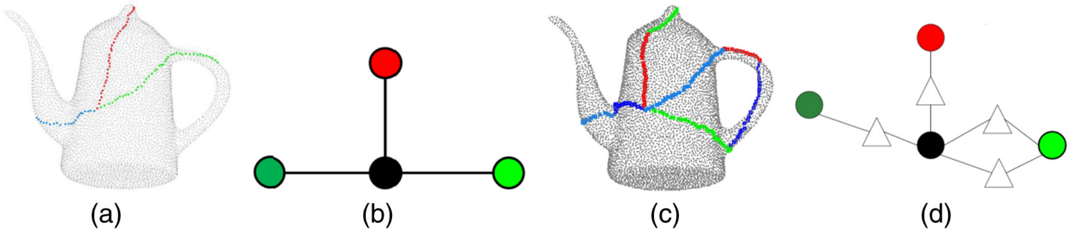


**Fig. 18** Comparison results on teapot data. (a) and (b) Skeleton and SSG in Ning et al.[13] (c) and (d) Our skeleton and SSG.

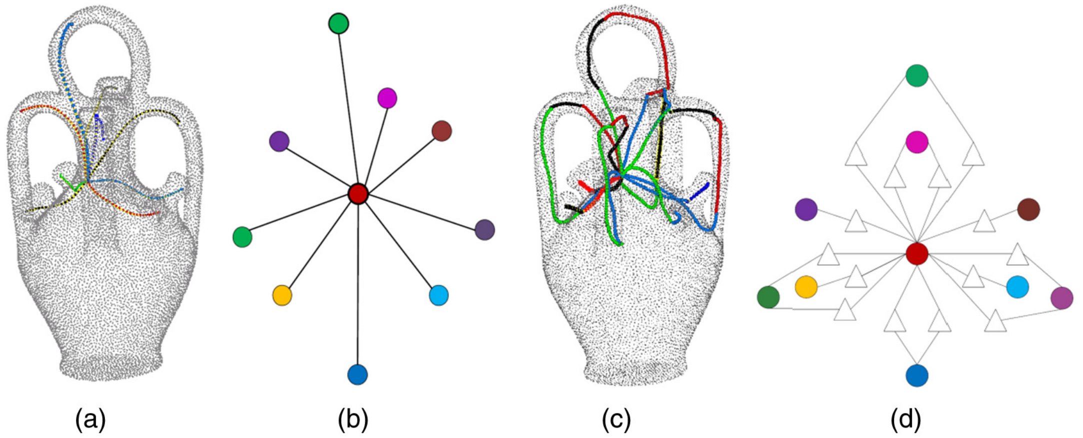


**Fig. 19** Comparison results on vase data. (a) and (b) Skeleton and SSG in Ning et al.[13] (c) and (d) Our skeleton and SSG.

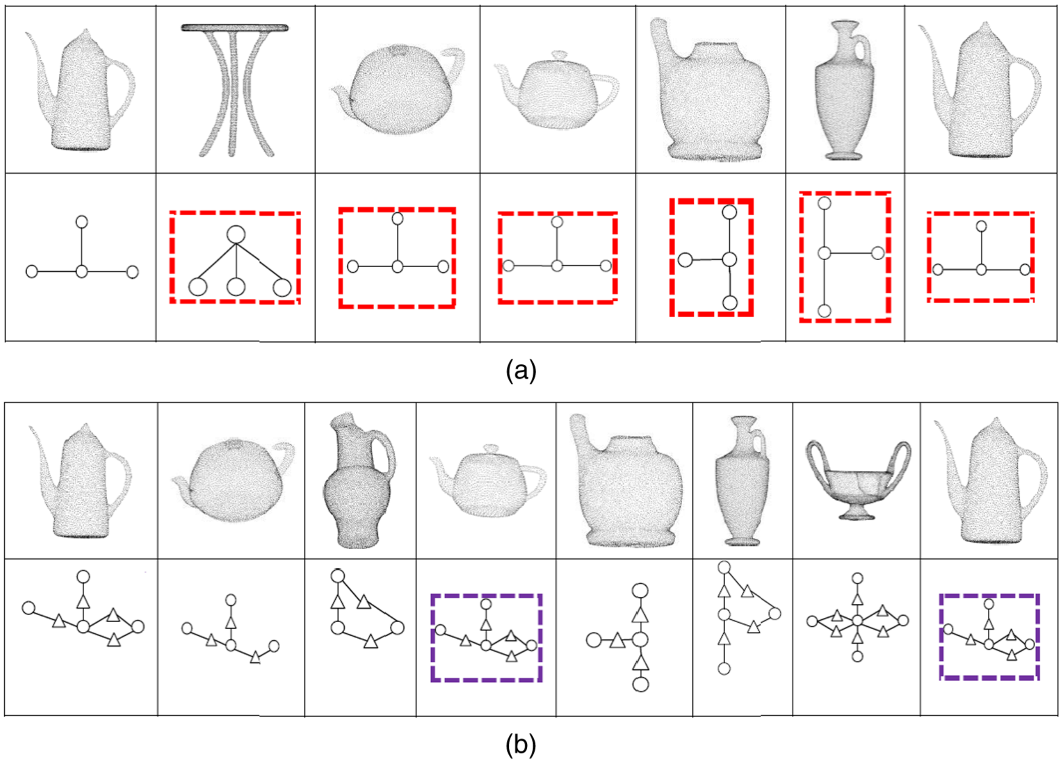


**Fig. 20** Retrieval results of teapot data based on SSG. (a) Results in reference.[13] (b) Our results.

Another limitation of our method is that the boundary surface between two neighboring regions could be smoothed and improved, providing a means of enhancing the shape decomposition results conversely.

## 8 Conclusion

In this paper, we propose a method to describe the shape semantic of different objects in point clouds. Our method is based on shape decomposition that produces components and structures of the object. The skeleton provides the topology of the object. Feature points, junction points, and the center point of the object are used to obtain a centralized skeleton to represent the topology. An SSG is generated to describe the shape of object and the similarity measurement provides an effective way to recognize different objects. Experimental results demonstrate the effectiveness and advantages of the SSG for understanding and recognition of objects with or without loops. Future research will concentrate on the improvement of the boundary surface between two neighboring regions to acquire a smooth and accurate boundary that can also be used to improve the shape decomposition results. In addition, more considerations should be given to the high-level semantics of an object when there are deformations (such as the tail of an animal, which may be straight, bent, or even curly).

### *Acknowledgments*

The work was supported in part by the National Natural Science Foundation of China under Nos. 61272284, 61302135, 61472319 and 61571439; in part by China

Postdoctoral Science Foundation under No. 2014M552469; in part by the Doctoral Fund of Ministry of Education of China under No. 20126118120022; in part by Shaanxi Postdoctoral Science Foundation under No. 434015014; in part by the Science and Technology Project of Xi'an under No. CXY1440(5).

## References

1. L. De Floriani, L. Papaleo, and N. Carissimi, "A Java 3D framework for inspecting and segmenting 3D models," in *Proc. of the 13th Int. Symp. on 3D Web Technology (Web3D '08)*, pp. 67–74, ACM, New York, NY (2008).
2. M. Attene et al., "Characterization of 3D shape parts for semantic annotation," *Comput.-Aided Des.* **41**, 756–763 (2009).
3. L. Papaleo and L. De Floriani, "Semantic-based segmentation and annotation of 3D models," in *Image Analysis and Processing–ICIAP 2009*, pp. 103–112 (2009).
4. L. Papaleo et al., "Towards a semantic web system for understanding real world representations," in *Proc. of the Tenth Int. Conf. on Computer Graphics and Artificial Intelligence* (2007).
5. R. De Figueiredo and N. Kehtarnavaz, "Model-based orientation-independent 3-D machine vision techniques," *IEEE Trans. Aerosp. Electron. Syst.* **24**(5), 597–607 (1988).
6. F. Stein and G. Medioni, "Structural indexing: efficient 3-D object recognition," *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(2), 125–145 (1992).
7. C. S. Chua and R. Jarvis, "Point signatures: a new representation for 3D object recognition," *Int. J. Comput. Vision* **25**(1), 63–85 (1997).
8. A. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(5), 433–449 (1999).
9. M. Hilaga et al., "Topology matching for fully automatic similarity estimation of 3D shapes," in *Proc. of SIGGRAPH*, pp. 203–212, ACM, New York, NY (2001).
10. S. Biasotti, "Topological techniques for shape understanding," in *5th Central European Seminar on Computer Graphics*, pp. 163–172 (2001).
11. T. K. Dey, J. Giesen, and S. Goswami, "Shape segmentation and matching with flow discretization," in *Proc. Workshop on Algorithms and Data Structure*, Vol. **2748**, pp. 25–36 (2003).
12. Y. Xiao, N. Werghi, and P. Siebert, "A topological approach for segmenting human body shape," in *Proc. of the 12th Int. Conf. on Image Analysis and Processing*, pp. 82–87 (2003).
13. X. Ning et al., "Shape decomposition and understanding of point cloud objects based on perceptual information," in *Proc. of the 9th ACM SIGGRAPH Conf. on Virtual-Reality Continuum and its Applications in Industry (VRCAI '10)*, pp. 199–206, ACM, New York, NY (2010).
14. D. Cohen-Steiner, P. Alliez, and M. Desbrun, "Variational shape approximation," *ACM Trans. Graph.* **23**, 905–914 (2004).
15. Z. Les, "Shape understanding: possible classes of shapes," *Int. J. Shape Model.* **7**, 75–109 (2001).
16. Z. Les, "Shape understanding system: understanding the thin object," *Comput. Graphics Forum* **26**, 951–970 (2002).
17. Z. Les and M. Les, "Shape understanding system: the visual reasoning process," *Int. J. Pattern Recognit. Artif. Intell.* **17**, 663–683 (2003).
18. M. Attene et al., "Computational methods for understanding 3D shapes," *Comput. Graphics* **30**(3), 323–333 (2006).
19. B. Bustos et al., "Feature-based similarity search in 3D object databases," *ACM Comput. Surv.* **37**, 345–387 (2005).
20. Y. Guo et al., "3D object recognition in cluttered scenes with local surface features: a survey," *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(11), 2270–2287 (2014).
21. Y. Guo et al., "A comprehensive performance evaluation of 3D local feature descriptors," *Int. J. Comput. Vision* **116**, 66–89 (2016).
22. O. Sidi et al., "Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering," *ACM Trans. on Graphics* **30**(6), 126 (2011).
23. S. Biasotti et al., "Describing shapes by geometrical-topological properties of real functions," *ACM Comput. Surv.* **40**, 12 (2008).
24. L. Čomić, L. De Floriani, and L. Papaleo, "Morse-Smale decompositions for modeling terrain knowledge," in *Proc. of the 2005 Int. Conf. on Spatial Information Theory*, A. G. Cohn and D. M. Mark, Eds., pp. 426–444, Springer-Verlag, Berlin, Heidelberg (2005).
25. D. Bespalov et al., "Scale-space representation of 3D models and topological matching," in *Proc. of the Eighth ACM Symp. on Solid Modeling and Applications (SM '03)*, pp. 208–215, ACM, New York, NY (2003).
26. T. Tony and S. Francis, "Augmented Reeb graphs for content-based retrieval of 3D mesh models," in *Int. Conf. on Shape Modeling and Applications*, pp. 157–166 (2004).
27. A. Hui and L. D. Floriani, "A two-level topological decomposition for non-manifold simplicial shapes," in *Proc. of the 2007 ACM Symp. on Solid and Physical Modeling*, pp. 355–360 (2007).
28. R. Schnabel et al., "Shape recognition in 3D point clouds," in *Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision*, Vol. **2**, pp. 4–7 (2007).
29. L. D. Floriani, L. Papaleo, and A. Hui, "TopMesh—a tool for extracting topological information from non-manifold objects," in *5th Int. Conf. on Computer Graphics Theory and Applications*, pp. 21–29 (2010).
30. T. K. Dey and J. Sun, "Defining and computing curve-skeletons with medial geodesic function," in *Proc. of the Fourth Eurographics Symp. on Geometry Process.*, pp. 143–152 (2006).
31. K.-C. O. Au et al., "Skeleton extraction by mesh contraction," *ACM Trans. Graph.* **27**(3), 44 (2008).
32. K.-C. O. Au et al., "Electors voting for fast automatic shape correspondence," *Comput. Graphics Forum* **29**(2), 645–654 (2010).
33. J. Cao et al., "Point cloud skeletons via Laplacian-based contraction," in *Proc., Int. Conf. on Shape Modeling and Applications (SMI '10)*, pp. 187–197 (2010).
34. A. Tagliasacchi et al., "Mean curvature skeletons," *Comput. Graphics Forum* **31**, 1735–1744 (2012).
35. C. Michael and K. Piyush, "Fast construction of k-nearest neighbor graphs for point clouds," *IEEE Trans. Visual Comput. Graphics* **16**, 599–608 (2010).
36. F. Wu et al., "Skeleton extraction of 3D objects with visible repulsive force," in *Eurographics Symp. on Geometry Processing* (2003).
37. R. Osada et al., "Shape distributions," *ACM Trans. Graph.* **21**, 807–832 (2002).
38. X. Chen, A. Golovinskiy, and T. Funkhouser, "A benchmark for 3D mesh segmentation," *ACM Trans. Graphics* **28**, 73 (2009).
39. Y. Ma, S. Worrall, and A. M. Kondoz, "3D point segmentation with critical point and fuzzy clustering," in *Proc. the 4th IET Conf. on Visual information Engineering*, London (2007).
40. M. Richtsfeld and M. Vincze, "Point cloud segmentation based on radial reflection," in *Computer Analysis of Images and Patterns*, pp. 955–962, Springer-Verlag, Berlin, Heidelberg (2009).
41. I. Yamazaki et al., "Segmenting point sets," in *IEEE Int. Conf. on Shape Modeling and Applications*, p. 6 (2006).
42. Z. Shu et al., "Unsupervised 3D shape segmentation and co-segmentation via deep learning," *Comput. Aided Geom. Des.* **43**, 39–52 (2016).
43. O. V. Kaick et al., "Shape segmentation by approximate convexity analysis," *ACM Trans. Graph.* **34**, 4 (2014).
44. A. Golovinskiy and T. Funkhouser, "Randomized cuts for 3D mesh analysis," *ACM Trans. Graph.* **27**, 145 (2008).
45. A. Golovinskiy and T. Funkhouser, "Consistent segmentation of 3D models," *Comput. Graphics* **33**, 262–269 (2009).
46. Y.-K. Lai et al., "Fast mesh segmentation using random walks," in *Proc. of the 2008 ACM Symp. on Solid and Physical Modeling (SPM '08)*, pp. 183–191, ACM, New York, NY (2008).
47. S. Shlafman, A. Tal, and S. Katz, "Metamorphosis of polyhedral surfaces using decomposition," *Comput. Graphics Forum* **21**, 219–228 (2002).

**Xiaojuan Ning** received her PhD in 2011. She is an associate professor at Xi'an University of Technology. Meanwhile, she has cooperated with the Lab of LIAMA and National Laboratory of Pattern Recognition at Institute of Automation, CAS. Her current research interests include scene modeling and shape analysis.

**Yinghui Wang** received his PhD degree from Northwest University, Xi'an, China, in 2002. From 2003 to 2005, he was a postdoctoral fellow at Peking University, Beijing, China. Now he is a professor at the Institute of Computer Science and Engineering, Xi'an University of Technology, China. His research interests include image analysis and pattern recognition.

**Weiliang Meng** is an assistant professor in National Laboratory of Pattern Recognition (NLPR) at Institute of Automation, Chinese Academy of Sciences. He received the BE degree in computer science from Civil Aviation University of China in 2003, an MSc degree in computer application from Tianjin University in 2006 and a PhD degree in computer application from State Key Laboratory of Computer Science at Institute of Software, Chinese Academy of Sciences, in 2010. His research interests include geometry modeling, image-based modeling and rendering, and augmented reality.

**Xiaopeng Zhang** is a professor in the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences. He received his PhD degree in computer application from the Institute of Software, Chinese Academy of Sciences. His research interests include 3D-shape analysis, complex modeling and rendering, and augmented reality.