IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 12, 2019, accepted November 27, 2019, date of publication December 2, 2019, date of current version December 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2957070

From 2D to 3D: Component Description for Partial Matching of Point Clouds

YUHE ZHANG, XIAONING LIU, CHUNHUI LI, JIABEI HU, GUOHUA GENG, AND SHUNLI ZHANG
School of Information Science and Technology, Northwest University, Xi'an 710127, China

Corresponding author: Shunli Zhang (slzhang@nwu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61802311, Grant 61772421, Grant 61731015, and Grant 61602380, in part by the National Key Research and Development Program of China under Grant 2017YFF0107201, in part by the Key Research and Development Program of Shaanxi Province under Grant 2019SF-272, and in part by the Natural Science Foundation of Shaanxi Provincial Department of Education under Grant 18JK0795.

ABSTRACT We present a method to compute the descriptor of components of point clouds, therefore, a novel component-oriented partial matching of point clouds is achieved based on the component descriptor. We observe that 3D components can be constructed by stacking 2D shapes using certain criteria so that the centers of the 2D shapes form a curve called a skeletal curve that is the trajectory of the 2D shapes. In addition, the scaling factors of the 2D shapes also impact the shape of the 3D components. Motivated by these observations, the computation of the component descriptor that is termed 2to3SSC (from 2D to 3D: 2D Shape and Skeletal Curve) is formulated as a 2D shape and skeletal curve extraction problem, and the component-oriented partial matching of the point clouds is based on the dissimilarity measure of 2to3SSCs of the components. Furthermore, for the 2D shape matching, which is crucial to the matching of the components, we present a novel 2D shape descriptor called VDTL (Vertical Distances to the Tangent Line). The proposed method outperforms previously proposed methods because it simultaneously encodes the local and global features of the components as opposed to only encoding the local or partial features as in previous studies. Finally, the effectiveness and performance of 2to3SSCs are compared with those of state-of-the-art feature description and matching methods for different point cloud datasets. Further, the benefits and the applicability of the proposed method are demonstrated; favorable results are obtained for real-world point clouds of Terracotta fragments.

INDEX TERMS Computer graphics, point clouds, partial matching, 2D shape matching.

I. INTRODUCTION

Feature extraction, description, and matching are the core and prerequisite of most point cloud processing techniques, such as point cloud registration [1], line drawings generated from point clouds [2], 3D object retrieval [3], 3D object partitioning [4], and 3D object reconstruction [5].

It is therefore not surprising that numerous studies have reported on techniques for addressing the problem of feature extraction, description, and matching of point clouds [3], [6]. However, the main differences among these techniques are the geometric scale at which the analysis is performed and the matching rules (or similarity measures) for the corresponding feature descriptors. Thus, the features referenced in previous studies can be roughly classified into three categories related to the geometric scale: micro-features, meso-features, and macro-features, as shown in Fig. 1.

Geometric operators at a fine scale (referred to as micro-features in this paper) concentrate on extracting small structures and capturing details that describe the gradients of the point clouds, as shown in Fig.1 (b). The extracted features are scattered points or line segments. These methods are often applied to (i) the generation of line drawings in non-photorealistic rendering (NPR) [7] since the line drawings convey the shape of the models; (ii) the reassembly of fragments based on the matching of feature points [8] or feature lines [5]; (iii) as a pre-requisite step for generating the meso-feature descriptors [6]. Micro-features extraction approaches typically estimate geometric invariants coupled with multiscale frameworks (the size of a

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

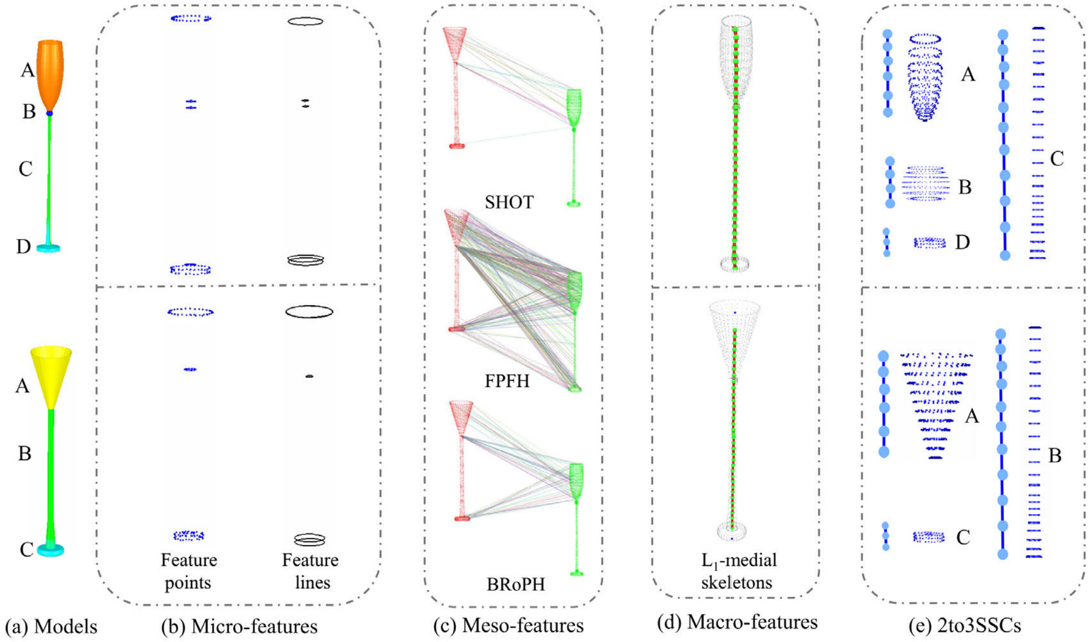


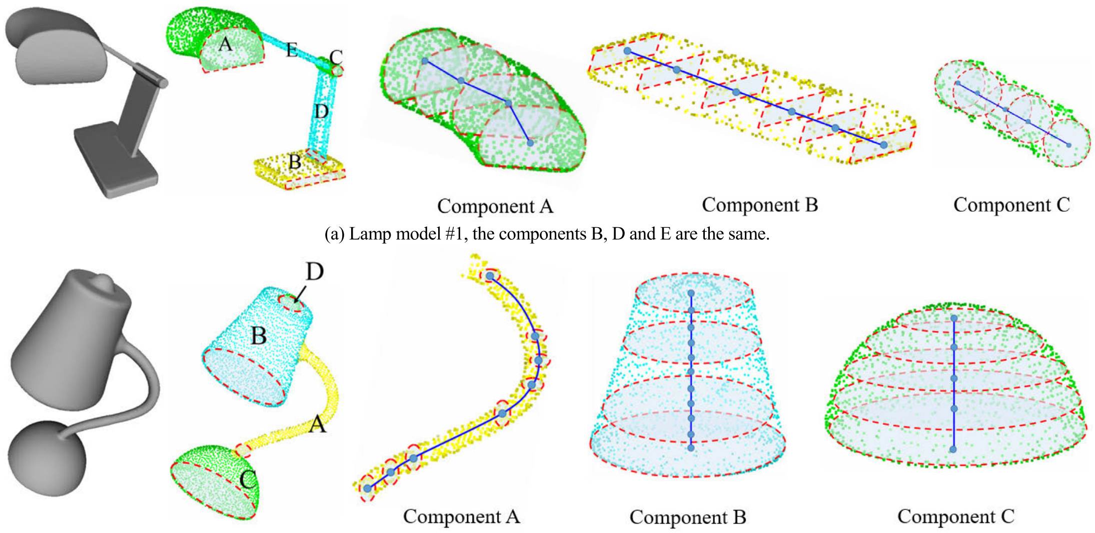
FIGURE 1. For the two goblets (the components are marked as A, B, C, and D) in (a), the feature points and feature lines (micro-features) in (b) are extracted using the methods described in [2] and PCA method; the matching pairs of the meso-features in (c) are represented by SHOT [29], FPFH [28] and BRoPH [6] respectively; the L_1- medial skeletons in (d) are computed using the method described in [45], and the 2to3SSCs in (e) are computed by the method proposed in this study. The matching pairs of components that are obtained by the similarity measurement of 2to3SSCs are marked in the same colors in (a).

neighborhood is often used as the scale parameter, which differs from the term "scale" used in this paper) and they commonly consist of two phases: first, the differential geometric invariants or integrated invariants of each point in the point clouds are determined; second, the feature points are distinguished by analyzing the invariant gradients in the neighborhood of each point using clustering methods [9], statistical approaches [10], and local surface- or curve-fitting methods [11].

Geometric operators at the meso-scale (referred to as meso-features in this paper) describe the partial shape of a 3D model from a specific viewing angle. The goal of these approaches is to extract local representations of point clouds rather than describing the local gradients of the point clouds [6]. Thus these features are considered meso-features. The approaches for the meso-feature description and their matching via feature descriptors to find true point correspondences typically consist of continuously rotating the point clouds and projecting the local surfaces onto a 2D plane; subsequently, the feature description is performed by generating feature descriptors from a series of 2D image patches. These approaches are mostly performed for 3D surface matching [6] that is achieved by using dissimilarity measures of the generated feature descriptors, as shown in Fig.1 (c).

Geometric operators at the macro scale (referred to as macro-features in this paper) describe the global shape of a 3D model and a skeleton is commonly used, as shown in Fig. 1 (d). A skeleton is a geometry-based shape representation and consists of the centers of the maximal spheres inside the shape surface, wherein the radii of the maximal spheres represent the thickness of the shape [12]. Skeleton-based shape matching and 3D object retrieval have been reported in a number of studies with the objective to measure the dissimilarity of two shapes based on the geometric dissimilarity and topological dissimilarity of the skeletons [3].

In most previous studies, feature points and feature lines (micro-features) only describe the gradients of the surfaces of the point clouds (Fig. 1 (b)), which provide little information on the entire shape of the components or point clouds. For meso-features, it was assumed that the spatial and geometric information of the point cloud can be encoded in feature descriptors; therefore, these approaches work only for feature descriptor matching to find the true matching pairs of points (Fig. 1 (c)) and the comparison of the meso-features does not validate the similarity of two components or point clouds. The skeleton-based shape matching (macro-features) for point clouds mainly focuses on the structure of each part of the point clouds so that point clouds with different composition and structure are distinguished but two different point clouds with the same skeletons cannot be distinguished, for example, the two goblets shown in Fig. 1 (d). However, three-dimensional (3D) objects with manually-labeled tags are commonly used in machine learning methods but the generation of tags is time-consuming, making it difficult to use for some practical applications. Reference [13] presented a framework which automatically finds part analogies among 3D objects, however, the analogous parts may not be similar as a whole. Hence, to the best of our knowledge,



(a) Lamp model #1, the components B, D and E are the same.

(b) Lamp model #2, the components C and D are the same.

FIGURE 2. Two different lamp models whose components are marked in different colors. The red dashed lines are the 2D shapes that are stacked to construct the component, the blue curves are the skeletal curves, and the blue solid dots are the equidistant sample points on the skeletal curves.

the more challenging partial matching of point clouds, in which the goal is the component-oriented partial matching of point clouds, has not been reported in previous studies, whereas the partial matching in 2D cases has gained increased attention[14].

In this study, rather than considering feature points, feature lines, surface patches, or skeletons, we survey the analytical techniques that focus on the component parts of point clouds, which are referred to as components. The analysis of the feature is also performed at the macro-scale because this describes the entire shape of the components by simultaneously encoding the local and global features of the components as opposed to only encoding the local or partial features as in previous studies, as shown in Fig. 1 (e).

Our method is inspired by the observation that most 3D shapes can be created by stacking the 2D shapes using certain criteria. For example, as shown in Fig. 2 (a), the components in the different colors are created by simply stacking different 2D shapes. In Fig. 2 (b), the components in different colors are all created by stacking circles; however, the components are different. The rod component in yellow is created by stacking circles of the same size; the tapered component in cyan and the half-sphere components in green consist of circles that are gradually scaled down but their shapes are also different. In addition, each 2D shape that is stacked to construct the 3D component has a unique center and the centers of all 2D shapes form a 3D curve, which represents the trajectory of the 2D shapes. We call this curve a skeletal curve in this paper, as shown in Fig. 2.

Hence, we formulate our approach as a 2D shape and skeletal curve extraction problem, in which the 2D shape and the skeletal curve are used to construct the 3D component descriptor 2to3SSC (as shown in Fig. 1(e)) such that stacking the 2D shapes along the skeletal curve directly yields the component. Consequently, the 3D component matching is performed by the dissimilarity measure of the 2to3SSCs. The proposed component description for the partial matching of point clouds can be used in many subsequent processing techniques, such as 3D object retrieval, 3D object recognition, and 3D object classification.

Our main contributions are as follows: (i) we propose the novel descriptor 2to3SSC for the components of point clouds that is generated based on the perception of 3D shapes originating from 2D shapes. The computation of the descriptor is conceptually simple and easy to implement; (ii) we propose a new 2D shape descriptor VDTL (Vertical Distances to the Tangent Line) which plays an important role in component matching; (iii) from a practical aspect, this method represents a novel component-oriented partial matching of point clouds with wide applicability.

II. PREVIOUS WORK

We review the related literature on feature extraction, description, and matching. As shown in the following parts of this section, the existing methods are unable to directly cope with components of point clouds.

A. *MICRO-FEATURES*

Most micro-feature extraction methods cope with differential invariants or integral invariants and their computations have been reported in a large number of publications. For a recent survey of normal and curvature estimation, we refer the reader to [15] and for the computation of integral invariants, the reader is referred to [16] for more details.

The micro-feature extraction methods can be classified into four categories. One category is related to local feature detection operators to extract feature points. The common techniques include covariance analysis [17], multi-scale analysis of the neighborhoods [18], curvedness analysis [19], and tensor voting methods [20].

Another category of micro-feature extraction methods is based on the observation that features are located near potential feature lines. For example, reference [11] used the robust moving least squares (RMLS) method to locally fit surfaces. Reference [21] estimated the principal curvatures and normal directions of the underlying surface and then iteratively computed covariance matrices with varying neighborhoods.

Another category considers that the features are located near intersections between the potential piecewise smooth surfaces. For example, reference [10] combined a region-growing strategy with a probability to determine whether a point was a feature point. Reference [9] used Gauss map clustering in local neighborhoods to discard non-feature points and reference [2] used a Poisson region-growing (PRG) method to distinguish the feature points from non-feature points. Based on such observation, there are many methods to extract micro-features by segmenting the 3D model into different surfaces as well [22].

The last category of micro-feature extraction methods is focused on the direct extraction of line segments, which is generally performed on specific datasets, for example, building models that contain many straight lines [23], and reference [24] directly detected the feature curves.

By examining the application goals, we can classify the micro-feature matching into: (i) matching of features that are similar to each other, which assumes that the features have nearly the same differential or integral invariants and (ii) complementary matching, in which the features share a common contact surface. In most micro-feature matching methods, distance-based methods are utilized for the matching; references [1], [5] and the references therein provide more details.

B. MESO-FEATURES

The goal of a meso-feature descriptor is to extract the local representation at a certain point, such as a micro-feature point. Among all existing feature descriptors, histogram-based methods are most frequently investigated. Commonly encoded information in the descriptors includes normal differences, point density, and depth. Point signature was proposed to describe the structural neighborhood of a point instead of just using the 3D coordinates of the point [25]. The most classic 3D local feature descriptor is a spin image (SI) that has been considered as a benchmark for the evaluation of 3D local feature descriptors [26]. It uses the normal of the point as a local reference frame (LRF) and then describes its neighboring points with the in-plane and out-plane distances of the neighboring points. Reference [27] combined a spin image (SI) and normal-based signatures to develop a new descriptor and computed the similarities between the input 3D range images by matching the descriptors with a pyramid kernel function. The fast point feature histogram (FPFH) is calculated as the weighted sum of the simplified point feature histogram (SPFH) that is a measure of the normal difference between the point and its neighbors [28]. The signature of histograms of orientations (SHOT) divides the neighbors into different volumes using the LRF and then the normal angles between the feature point and all points falling in each volume are accumulated into corresponding sub-histograms, which are concatenated to create the SHOT [29].

Rather than describing in floating representation as mentioned above, which is generally computationally complex and memory-consuming, many feature descriptors are described in a binary format. 3D-BRIEF [30], 3D-ORB [31], and 3D LBP [32] are first encoded in a binary pattern, and are generated by extending BRIEF, ORB, and the LBP to 3D dimensions. B-SHOT is also generated by applying a binary quantization to SHOT [33].

Reference [34] used projection techniques and encoded the spatial and geometric information in the projected 2D image patches. Reference [35] also used projection techniques consisting of a rotational contour signature (RCS) method for the descriptions of the 3D local shape. Similar, triple orthogonal local depth images (TOLDI) and the binary rotational projection histogram (BRoPH) are computed by concatenating depth images generated from three orthogonal view planes in the LRF; however, TOLDI is a histogram-based descriptor [36], whereas BRoPH encodes information using an improved version of the LBP [6].

C. MACRO-FEATURES

A skeleton is a thin centered structure which jointly describes the topology and the geometry of 3D objects [37], thus it became a powerful tool for shape analysis and matching [12], [13], [38]. Recent approximation methods of 3D medial axes mostly fall into two categories: voxel-based methods based on a shape represented by a union of voxels and sampling-based methods, which consider point samples at the shape boundary and are often based on the Voronoi diagram of these samples. Since our method is little related to skeleton extraction methods, we only list some recent references.

Reference [39] extracted skeletons using an extended Reeb graph that was originally used for triangle meshes and a discrete Reeb graph. References [40] and [41] used a voxel shape and identifies a subset of the voxels that shared similar properties with the medial axis, such as being thin, centered, and preserving both the shape's topology and shape components. Some other voxel-based methods are guided by a non-Euclidean distance metric that can be locally evaluated, for example, the <3,4,5> distance [42]. Reference [12] observed that the medial axis of a voxel shape could be easily and faithfully approximated by the interior Voronoi diagram of the boundary vertices.

The sampling-based methods are more efficient and scalable than the voxel-based methods. References [43] and [44] located the points of maximal balls given the normals at the sample points. Reference [45] introduced an L_1-medial skeleton as a curve representation for 3D point clouds.

D. MACHINE LEARNING-BASED METHODS

The increasingly popular machine learning-based methods are often applied for 3D object segmentation and classification [46]. Unsupervised techniques utilize shape similarities and geometric features to discern objects in point clouds but provide no significant output control [47], [48]. Existing scene graphs are exploited in [49] but they are sparse and not always useful.

Supervised methods use shapes with manually-labeled regions to train a model and new point clouds with similar semantic regions can then be segmented. Recent methods have exploited deep neural networks with diverse styles, such as using multiple discriminative auto-encoders to form a shape descriptor [50], projective convolutional networks [51], synchronized spectral convolutional neural networks (CNN) [52], and octree-based CNN [53]. However, collecting user-prescribed labels is the main bottleneck for supervised methods because it is time-consuming.

Weakly or semi-supervised strategies represent a compromise between supervised and unsupervised methods [54], [55]. A recent study [56] proposed a novel method for segmenting 3D objects, which strongly correlate with user-prescribed tags.

E. 2D SHAPE MATCHING METHODS

Since the 2D shape descriptor is not the main focus of our work, only a few recent references are discussed in this section.

Most previous studies on 2D shape matching have assumed that the shape can be represented by a closed contour that is usually defined by a sequence of sample points. Thus, shape matching can be achieved by determining the one-to-one point correspondences between the points of two contours. Curvatures are a popular local invariant feature descriptor [57] and are less descriptive for the representation of the features. It is common to use the shape context (SC) [58] that encodes the local distribution of the contour points in the histograms. Many variants were reported to perform better than many other descriptors. For example, in the inner-distance shape context (IDSC), the Euclidean distance and relative angles are replaced by the ‘‘inner distances’’ and ‘‘inner angles’’ [59], and an independent-IDSC feature descriptor was proposed for plant identification [60]. A self-containing shape descriptor for open and closed contours was proposed for part-to-part partial shape matching [61]. The triangular centroid distances (TCD) descriptor that encodes the triangular centroid distances for shape representation was proposed for 2D non-rigid partial shape matching [14].

III. OVERVIEW

The original processing target of our method is an unorganized point cloud. However, the descriptor presented in this paper deals with the components of point clouds, thus the preliminary step first segments the point cloud into component parts (Fig. 2) using the method described in [4] since segmentation is not the focus of this study. Thus, the components are the actual inputs to our method, which computes the component descriptor 2to3SSC.

In some cases, the skeletal curve of a component is the skeleton of the component, as shown in component A in Fig. 2 (b). However, in many cases, the skeletal curve is not the skeleton of the component; examples are the components B and C in Fig.2 (b). Therefore, we first compute the skeletal curve of the component by segmenting the components, and then the skeletal curves are extracted.

Given the skeletal curve of each component, we then extract the 2D shapes that are stacked to construct the 3D components. The extraction of the 2D shapes is based on the observation that the planes where the 2D shapes are located are orthogonal to the local skeletal curve segments, such as the ones shown in Fig.2. By continuously traversing the 3D component along the skeletal curve, all of the 2D shapes can be extracted.

Based on the observation that most 3D components can be created by stacking the 2D shapes, the 2D shapes and the skeletal curve of the components that are obtained in the preceding two steps describe the shape of the 3D component. Additionally, we found that the scales of the 2D shapes also impact the shape of the 3D component, as shown in Fig. 2. Thus, the 2D shapes and the skeletal curve, along with the scaling factors of 2D shapes, build the component descriptor 2to3SSCs.

Given the 2to3SSC descriptors of two components, the component matching is performed based on the similarity measure of the 2to3SSCs, in which three different constraints are considered.

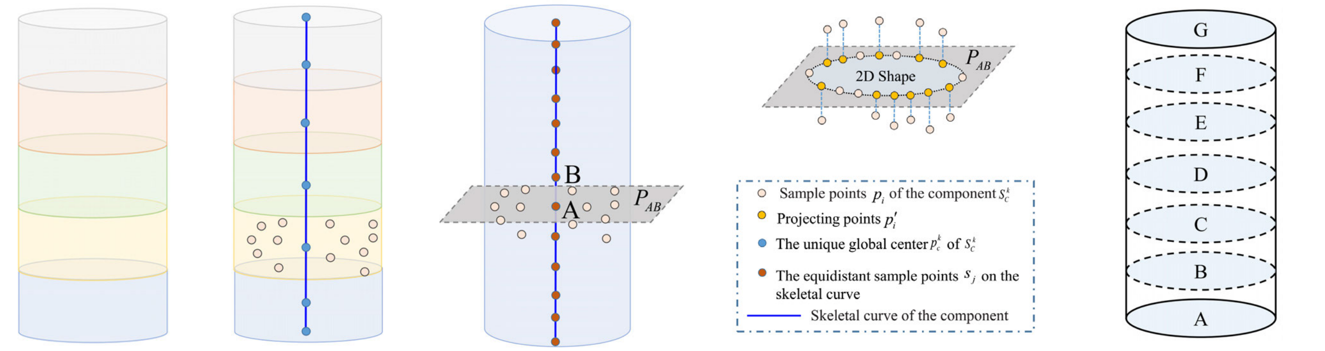
Therefore, the proposed method is summarized as follows:
Step 1: 2to3SSC descriptor generation. (Section 4)
Step 1.1: Component segmentation.
Step 1.2: Skeletal curve generation.
Step 1.3: 2D shape extraction.
Step 1.4: Component descriptor 2to3SSC construction.
Step 2: Component matching. (Section 5)
Step 2.1: 2D shape matching.
Step 2.2: Skeletal curve matching.
Step 2.3: Scaling factors matching.

IV. 2TO3SSC DESCRIPTOR GENERATION

In the preliminary step, the given point cloud is segmented into components. Therefore, our approach starts with the components rather than the entire point clouds. In this section, we concentrate on the generation of the 2to3SSC descriptor for the components.

A. COMPONENT SEGMENTATION

When the skeletal curve of a component is the skeleton of the component, we use the skeleton computation method described in [45] to extract the skeletal curve of a component. If this is not the case, we compute the skeletal curve of the component.


B
A
P_{AB}
2D Shape
P_{AB}
○ Sample points p_i of the component S_C^k
● Projecting points p_i'
● The unique global center p_c^k of S_C^k
● The equidistant sample points s_j on the skeletal curve
Skeletal curve of the component
G
F
E
D
C
B
A

(a) The segments of the component.

(b)The skeletal curve generation.

(c)The tangent plane P_{AB}.

(d)Points are projected onto P_{AB}.

(e)The extracted 2D shapes.

FIGURE 3. The illustration of skeletal curve generation and 2D shape extraction.

We found that the method of computing the L_1-medial skeleton [45] cannot cope with some components (e.g., components A and B in Fig.2 (a) and components B and C in Fig.2 (b)) because it uses the local ball neighborhoods as the local neighborhood for the L_1-medial skeleton computing, the ball that covers the local supporting neighborhoods may cover the entire component. Thus we present a variant version of the L_1-medial skeleton method, which first segments the components and then compute the skeletal curve.

First, we detect the feature points in the component by using the PRG method described in [2] and the PCA (Principal Component Analysis) method. According to the region information, the feature points sharing the same surface are considered border points of the component. Next, the k nearest neighbors (kNN) of the border points form a new segment of the component. The points are regarded as new border points if at least one of its kNN is not part of the current segment and another new segment is formed by the kNN of these border points. The partition process is repeated with new border points and their neighbors until no new border point can be selected. Consequently, the component can be segmented into various segments, as shown in Fig.3 (a).

B. SKELETAL CURVE GENERATION

For each segment of the component, we compute the L_1-medial point p_c^i that is the unique global center of the point sets of the segment. For that purpose, let $p_i = (x_i, y_i, z_i)$ $(i = 1, 2, \ldots, N)$ denote the sample points of the segment S_C^k, here i is the index of the sample points, k is the index of the segments, x_i, y_i and z_i are the coordinates of points p_i on S_C^k, C is the index of the given component, and N is the number of the points of S_C^k. The unique global center p_c^k of S_C^k is calculated using Equation (1) [62].

$$p_C^k = \underset{x}{\arg\min} \left\{ \sum_{p_i \in S_C^i} \|p_i - x\| \right\} \tag{1}$$

Given the center points p_c^k, we use the curve reconstruction method in [2] to obtain the skeletal curve, as shown in Fig.3 (b).

C. 2D SHAPE EXTRACTION

Given the component C and its skeletal curve, we then extract the 2D shapes that are stacked to construct the 3D components. Based on the observation that the 2D shapes are on the plane that is orthogonal to the local skeletal curve segments, we begin by creating such a plane.

For that purpose, let $p_i = (x_i, y_i, z_i)$ $(i = 1, 2, \ldots, N)$ denote the sample points of the component C, here, i is the index of the sample points, x_i, y_i and z_i are the coordinates of p_i, and N is the number of the points of C. s_j are the equidistant sample points on the skeletal curve, which divides the skeletal curve into line segments of the same length and determines the location of the extracted 2D shapes. Here, j is the index of the sample points.

The illustration of the 2D shape extraction is shown in Fig.3(c-e). First, for each line segment AB of the skeletal curve (A and B are two adjacent sampling points on skeletal curve), we compute its vertical plane P_{AB} by using a vector dot product, where the coordinates of the points on the skeletal curves are known, as shown in Fig. 3(c). Then, we project the points p_i that are in the nearest segments onto the tangent plane P_{AB}, yielding points $p_i' = \left(x_i', y_i'\right)$ that constitute the 2D shapes, as shown in Fig. 3(d). However, post-processing must be applied to refine the 2D shapes that may contain outliers and noisy points due to the non-uniform sampling of the point clouds. Therefore, as the last step, we also use the curve reconstruction method described in [2] to obtain the refined 2D shapes. We trace the skeletal curve by traversing all the line segments to extract all the 2D shapes, as shown in Fig.3 (e).

D. COMPONENT DESCRIPTOR 2TO3SSC

The skeletal curve and the 2D shapes describe the shape of the component. Therefore they are considered the two elements of the component descriptor 2to3SSC. Additionally,

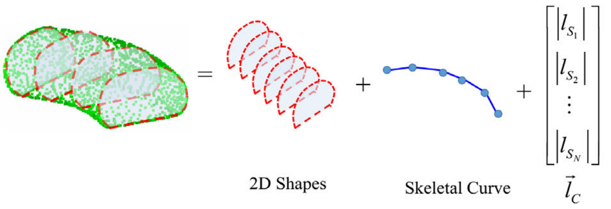

=
+
+
$\begin{bmatrix} |l_{S_1}| \\ |l_{S_2}| \\ \vdots \\ |l_{S_N}| \end{bmatrix}$
2D Shapes
Skeletal Curve
$\vec{l}_C$

FIGURE 4. The illustration of 2to3SSC of the component A of the lamp model #1.

the scales of the 2D shapes also have a large impact on the shape of the component (for example, the components B and C in Fig.2 (b)). Because scaling is a rigid transformation, we compute the longest line l_{S_i} connecting any two points of the 2D shape to represent the scale of the 2D shape; here i is the index of the 2D shapes.

Hence, the 2D shapes, the skeletal curve, and the vector $\vec{l}_C$ containing the length of l_{S_i} of all the 2D shapes construct the component descriptor 2to3SSC, as shown in Fig. 4; N is the number of the 2D shapes.

V. COMPONENT MATCHING

In this section, we will introduce how to measure the similarity between two 2to3SSC descriptors, which can validate the similarity of the components. Let 2to3SSC(A) and 2to3SSC(B) denote the 2to3SSC descriptors that are extracted from the components A and B respectively. The similarity between them is measured by three constraints: 2D shape matching, skeletal curve matching, and scaling factor matching.

A. CONSTRAINT 1: 2D SHAPE MATCHING

If two components match, the 2D shapes that are stacked to construct the components must be matching. Therefore, it is necessary to match the 2D shapes, and for this purpose, we propose a novel descriptor, namely VDTL. Let $p_i = (x_i, y_i)$ $(i = 1, 2, \ldots, N)$ denote the sequence of equidistant sample points on the outer contour of a 2D shape S; here, i is the index of the sample points, x_i and y_i are the coordinates of the point p_i, and N is the number of the points on the contour.

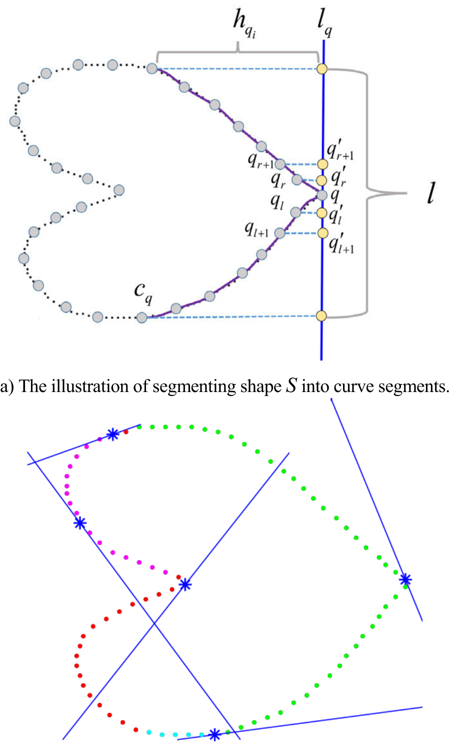
First, we segment the contours into curve segments. There are two levels of iteration in the segmentation step. The outer iteration is called the benchmark point selection iteration and the inner iteration is called the region-growing iteration.

1) BENCHMARK POINT SELECTION

A point with the largest curvature that is not part of any curve segment is taken as the benchmark point. For the computation of the curvature of the points, we use the method described in [63].

2) REGION GROWING

Each region-growing process begins at the selected benchmark point q and the nearby neighbors q_{i+1} and q_{i-1} in opposite directions are assigned to the curve c_q that is generated from q if q_{i+1} and q_{i-1} are not part of any curve segments. We then treat q_{i+1} and q_{i-1} as seed points, namely q_l and q_r and then perform region growing from these seed points to nearby candidates in the local neighborhood. The points q_{l+1} and q_{r+1} that are the nearby neighbors of q_l and q_r are considered the new endpoints of c_q that have to satisfy the condition:



(a) The illustration of segmenting shape S into curve segments.

(b) The segmenting result of shape S

FIGURE 5. The illustration of the computation of VDTL.

$$\begin{cases} \left\| q'_{l+1} - q \right\|_2 > \left\| q'_l - q \right\|_2 \\ \left\| q'_{r+1} - q \right\|_2 > \left\| q'_r - q \right\|_2 \end{cases} \tag{2}$$

where q'_{l+1}, q'_l, q'_{r+1}, q'_r are obtained by projecting q_{l+1}, q_l, q_{r+1}, q_r onto the line l_q defined by q and its tangent vector. Equation (2) ensures that the projected points on l_q do not overlap. Region growing stops when there is no additional q_{l+1} and q_{r+1} satisfies the above criterion. The procedure is repeated with q_l, q_r (the endpoints of the curve segments) and q_{l+1}, q_{r+1} until the region growing stops, as shown in Fig. 5(a).

When the growth iteration fails to detect any new points, the growth of the current curve segment is terminated. Subsequently, a new benchmark point (with the largest curvature among the remaining points) is selected to generate another curve segment. When there are no new benchmark points, the algorithm is terminated. Finally, the shape S is divided into different curve segments, as shown in Fig.5 (b); the points marked with an asterisk are the benchmark points and the curve segments are shown in different colors.

Next, we compute the descriptor of each contour point p_i of the shape S. For each point p_i, we can compute a 1D feature

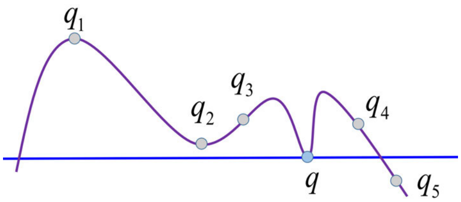


FIGURE 6. Four types of points on 2D curves.

TABLE 1. α, θ^+ and θ^* of points in Fig. 6.

Points	α	θ^+, and θ^*
q_1	$\alpha > 0$	$\theta^+ > 0$ and $\theta^* > 0$
q_2	$\alpha > 0$	$\theta^+ < 0$ and $\theta^* > 0$
q_3	$\alpha > 0$	$\theta^+ > 0$ or $\theta^+ < 0$ and $\theta^* < 0$
q_4	$\alpha > 0$	$\theta^+ > 0$ or $\theta^+ < 0$ and $\theta^* < 0$
q_5	$\alpha < 0$	$\theta^+ > 0$ or $\theta^+ < 0$ and $\theta^* < 0$

vector containing 6 different values:

$$[\alpha, \nabla_L^i, \nabla_R^i, \gamma, \theta^+, \theta^*] \tag{3}$$

where, α denotes the location of p_i, where 0 and 1 denote if p_i above or below the line l_q respectively; ∇_L^i and ∇_R^i are the left and right gradients of p_i, as computed by the following equation:

$$\begin{cases} \nabla_L^i = \sum\limits_{j=1}^{k} \dfrac{h_{p_i} - h_{p_{i-j}}}{\left\|p_i' - p_{i-j}'\right\|_2} \cdot \dfrac{1}{\sqrt{2\pi}} \exp \dfrac{\left\|p_i' - p_{i-j}'\right\|_2}{l} \\ \nabla_R^i = \sum\limits_{j=1}^{k} \dfrac{h_{p_i} - h_{p_{i+j}}}{\left\|p_i' - p_{i+j}'\right\|_2} \cdot \dfrac{1}{\sqrt{2\pi}} \exp \dfrac{\left\|p_i' - p_{i+j}'\right\|_2}{l} \end{cases} \tag{4}$$

where $h_x = \left\|x - x'\right\|_2$, and x' is yielded by projecting x onto l_q; k is the number of neighbors; j is the index; l is the length of the straight line yielded by projecting the points of c_q onto l_q; ∇_L^i and ∇_R^i describe the gradients of the points. We normalize h_{p_i} by dividing by l, resulting in $\gamma = {}^{h_{p_i}}/_{l}$; θ^+ and θ^* are the positive and negative signs of ∇_L^i plus ∇_R^i and ∇_L^i multiplied by ∇_R^i; this classifies the points into four classes. Fig. 6 and Table 1 provide an illustration.

Given the feature vector of each point, we can obtain the descriptor of shape S, which is expressed as follows:

$$\begin{aligned} &VDTL\,(S) \\ &= (VDTL\,(C_1)\,, VDTL\,(C_2)\,, \ldots, VDTL\,(C_n)) \\ &= \begin{pmatrix} \alpha_{p_i \in C_1} & \nabla_L^{p_i \in C_1} & \nabla_R^{p_i \in C_1} & \gamma_{p_i \in C_1} & \theta^+_{p_i \in C_1} & \theta^*_{p_i \in C_1} \\ \alpha_{p_j \in C_2} & \nabla_L^{p_j \in C_2} & \nabla_R^{p_j \in C_2} & \gamma_{p_j \in C_2} & \theta^+_{p_j \in C_2} & \theta^*_{p_j \in C_2} \\ \vdots & & & & & \\ \alpha_{p_m \in C_n} & \nabla_L^{p_m \in C_n} & \nabla_R^{p_m \in C_n} & \gamma_{p_m \in C_n} & \theta^+_{p_m \in C_n} & \theta^*_{p_m \in C_n} \end{pmatrix} \end{aligned} \tag{5}$$

VDTL is an $N \times 6$ matrix, with row i being the descriptor of each sample point p_i of shape S.

From this definition, we can easily prove that the VDTL shape descriptor has intrinsic invariance to translation, rotation, and scaling of the shape S. When the shape S is rotated, the curvature and tangent vector of the points on S do not change since they are the intrinsic invariants of the point. Furthermore, to ensure that our shape descriptor is scale-invariant, we normalize ∇_L^i, ∇_R^i, and h_{p_i} by dividing by l.

The shape matching based on VDTLs is very simple: $VDTL\ (S_1)$ and $VDTL\ (S_2)$ denote the VDTL descriptors that are extracted from shapes S_1 and S_2, respectively. Therefore, the shape matching based on VDTLs can be performed by:

$$Dist(S_1, S_2) = \frac{1}{6} \sum_{j=1}^{6} \left(\frac{1}{N} \left(\sum_{i=1}^{N} VDTL_i^j(S_1) - \sum_{i=1}^{N} VDTL_i^j(S_2) \right) \right) \tag{6}$$

where i is the row index and j is the column index. The smaller the distance, the more similar the two shapes are.

Since the 2D shapes are refined by using the curve reconstruction method described in [2] that is highly robust to noise, the VDTL is still effective although it depends on the computing of curvatures that is sensitive to noise. The searching of the matching pairs based on VDTL is very simple since there is no need to set any thresholds, that is to say all of the parameters are computed adaptively. Furthermore, the major advantage of the proposed method is its computing efficiency in searching the matching 2D shapes because it can distinguish two shapes first according to the number of the curve segments and then computes $Dist(S_1, S_2)$ for two shapes having the same number of curve segments.

B. CONSTRAINT 2: SKELETAL CURVE MATCHING

The skeletal curve is a 3D curve and its matching can be achieved by any 3D curve matching methods. We use the method proposed in [5] in our experiments.

C. CONSTRAINT 3: SCALING FACTOR MATCHING

For the scaling factor matching of each 2D shape, we compute the scaling gradients among its neighboring 2D shapes using Equation (7):

$$\begin{cases} \nabla_L^{l_{S_i}} = \sum\limits_{j=1}^{k} \dfrac{l_{S_i} - l_{S_{i-j}}}{j \cdot t} \cdot \dfrac{1}{\sqrt{2\pi}} \exp \dfrac{j}{N} \\ \nabla_R^{l_{S_i}} = \sum\limits_{j=1}^{k} \dfrac{l_{S_i} - l_{S_{i-j}}}{j \cdot t} \cdot \dfrac{1}{\sqrt{2\pi}} \exp \dfrac{j}{N} \end{cases} \tag{7}$$

where k is the number of neighbors, j is the index, t is the distance between two adjacent equidistant sample points on the skeletal curve, N is the number of the sample points on the skeletal curve. The $Dist(l_{C_1}, l_{C_2})$ measures the difference degree of the scaling factors l_{C_i} of the two components C_i. The smaller the distance, the more similar the two l_{C_i} are.

$$\begin{aligned} Dist(l_{C_1}, l_{C_2}) = &\left| \sum \nabla_L^{l_{S_i \in C_1}} - \sum \nabla_L^{l_{S_i \in C_2}} \right| \\ &+ \left| \sum \nabla_R^{l_{S_i \in C_1}} - \sum \nabla_R^{l_{S_i \in C_2}} \right| \end{aligned} \tag{8}$$

Two given components match if the above three constraints are met.

VI. RESULTS AND DISCUSSION

We implement the proposed method on a variety of point-sampled models. Our experiments consist of five parts. First, the performance of the 2to3SSC generation is tested thoroughly with regards to the sensitivity to the key parameter k (kNN), the robustness to the non-uniform distribution of points, the robustness to noise, and the robustness to the presence of holes. Furthermore, the timing information is assessed for different configurations. Second, the effectiveness of the VDTL descriptor is evaluated using the MPEG-7 dataset. Since 2D shape matching is a part of component matching, the details of the similarity errors are discussed in the component matching results. Third, we test the descriptiveness of the 2to3SSC using the well-known 3D McGill Shape Benchmark (MSB) dataset and discuss the similarity errors, robustness to noise, and the matching efficiency. Fourth, we compare our method and state-of-the-art feature description and matching methods to demonstrate the advantage of our method over traditional feature descriptors using models from the SHREC10 dataset. Finally, the proposed method is applied to a real-world dataset (i.e., the Terracotta fragment models that are raw scans without any noises removal) to highlight the utility of the proposed method.

Our experiments were implemented on an Intel Core i7-5500U CPU with 2.40 GHz and 8 GB RAM. The code used to implement the proposed method was not optimized. All input point clouds to our algorithm were unorganized and non-oriented and had varying degrees of noise and non-uniform acquisition density. We empirically set $k = 30$ as the default setting for all results, except for the example in Fig. 7, where we demonstrate the influence of different k values. The limitations of the proposed method are summarized in the last subsection.

A. THE PERFORMANCE OF 2TO3SSC GENERATION

In this part, we focus on the performance of the 2to3SSC generation concerning the robustness to various factors and the efficiency; therefore, we use the six components of lamp model #1 and #2 (in Fig. 2) as the test models for the sake of readability.

1) THE PARAMETERS OF 2TO3SSC

The 2to3SSC descriptor has two key parameters: (i) the number of segments, which is determined by the scale of the neighborhood (a smaller k generates more segments) in the component segmentation step and affects the extraction of skeletal curves; and (ii) the number of 2D shapes in the 2D shape extraction step, which is the same as the number of sample points on the skeletal curve. These two parameters present the "completeness" and "details" of the 2to3SSC descriptors. In order to simplify the parameter settings, we set the number of 2D shapes that is user-defined as the number of segments. Therefore, only the neighborhood scale impact the results. Hence, the sensitivity to the neighborhood scale is assessed at different neighborhood scales, i.e., $k = 10,\ 20$, and k increasing from 20 to 60 in increments of 20. Fig. 7 shows the results of using different scales of neighborhoods for the components of lamp model #1 and #2; all exhibit varying degrees of non-uniform acquisition density and the detailed information is listed in Table 2. The results of using different neighborhood scales indicate that the 2to3SSC generation proved insensitive to the neighborhood scale, except if the neighborhood is very large (larger than 60) or very small (smaller than 20). This is because a larger or smaller neighborhood results in incomplete segments that fail to outline the local shape of the component.

In terms of the descriptiveness of the 2to3SSC, more 2D shapes provide more details about the components, as shown in Fig. 7, therefore, the similarity errors of 2to3SSC are lower for a larger number of 2D shapes; the details are listed in Table 5. However, according to the matching information listed in Table 5, we observed that the similarity errors of 2to3SSC decreased slightly as the number of 2D shapes increased, whereas the computational cost for matching increased significantly. Thus, we set $k = 30$ (the corresponding number of segments are listed in Table 3) as a default to avoid unnecessary details and ensure computational efficiency. The results using $k = 30$ are shown in Fig. 8.

2) THE ROBUSTNESS TO NON-UNIFORM DISTRIBUTION OF POINTS

We define the average distance d_p of a data point p with the k nearest neighboring points N_p as the local sampling density [17]:

$$d_p = \frac{1}{|N_p|} \sum_{p_j \in N_p} |p - p_j| \tag{9}$$

Table 3 lists the minimum and average d_p of the tested models and Fig. 7 and Fig. 8 show the handling of point clouds that exhibit strong local imbalances in the sampling pattern. In order to test the robustness to non-uniform distribution of points, we also tested our approach on simplified models (at simplification rate of approximately 15%), which have the same surfaces but different sampling densities. The detailed information is given in Table 3. Here again, the proposed method delivers favorable results as shown in Fig. 9 and the number of segments of some models decreases when the models are simplified, such as component A of Lamp #1 and component A of Lamp #2.

3) THE ROBUSTNESS TO NOISE

In order to evaluate the robustness to noise, we add Gaussian noise with variances of $0.03A\_d_p$, $0.06A\_d_p$ and $0.09A\_d_p$ to the models, A_d_p is the average d_p that are listed in Table 3. The results of skeletal curve generation and 2D shape extraction at different levels of Gaussian noises are presented

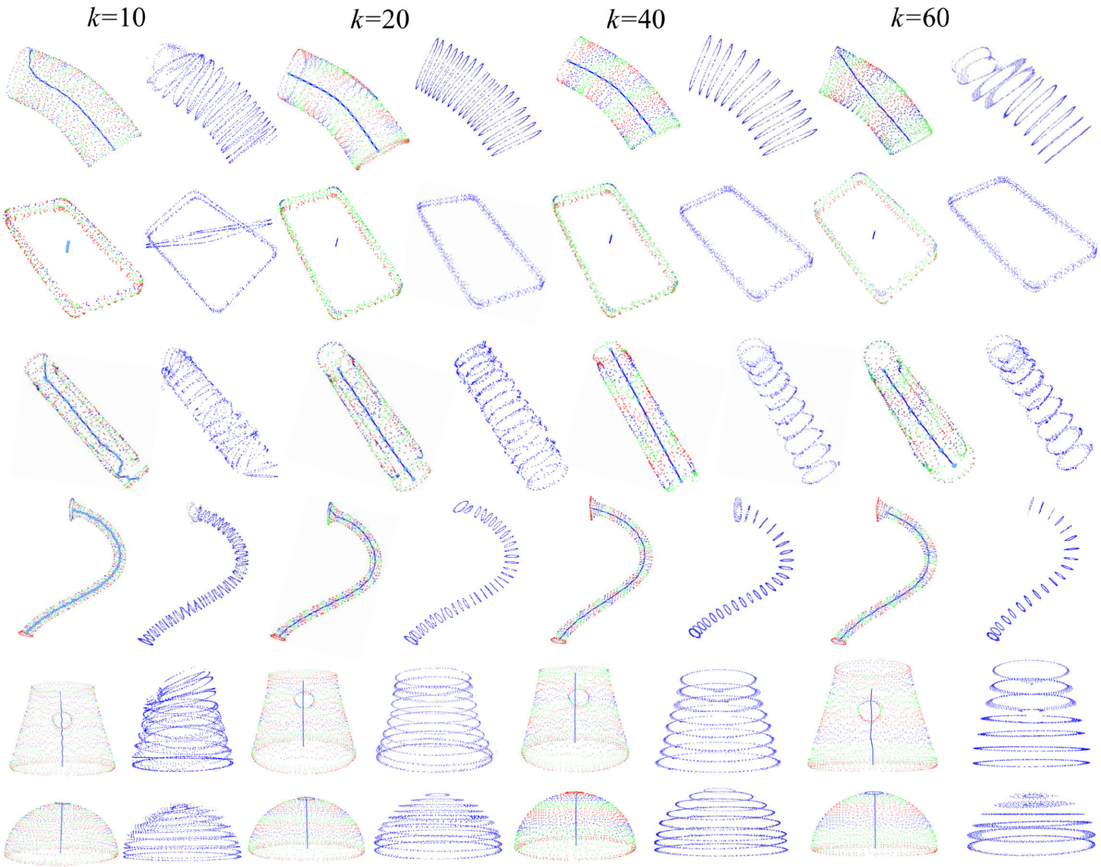


FIGURE 7. Skeletal curves and 2D shapes that are extracted by the proposed method using different neighborhood scales.

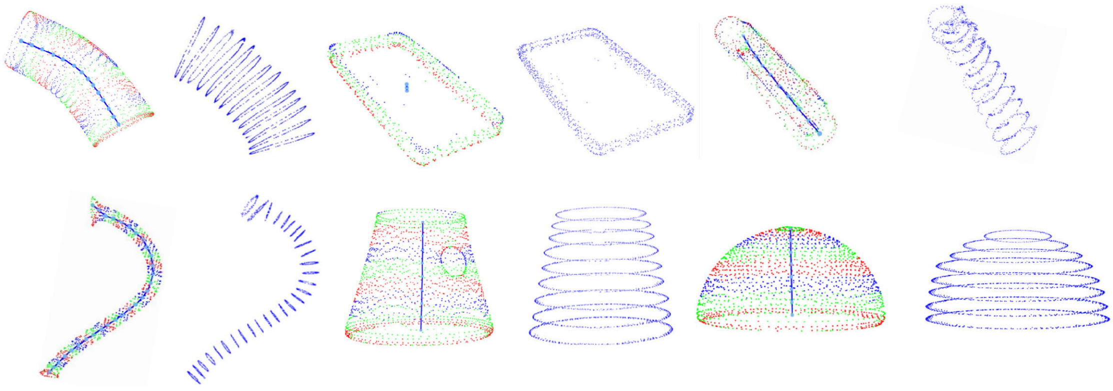


FIGURE 8. The results of skeletal curve and 2D shapes extraction from the original models using $k = 30$.

in Fig. 10, and the detailed information is listed in Table 4. It can be seen from Fig. 10 that the results of the skeletal curve generation and 2D shape extraction are not significantly affected at low noise levels since the segments are well delineated and the L_1 median is not sensitive to the presence of noise. At only high noise levels, we observe that some parts of the skeletal curves are incorrect, resulting in unexpected 2D shapes. This is because the noisy points influence the segments generation, thus the segments fail to outline the shape.

4) THE ROBUSTNESS TO HOLES

When segmenting point clouds, the appearance of holes is common. The location and size of the holes affect the results of the skeletal curve generation and the extraction of the 2D shapes. If the holes are on the top or bottom surfaces of the components, such as component B in lamp model #1 and components A and C in lamp model #2, they do not impact the results. However, if the holes are on the side surfaces, the size of the holes is a key influencing factor. For example, the results of the skeletal curve generation of the

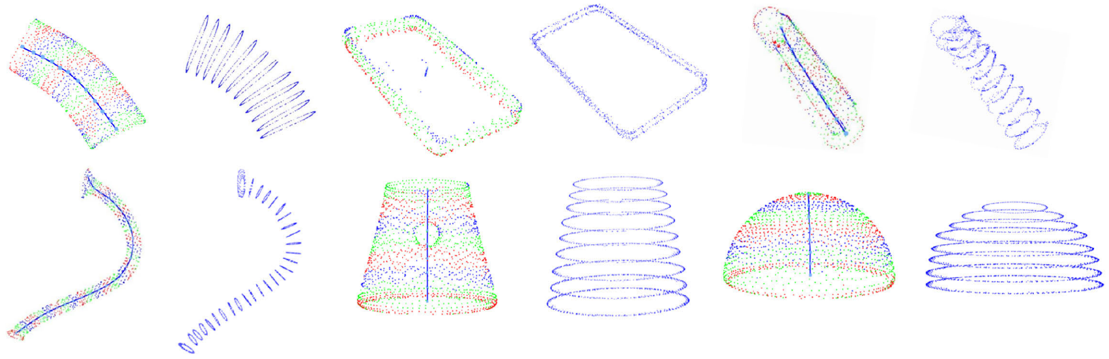


FIGURE 9. The results of skeletal curve and 2D shapes extraction from the simplified models using $k = 30$.

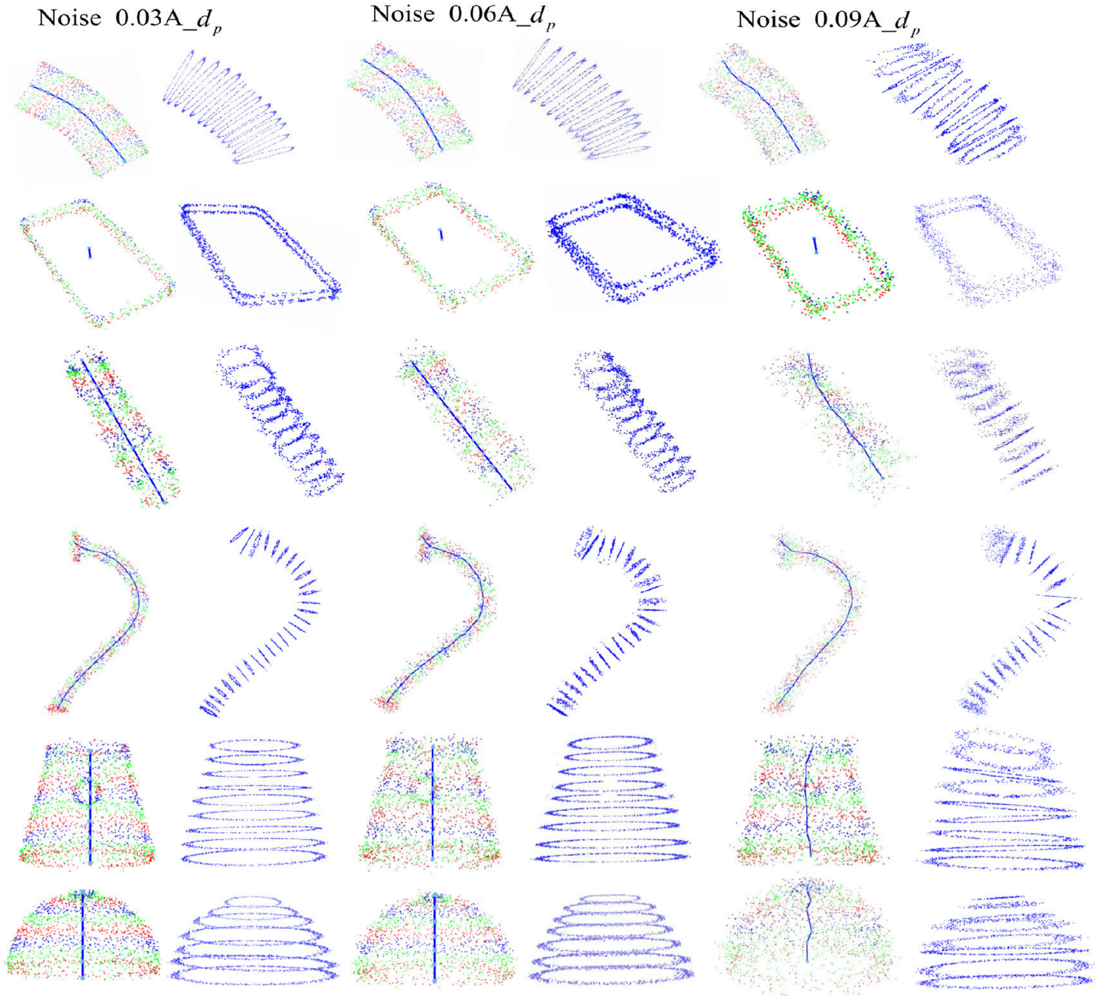


FIGURE 10. The results of skeletal curve and 2D shapes extraction at varying noise levels.

component A of lamp model #1 and B of lamp model #2 are not influenced by the holes because the size of the holes is small. However, the holes on the side surfaces inevitably result in incomplete 2D shapes, which leads to an increase in matching errors but the wrong matching results can be avoided by increasing the similarity thresholds (which is discussed in the component matching results). If the size of the holes on the side surfaces increases, the holes need to be filled first, for example, the component C of lamp model #1. Otherwise, the generated skeletal curves may be incorrect; this is considered a drawback of the 2to3SSC descriptor.

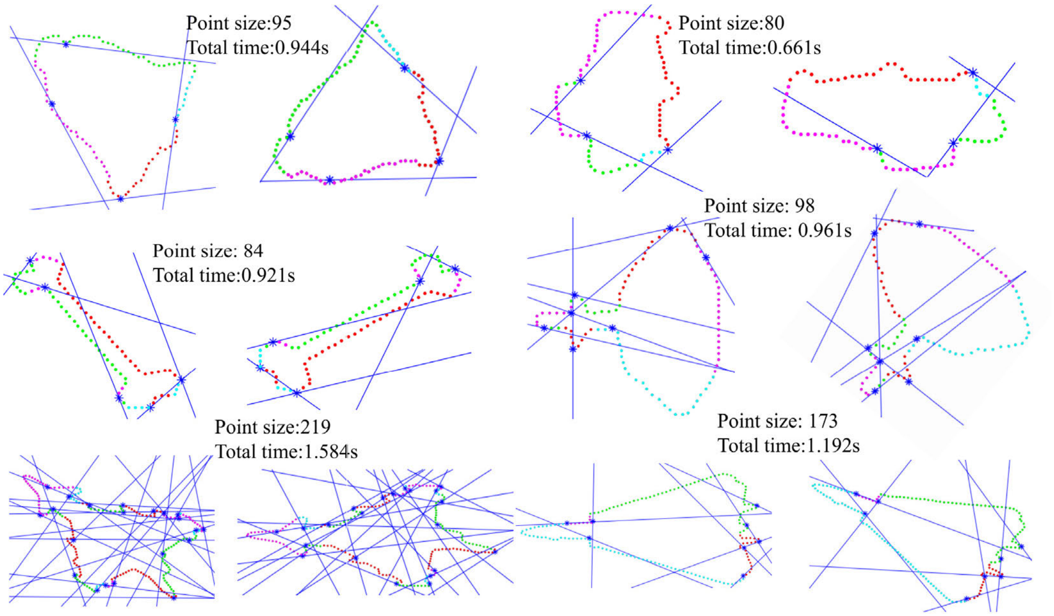


FIGURE 11. Examples of 2D shape matching in the MPEG-7 dataset with the proposed VDTL descriptor. The points marked with an asterisk are the benchmark points, the blue lines are the tangent lines of the benchmark points, and the contours of the shapes are divided into different curve segments with different colors.

TABLE 2. The number of segments and the execution time using different *k*NNs.

Component	*K*	Segments	Time(s)			
			Step 1.1	Step 1.2	Step 1.3	Total
A of Lp1	10	27	4.992	0.374	0.230	5.596
	20	18	4.802	0.332	0.155	5.289
	40	12	5.598	0.324	0.121	6.043
	60	10	5.659	0.302	0.101	6.062
B of Lp1	10	4	2.009	0.215	0.050	2.274
	20	3	2.276	0.212	0.031	2.519
	40	2	2.251	0.205	0.013	2.469
	60	2	2.718	0.204	0.012	2.934
C of Lp1	10	34	3.387	0.452	0.365	4.204
	20	13	3.338	0.335	0.182	3.855
	40	10	3.356	0.315	0.157	3.828
	60	9	3.631	0.303	0.113	4.047
A of Lp2	10	54	6.660	0.539	0.532	7.731
	20	36	5.453	0.422	0.363	6.238
	40	25	6.758	0.420	0.264	7.442
	60	20	6.401	0.409	0.256	7.066
B of Lp2	10	17	10.733	0.543	0.201	11.477
	20	10	10.646	0.540	0.185	11.371
	40	8	12.141	0.533	0.109	12.783
	60	7	13.431	0.557	0.091	14.079
C of Lp2	10	15	8.297	0.525	0.223	9.045
	20	10	7.917	0.477	0.153	8.547
	40	8	8.622	0.442	0.102	9.166
	60	6	9.768	0.421	0.086	10.275

5) EFFICIENCY

In this section, the efficiency of the 2to3SSC method in terms of the execution time for the generation is assessed for different configurations. Table 2 displays the execution time of each step and total execution time using different k values for the tested models. It is evident that a significant portion of the running time is devoted to component segmentation. However, there is no relationship between the execution time of the component segmentation and the k values since the time spent on this step can be divided into two parts: the number of region-growing operations and searching of the k NN. A larger k results in a lower number of region-growing operations but the searching of the k NN requires more time and vice versa. However, the execution time for skeletal curve generation and 2D shape extraction decreases as k increases because the number of segments and 2D shapes decreases as k increases.

From another perspective, it is apparent that the execution time is directly related to the point cloud size of the models. As shown in Table 3, a decrease in the point cloud size results in a decrease in the total execution time. However, an increase in the noise level results in an increase in computational costs (as shown in Table 4), which is attributed to the fact that a noisy point cloud increases the data volume of the model and, therefore, the kNN searching requires more time.

B. THE EFFECTIVENESS OF VDTLS

The effectiveness of the VDTLs is tested using the MPEG-7 shape dataset [64], which has been extensively used to evaluate 2D shape descriptor performance. Since the 2D shapes matching in 2to3SSC focuses on the complete matching for 2D shapes with no deformation, we only present such matching results.

Six groups of 2D shape matching results are shown in Fig.11, the shapes in the datasets have been randomly rotated, translated and scaled down. For each shape on the left, the shape (on the right) with the smallest $Dist(S_1, S_2)$ is considered the best match. It can be seen from Fig. 11 that the VDTL is effective for 2D shapes matching and robust to the rigid transformations. The point size and the total time for matching are shown in the figure as well. It can be

TABLE 3. The sampling and execution time of the original models and simplified models using $k = 30$.

Components	Original sampling					Models simplified to 85%				
	Point Size	Segments	Min d_p	Average d_p	Total time (s)	Point Size	Segments	Min d_p	Average d_p	Total time (s)
A of Lp1	1463	14	0.055098	0.271714	5.530	1243	12	0.056545	0.285275	4.461
B of Lp1	1467	2	0.018703	0.314168	2.006	1246	2	0.018755	0.340441	1.978
C of Lp1	990	11	0.016363	0.125594	3.866	809	11	0.016574	0.135990	3.526
A of Lp2	1266	29	0.012376	0.281555	5.948	1052	28	0.016917	0.281838	5.615
B of Lp2	2505	9	0.051378	0.315789	12.073	2121	9	0.052917	0.335791	10.688
C of Lp2	1970	8	0.025246	0.154778	8.741	1671	8	0.025246	0.166040	8.658

TABLE 4. The total execution time for different noise levels using $k = 30$. The times are in seconds.

Components	Total Time		
	Noise $0.03A\_d_p$	Noise $0.06A\_d_p$	Noise $0.09A\_d_p$
A of Lp1	5.562	5.665	5.745
B of Lp1	2.169	2.200	2.224
C of Lp1	3.878	3.980	3.995
A of Lp2	6.667	6.600	8.515
B of Lp2	12.429	12.435	12.618
C of Lp2	8.875	8.892	9.177

observed that, the time spent for matching are increased with the increase of the point size. The detailed similarity errors are discussed in the next subsection with respect to the present of noises and holes.

C. COMPONENT MATCHING RESULTS

In this section, we first select three typical components to test the performance of the component matching for the sake of readability. And then we present the results on the entire models.

1) SIMILARITY ERRORS

The lamp model dataset contains different styles of lamps; some components are highly similar whereas some are entirely different. Therefore, this dataset is very suitable for testing the descriptiveness of 2to3SSC. The matching results for the three components of lamp model #2 are shown in Fig. 12. The first column of Fig. 12 shows the given components and the other columns are the matching components in the lamp model dataset (with the matched part highlighted) in order of increasing similarity errors. For a quantitative evaluation, we use the total $Dist(S_i \in Component\#1, S_j \in Component\#2)$ to measure the similarities of 2D shapes, the ratio for non-matching characters to the total characters in OLAPs [5] to measure the similarities of the skeletal curves, and $Dist(l_{C_1}, l_{C_2})$ to measure the similarities of the scaling factors of the two series of 2D shapes that are stacked to construct the components.

We observe that the proposed method effectively determines the matching components and all of the three mentioned constraints contribute to the component matching. Specifically, the skeletal curves of the components in the first row of Fig. 12 are different so that the second values (similarity errors of skeletal curves) are all higher than the given thresholds (listed in Table 6); the scaling factors of the components from the fourth column to the sixth column in the 2nd row and 3rd row are different, thus the third values are all larger than the thresholds. However, the values in magenta are relatively larger but not larger than the given thresholds since small holes on side surfaces lead to a slightly increase in similarity errors of 2D shapes and scaling factors.

2) ROBUSTNESS TO NOISE

In this section, we tested the robustness to Gaussian noise. The skeletal curve and 2D shapes extracted from noisy models with $0.03A\_d_p$, $0.06A\_d_p$ (shown in Fig. 10) are implemented to search matching components. The search results are the same as the results in Fig. 12 and the similarity errors are listed in Table 5. We observe that the matching results are not affected at low noise levels because the skeletal curve and 2D shape extraction are robust to noise and both the skeletal curve and 2D shapes are reconstructed using the L_1 curve reconstruction method, which is also robust to noise. Although larger noise levels result in larger similarity errors, the different components can still be distinguished.

3) EFFICIENCY

In this section, we analyzed the efficiency of 2to3SSC in terms of the time required to match the components. Because the skeletal curve matching, the 2D shape matching, and the scaling factor matching can be achieved by other methods, here we only analyze the total time required for matching with respect to specific parameters of the 2to3SSCs. The total matching time for 2to3SSCs and the different k values (which determine the number of segments, 2D shapes, and sampling points on the skeletal curves) are listed in Table 5. It is evident that the number of 2D shapes affects the matching time, i.e. a larger number of 2D shapes (smaller k value) increases the matching accuracy but also increases the computational cost. This is a trade-off between accuracy and efficiency.

4) RESULTS

In this section, we highlight the interesting results to demonstrate the applicability of the proposed method. The adaptive computations of the parameters for similarity measurements

TABLE 5. The similarity errors and execution time of the best matches using different *k*NN, and the similarity errors for different levels of noise using $k = 30$. The number of 2D shapes and the number of sampling points on skeletal curves are listed in Table 2 and Table 3. The execution times are in seconds.

Components	*K*=20		*K*=30		*K*=40		The similarity errors	
	The similarity errors	Total tine	The similarity errors	Total tine	The similarity errors	Total tine	Noise $0.03A\_d_p$	Noise $0.06A\_d_p$
A of Lp1	0.125	6.171	0.137	4.771	0.149	4.571	0.155	0.172
B of Lp1	0.033	2.035	0.042	1.697	0.053	1.197	0.068	0.082
C of Lp1	0.029	2.142	0.038	1.725	0.046	1.239	0.062	0.077

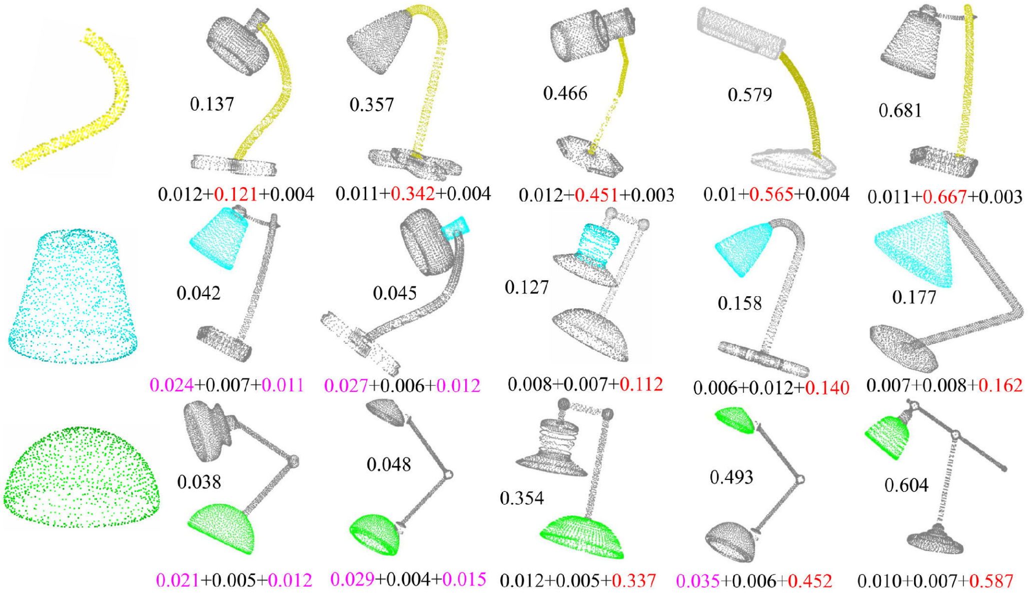


FIGURE 12. Results of component matching for lamp model #2 with 2to3SSC. The second to sixth columns depict the matching components in the lamp model dataset in order of increasing similarity errors. The matching pairs of the components are in the same color, but the components in grey are not matched. The kNN is set as $k = 30$. The total similarity errors are beside each matching component. The three values listed under each matching object are the similarity errors of 2D shapes, skeletal curves and the scaling factors, respectively. The error values in magenta are relatively larger but they are still not larger than the given thresholds, whereas the error values in red are larger than the given thresholds.

TABLE 6. parameters of matching methods.

Parameters	Values
ω_1	$0.2 \cdot \min(\sum\sum VDTL(S_i), \sum\sum VDTL_i^j(S_j))$
ω_2	0.1
ω_3	$0.2 \cdot \min(\sum \nabla_L^{l_{C_1}}, \sum \nabla_R^{l_{C_2}})$
ω_4	0.8

are given in Table 6, where ω_1 measures the $Dist(S_i \in$ Component #1, $S_j \in$ Component #2) for 2D shape matching, ω_2 measures the ratio for non-matching characters to the total characters in the OLAPs [5] for skeletal curve matching, ω_3 measures the $Dist(l_{C_1}, l_{C_2})$ for scaling factor matching, and ω_4 measures the percentage ratio for matching the 2D shapes to the total number of 2D shapes because the presence of holes may increase the similarity errors between the 2D shapes to greater than ω_1 and ω_3 respectively.

For these classes in the dataset, we first select one model as the query and the matching components are searched from the remaining models in the class. Then the query model is put back, and the procedure is repeated with another selected model from the class, etc., until all of the models in the same class have been selected as queries.

The matching results for the lamp class are shown in Fig. 13. We observe that there are 32 components (in blue), which are constructed by stacking circles of the same size; 7 components (in orange) are constructed by stacking rectangles of the same size along straight lines, 3 components are constructed (in cyan) by stacking circles but they are different from the other 3 components (in green) that are also constructed by stacking circles since the scale of the circles is different. All of the lamps have at least one component that matches the others, except for the lamp model in the red rectangle. It is also observed in Fig. 13 that lots of lamp posts are constructed by stacking circles with no scaling transformation and the only difference is the shape of the skeletal curves. Thus, if we only consider the matching of the 2D shapes (Constraint 1) and the scaling factors (Constraint 3), all of the cylindrical components that are created by stacking the circles

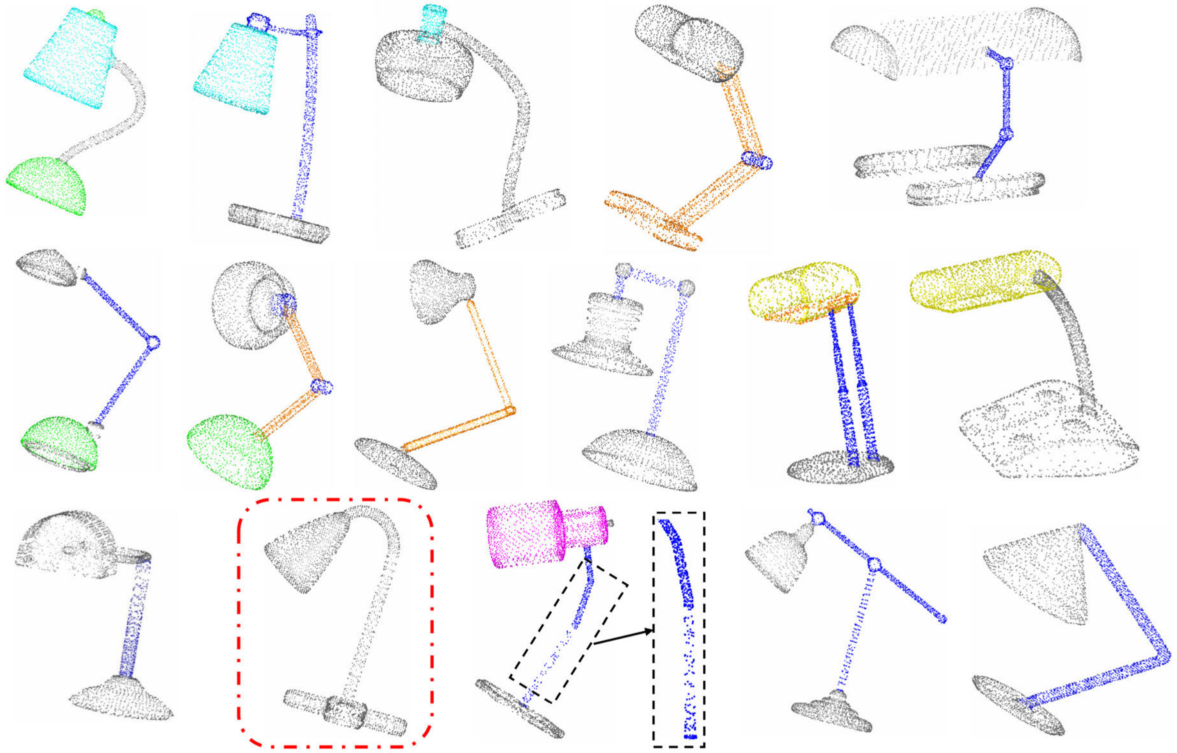


FIGURE 13. Component matching results of the lamp class; the matching pairs of components that are searched by the similarity measure of 2to3SSC are shown in the same color, but the components in gray are not matched.

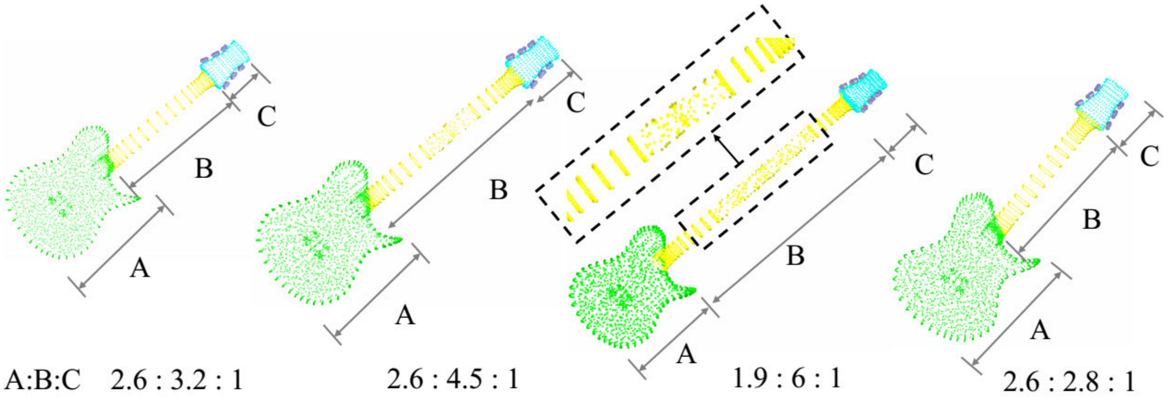


FIGURE 14. Four guitars that have the same components are searched by the similarity measure of 2to3SSCs.

of the same size are matched. However, they are dissimilar from a human visual perspective. Therefore, we concluded that all of the constraints mentioned in section 5 are important for shape matching of the components or the point clouds.

Fig. 14 and Fig. 15 show the interesting matching results of the guitar class and the vase class, respectively. It can be seen in Fig. 14 that the four guitars have the same four components. However, the differences are the sizes of the components, which results in different styles and beauty. Fig. 15 shows that the first four vases are the same but the positions of the components A in yellow imply different statuses of the vases, and the component A is unfixed (note that this is different from a non-rigid transformation), and the last vase has two components A that are shown in yellow.

5) SUMMARY

All of the tested point clouds are sampled non-uniformly and some special cases are marked with black dashed-line rectangles in Fig. 13 - Fig. 14. The results demonstrate the robustness of the 2to3SSC method to the non-uniform sampling of points. However, Fig. 16 shows the matching results of two vase models. The components in green are highly similar and the components in yellow are only matched by 2D shape matching and scale factor matching. However, the other four components in orange, cyan, purple, and pink cannot be matched because no 2D shape is detected due to the poor sampling of the points. This is a condition that our proposed method does not handle well. Fig. 13 - Fig. 16 show various models with different skeletal curves, 2D shapes, and

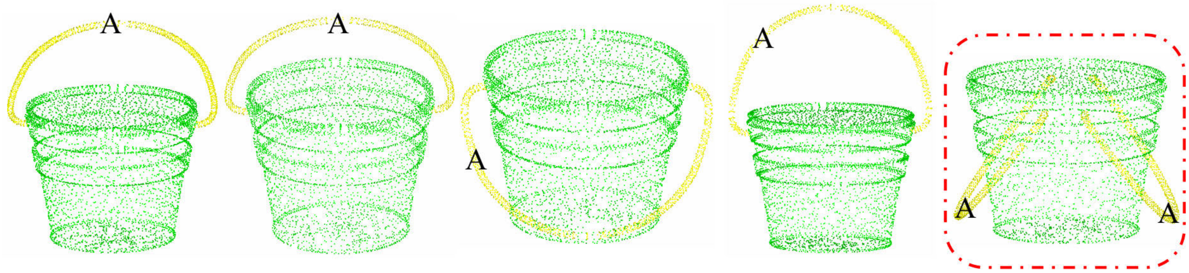


FIGURE 15. Five vases that have the same components, the matching pairs of components that are searched by the similarity measure of 2to3SSCs are in the same color, whereas the last one has two components A that are shown in yellow.

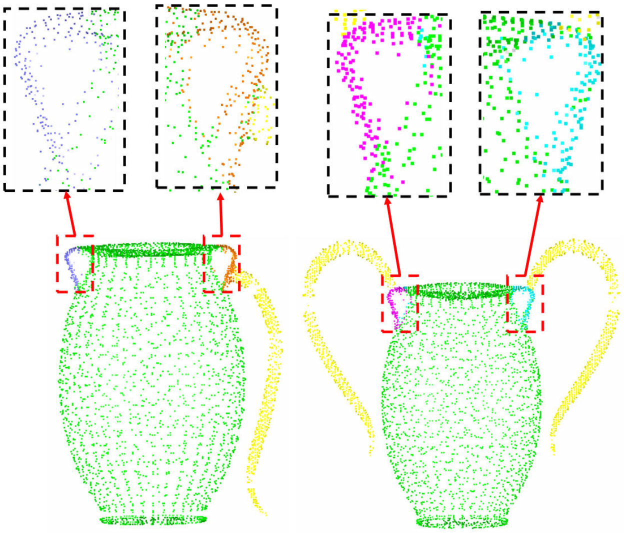


FIGURE 16. Two vase models that are partial matched by the similarity measure of 2to3SSCs.

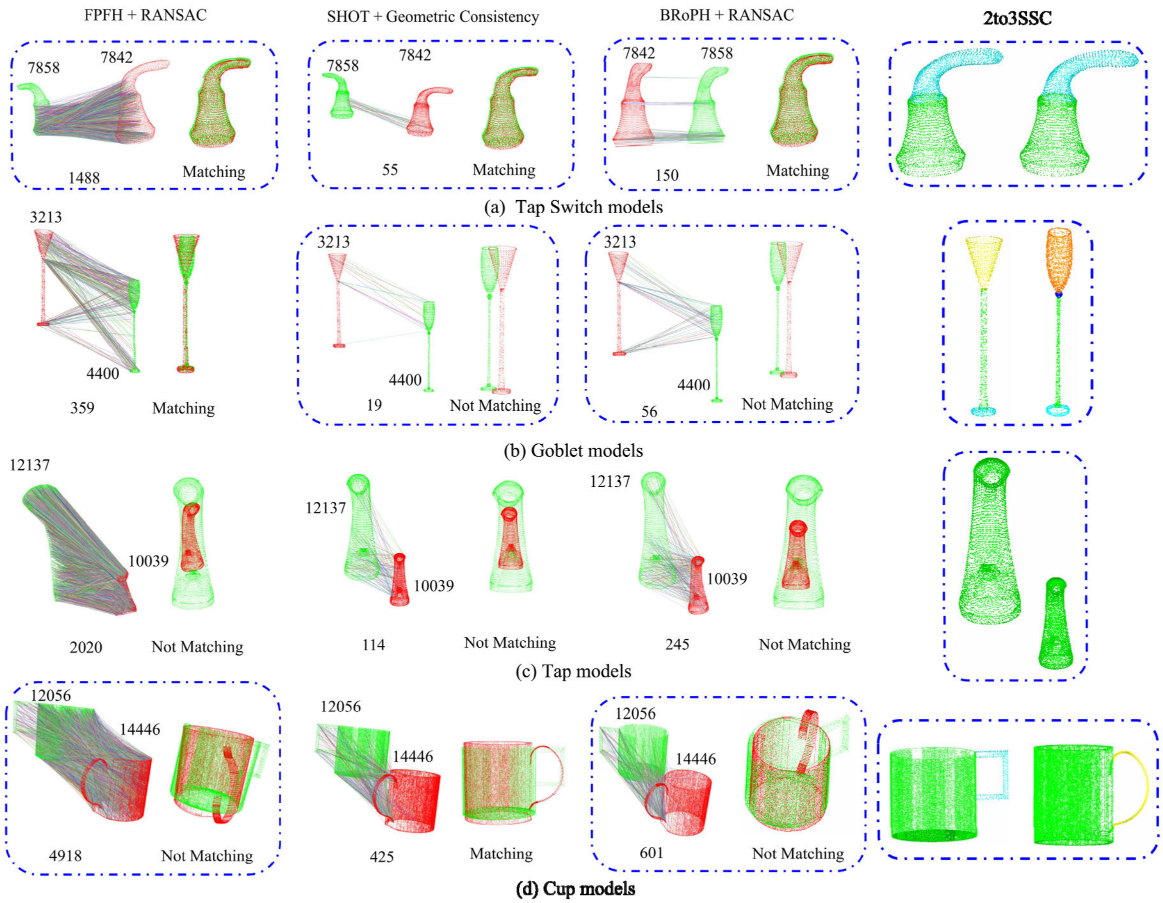
scaling factors; the results solidly confirm the effectiveness of the 2to3SSC. Although a larger noise level results in larger similarity errors, the different components can still be distinguished. Furthermore, the matching results in Fig. 14 demonstrate that the component matching can be utilized for a style analysis for the same type of objects and Fig.15 demonstrates that the component matching can detect different states or actions of the same object.

D. COMPARISONS

Fig. 1 shows that the micro-features and the skeleton-based macro-features provide insufficient information on the shape of the two goblets. Specifically, it is evident that the feature lines (Fig. 1(b)) are all matched (at least for some components); however, the similarity of the components in the point clouds cannot be determined. As shown in Fig. 1 (d), the skeletons are matched as well but the point clouds are different. The reason for the unsatisfactory results of the micro-features and macro-features is that they only encode local or partial features of the point clouds; this is insufficient to obtain a similarity measure of the components, not to mention the matching of the entire point clouds.

If only meso-features are used, the similarity of the point clouds cannot be determined as shown in Fig. 1 and Fig. 17. However, in some cases, the meso-features can be utilized for similarity measures and registration of point clouds but some post-procedures must be applied, such as random sample consensus (RANSAC) or geometric consistency (GC) tests. In order to compare the effectiveness of using the meso-features and 2to3SSC for obtaining a similarity measure of the point clouds, we combined three typical meso-features (SHOT [29], FPFH [28] and BRoPH [6] with either RANSAC or GC in PCL (Point Cloud Library) to measure the similarity of the given point clouds. The FPFH is computed for all of the points, whereas the SHOT and BRoPH are computed for key points that are detected using uniform sampling in PCL. The point correspondences are searched using KD-Tree. The results are shown in Fig. 17, where all of the parameters are fine-tuned. Although the combination of the meso-features and RANSAC or GC can measure the similarity of the point clouds (as shown in Fig. 17 (a)), there are certain disadvantages. For example, as shown in the Goblet models in Fig. 17 (b), FPFH with RANSAC fails since it uses the distances between corresponding points to measure the similarity of the aligned two models, which is ineffective, whereas SHOT with GC and BRoPH with RANSAC provided satisfactory results. However, for the two Tap models of different sizes (Fig. 17 (c)), all applied methods failed to provide the similarity of the given models because RANSAC is sensitive to scales, and GC is ineffective on the smooth surfaces, on which the point correspondences obtained by the meso-features are massive due to the similarity of the feature points. We also observed that for models with common components, as shown in Fig. 17 (d), the three meso-features with RANSAC or GC cannot detect the matching components directly; therefore we segmented the cups into components and then used the three meso-features with RANSAC or GC to measure the similarity of the components; this also failed since the cup sizes are different and the cups have smooth surfaces. Hence, we concluded that the results of the similarity measure using meso-features highly depend on the performance of the post-procedures, such as RANSAC and GC tests.

The proposed method, however, outperforms the other methods because it encodes both the local features (2D shapes) and global features (skeletal curve and scaling



(a) Tap Switch models

(b) Goblet models

(c) Tap models

(d) Cup models

FIGURE 17. The comparison of different meso-features and 2to3SSC. For each model, the scale of the points are beside each model; for each group of models, the number below the models is the number of the point correspondences; the results in the blue rectangles are correct whereas the other results are incorrect.

factors) rather than only encoding local features or partial features. Consequently, the proposed method handles the matching of the components and provides favorable results, as shown in the last column of Fig. 17 in which the matched components have the same color. Furthermore, it should be noted that the cups have both inner and outer surfaces and the results show that the inner surface does not influence the skeletal curve generation and the 2D shape extraction because the L_1-median based line reconstruction has excellent performance. However, the method fails to handle the inner surfaces if the distances between the local inner and outer surface are non-uniform.

E. APPLICATIONS

Our method focuses on the component-oriented partial matching of point clouds, which is accomplished by searching for similar parts in the original models based on 2to3SSC. Compared to traditional feature matching of point clouds (feature points, feature lines, surface patches, and skeletons), the proposed descriptor encodes more shape information; therefore, it has broad application value. In this section, we present an application of the proposed method, i.e., a novel point cloud classification method based on typical element recognition.

The example is the cultural element detection in the point clouds of Terracotta fragments. The point clouds of the fragments were scanned by students using a Creaform VIU handy scanner. The scan resolution was 3.91 mm, which favors speed but results in relatively low precision. The point clouds exhibit strong local imbalances in the sampling pattern and contain realistic noise that was the results of the scanning process.

A typical cultural element in Terracotta is the approximate hemisphere called ‘‘Jia Ding’’ in Chinese, which is marked in green in Fig. 18, and another typical cultural element is called ‘‘Jia Dai’’, which is marked in cyan in Fig.18. The Terracotta fragments have the following characteristics: (i) the fragments with Jia Ding and Jia Dai belong to a Terracotta warrior but not a horse or others; (ii) the fragments with Jia Ding and Jia Dai belong to an armor located on the upper body but not on the four limbs or the robe. Thus the detection of the Jia Ding and Jia Dai elements can help with the classification of the fragments and further facilitate the reassembly of the fragments because (i) the classification results reduce the

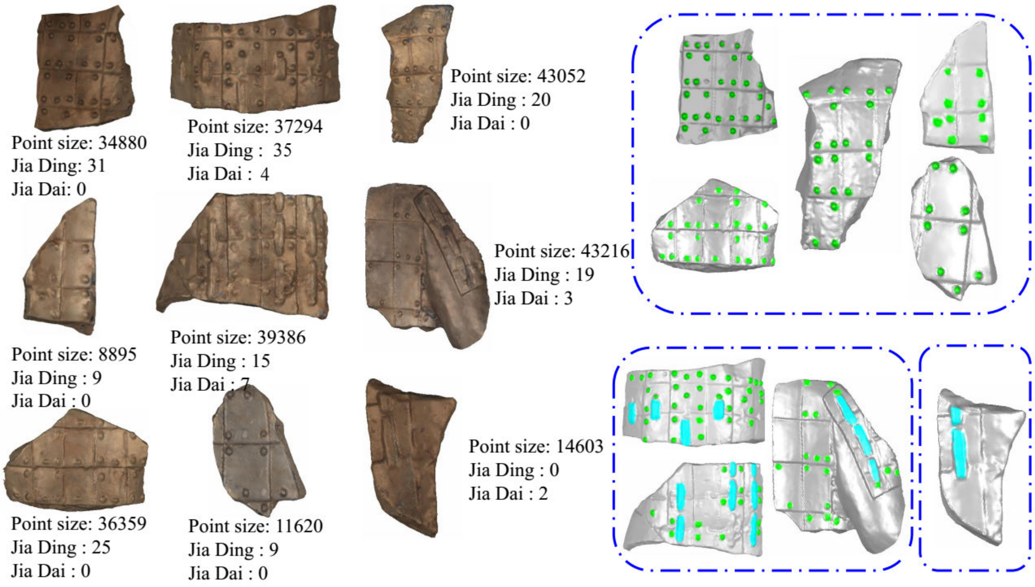


FIGURE 18. **The Terracotta fragment models and the classification results that are achieved by the proposed component matching.**

search range for the adjacent fragments and (ii) the layout of the Jia Ding and Jia Dai on the armor is mostly regular.

The fragments we tested and the detection results of the Jia Ding and Jia Dai elements are shown in Fig.18; the point size and the number of Jia Ding and Jia Dai are listed beside each fragment. Subsequently, the fragments are classified into three classes: the first class contains the fragments with only Jia Ding elements, the second class contains the fragments with only Jia Dai elements, and the third class contains fragments with both Jia Ding and Jia Dai elements. Here again, the proposed method delivers favorable results for the real scanned data. The total time required to classify the 9 fragments is about 35 minutes since the point size of the fragment ranges from ten thousand to forty thousand. The recognition results of the Jia Ding and Jia Dai elements can also be used for the auto-generation of labels, which is a necessary part of supervised learning because manual labeling is time-consuming. These examples demonstrate the effectiveness of 2to3SSC and highlight the utility of the component-oriented partial matching of point clouds.

F. LIMITATIONS

As the experiments have shown, failures may occur if our method cannot detect the 2D shape correctly in rare cases (as shown in Fig. 16). Similar, if the components are not constructed by stacking 2D shapes, which may happen in rare cases, the proposed method fails to describe the entire shape. Also similar to other techniques, the feature description of point clouds is an ill-posed problem, especially when there are missing data. Thus, if the distribution of the point data is highly non-uniform or large holes exist, the proposed method may produce erroneous outputs. One solution to partially alleviate the problem of the holes is to fill the holes first and then apply the proposed method. For models that have both inner and outer surfaces, the method may fail if the distances between the local inner surface and outer surface are non-uniform.

VII. CONCLUSION

While partial matching of 2D shapes has received significant attention in the field of computer graphics, the more challenging partial matching of point clouds, in which the goal is the description and matching of the components in the point clouds, has not been previously reported to the best of our knowledge. Motivated by the perception of 3D components originating from 2D shapes, we have presented a method that computes the component descriptor 2to3SSC of point clouds. Based on the dissimilarity measure of the 2to3SSC, the component-oriented partial matching of point clouds can be achieved. The development of the 2to3SSC and the novel component-oriented partial matching of point clouds are two key contributions.

An interesting direction for future research is the use of deep learning to simplify the steps to compute the component descriptors.

REFERENCES

[1] S. Quan, J. Ma, F. Hu, B. Fang, and T. Ma, "Local voxelized structure for 3D binary feature representation and robust registration of point clouds from low-cost sensors," *Inf. Sci.*, vol. 444, pp. 153–171, May 2018.

[2] Y. Zhang, G. Geng, X. Wei, S. Zhang, and S. Li, "A statistical approach for extraction of feature lines from point clouds," *Comput. Graph.*, vol. 56, pp. 31–45, May 2016.

[3] Y. Sirin and M. F. Demirci, "2D and 3D shape retrieval using skeleton filling rate," *Multimedia Tools Appl.*, vol. 76, no. 6, pp. 7823–7848, May 2017.

[4] M. Schoeler, J. Papon, and F. Worgotter, "Constrained planar cuts–object partitioning for point clouds," in *Proc. CVPR*, Jun. 2015, pp. 5207–5215.

[5] Y. Zhang, K. Li, and X. Chen, S. Zhang, and G. Geng, "A multi feature fusion method for reassembly of 3D cultural heritage artifacts," *J. Cultural Heritage*, vol. 33, pp. 191–200, Sep./Oct. 2018.

[6] Y. Zou, X. Wang, T. Zhang, B. Liang, J. Song, and H. Liu, "BRoPH: An efficient and compact binary descriptor for 3D point clouds," *Pattern Recognit.*, vol. 76, pp. 522–536, Apr. 2018.

[7] T. Judd, F. Durand, and E. Adelson, "Apparent ridges for line drawing," *ACM Trans. Graph.*, vol. 26, no. 3, p. 19, 2007.

[8] Q.-X. Huang, S. Flöry, N. Gelfand, M. Hofer, and H. Pottmann, "Reassembling fractured objects by geometric matching," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 569–578, Aug. 2006.

[9] C. Weber, S. Hahmann, and H. Hagen, "Sharp feature detection in point clouds," in *Proc. Shape Modeling Int. Conf.*, Jun. 2010, pp. 175–186.

[10] P. Jenke, M. Wand, M. Bokeloh, A. Schilling, and W. Straßer, "Bayesian point cloud reconstruction," *Comput. Graph. Forum*, vol. 25, no. 3, pp. 379–388, 2006.

[11] D. Joel, L. K. Ha, T. Ochotta, and C. T. Silva, "Robust smooth feature extraction from point clouds," in *Proc. IEEE Int. Conf. Shape Modeling Appl.*, Lyon, France, Jun. 2007, pp. 123–136.

[12] 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, p. 44, Aug. 2018.

[13] L. Shapira, S. Shalom, A. Shamir, D. Cohen-Or, and H. Zhang, "Contextual part analogies in 3D objects," *Int. J. Comput. Vis.*, vol. 89, nos. 2–3, pp. 309–326, Sep. 2010.

[14] C. Yang, H. Wei, and Q. Yu, "A novel method for 2D nonrigid partial shape matching," *Neurocomputing*, vol. 275, pp. 1160–1176, Jan. 2018.

[15] A. Boulch and R. Marlet, "Fast and robust normal estimation for point clouds with sharp features," *Comput. Graph. Forum*, vol. 31, no. 5, pp. 1765–1774, 2012.

[16] H. Pottmann, J. Wallner, Q.-X. Huang, and Y.-L. Yang, "Integral invariants for robust geometry processing," *Comput. Aided Geometric Des.*, vol. 26, no. 1, pp. 37–60, Jan. 2009.

[17] S. Gumhold, X. Wang, and R. MacLeod, "Feature Extraction From Point Clouds," in *Proc. 10th Int. Meshing Roundtable*, 2001, pp. 293–305.

[18] M. Pauly, R. Keiser, and M. Gross, "Multi-scale feature extraction on point-sampled surfaces," *Comput. Graph. Forum*, vol. 22, no. 3, pp. 281–289, Sep. 2003.

[19] H. T. Ho and D. Gibbins, "Curvature-based approach for multi-scale feature extraction from 3D meshes and unstructured point clouds," *IET Comput. Vis.*, vol. 3, no. 4, pp. 201–212, Dec. 2009.

[20] M. K. Park, S. J. Lee, and K. H. Lee, "Multi-scale tensor voting for feature extraction from unstructured point clouds," *Graph. Models*, vol. 74, no. 4, pp. 197–208, 2012.

[21] Q. Merigot, M. Ovsjanikov, and L. J. Guibas, "Voronoi-based curvature and feature estimation from point clouds," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 6, pp. 743–756, Jun. 2011.

[22] J. M. Biosca and J. L. Lerma, "Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods," *ISPRS J. Photogramm. Remote Sens.*, vol. 63, no. 1, pp. 84–98, 2008.

[23] Y. Lin, C. Wang, J. Cheng, B. Chen, F. Jia, Z. Chen, and J. Li, "Line segment extraction for large scale unorganized point clouds," *ISPRS J. Photogramn. Remote Sens.*, vol. 102, pp. 172–183, Apr. 2015.

[24] M.-L. Torrente, S. Biasotti, and B. Falcidieno, "Recognition of feature curves on 3D shapes using an algebraic approach to Hough transforms," *Pattern Recognit.*, vol. 73, pp. 111–130, Jan. 2018.

[25] C. S. Chua and R. Jarvis, "Point signatures: A new representation for 3D object recognition," *Int. J. Comput. Vis.*, vol. 25, no. 1, pp. 63–85, 1997.

[26] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, May 1999.

[27] X. Li and I. Guskov, "3D object recognition from range images using pyramid matching," in *Proc. ICCV*, Oct. 2007, pp. 1–6.

[28] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2009, pp. 1848–1853.

[29] F. Tombari, S. Salti, and L. D. Stefano, "Unique signatures of histograms for local surface description," in *Proc. ECCV*, 2010, pp. 356–369.

[30] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, "BRIEF: Computing a local binary descriptor very fast," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1281–1298, Jul. 2012.

[31] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. ICCV*, Nov. 2012, pp. 2564–2571.

[32] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Gray scale and rotation invariant texture classification with local binary patterns," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany, 2000, pp. 404–420.

[33] S. M. Prakhya, B. Liu, and W. Lin, "B-SHOT: A binary feature descriptor for fast and efficient keypoint matching on 3D point clouds," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep./Oct. 2015, pp. 1929–2934.

[34] S. Srivastava and B. Lall, "3D binary signatures," in *Proc. 10th Indian Conf. Comput. Vis., Graph. Image Process.*, Dec. 2016, p. 77.

[35] J. Yang, Q. Zhang, K. Xian, Y. Xiao, and Z. Cao, "Rotational contour signatures for both real-valued and binary feature representations of 3D local shape," *Comput. Vis. Image Understand.*, vol. 160, pp. 133–147, Jul. 2017.

[36] J. Yang, Q. Zhang, Y. Xiao, and Z. Cao, "TOLDI: An effective and robust approach for 3D local shape description," *Pattern Recognit.*, vol. 65, pp. 175–187, 2017.

[37] A. Tagliasacchi, T. Delame, M. Spagnuolo, N. Amenta, and A. Telea, "3D skeletons: A state-of-the-art report," *Comput. Graph. Forum*, vol. 35, no. 2, pp. 573–597, 2016.

[38] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson, "Skeleton based shape matching and retrieval," in *Proc. Shape Modeling Int.*, May 2003, pp. 130–139.

[39] M. Natali, S. Biasotti, G. Patanè, and B. Falcidieno, "Graph-based representations of point clouds," *Graph. Models*, vol. 73, no. 5, pp. 151–164, 2011.

[40] A. Sobiecki, A. Jalba, and A. Telea, "Comparison of curve and surface skeletonization methods for voxel shapes," *Pattern Recognit. Lett.*, vol. 47, pp. 147–156, Oct. 2014.

[41] P. K. Saha, G. Borgefors, and G. S. di Baja, "A survey on skeletonization algorithms and their applications," *Pattern Recognit. Lett.*, vol. 76, pp. 3–12, Jun. 2016.

[42] C. Arcelli, G. S. di Baja, and L. Serino, "Distance-driven skeletonization in voxel images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 4, pp. 709–720, Apr. 2011.

[43] J. Ma, S. W. Bae, and S. Choi , "3D medial axis point approximation using nearest neighbors and the normal field," *Vis. Comput.*, vol. 28, no. 1, pp. 7–9, Jan. 2012.

[44] A. C. Jalba, J. Kustra, and A. C. Telea, "Surface and curve skeletonization of large 3D models on the GPU," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1495–1508, Jun. 2013.

[45] H. Huang, S. Wu, D. Cohen-Or, M. Gong, H. Zhang, G. Li, and B. Chen, "L_1-medial skeleton of point cloud," *ACM Trans. Graph.*, vol. 32, no. 4, p. 65, Jul. 2013.

[46] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. CVPR*, Jul. 2017, pp. 652–660.

[47] A. Golovinskiy and T. Funkhouser, "Consistent segmentation of 3D models," *Comput. Graph.*, vol. 33, no. 3, pp. 262–269, Jun. 2009.

[48] Q. Huang, V. Koltun, and L. Guibas, "Joint shape segmentation with linear programming," *ACM Trans. Graph.*, vol. 30, no. 6, p. 125, Dec. 2011.

[49] L. Yi, L. Guibas, A. Hertzmann, V. G. Kim, H. Su, and E. Yumer, "Learning hierarchical shape segmentation and labeling from online repositories," *ACM Trans. Graph.*, vol. 36, no. 4, p. 70, Jul. 2017.

[50] J. Xie, Y. Fang, F. Zhu, and E. Wong, "Deepshape: Deep learned shape descriptor for 3D shape matching and retrieval," in *Proc. CVPR*, Jun. 2015, pp. 1275–1283.

[51] E. Kalogerakis, M. Averkiou, S. Maji, and S. Chaudhuri, "3D shape segmentation with projective convolutional networks," in *Proc. CVPR*, Jul. 2016, pp. 3779–3788.

[52] L. Yi, H. Su, X. Guo, and L. J. Guibas, "SyncSpecCNN: Synchronized spectral CNN for 3D shape segmentation," in *Proc. CVPR*, Jul. 2017, pp. 2282–2290.

[53] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong, "O-CNN: Octree-based convolutional neural networks for 3D shape analysis," *ACM Trans. Graph.*, vol. 36, no. 4, p. 72, Jul. 2017.

[54] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Is object localization for free?—Weakly-supervised learning with convolutional neural networks," in *Proc. CVPR*, Jun. 2015, pp. 685–694.

[55] R. G. Cinbis, J. Verbeek, and C. Schmid, "Weakly supervised object localization with multi-fold multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 189–203, Jan. 2017.

[56] S. Muralikrishnan, V. G. Kim, and S. Chaudhuri, "Tags2Parts: Discovering semantic regions from shape tags," in *Proc. CVPR*, Jun. 2018, pp. 2926–2935.

[57] A. M. Bruckstein, N. Katzir, M. Lindenbaum, and M. Porat, "Similarity-invariant signatures for partially occluded planar shapes," *Int. J. Comput. Vis.*, vol. 7, no. 3, pp. 271–285, 1992.

[58] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002.

[59] H. Ling and D. W. Jacobs, "Shape classification using the inner-distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 286–299, Feb. 2007.

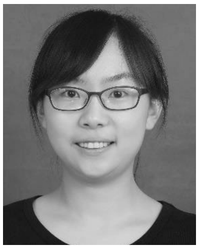
[60] C. Zhao, S. S. F. Chan, W.-K. Cham, and L. M. Chu, "Plant identification using leaf shapes—A pattern counting approach," *Pattern Recognit.*, vol. 48, no. 10, pp. 3203–3215, 2015.

[61] L. Chang, M. Arias-Estrada, J. Hernández-Palancar, and L. E. Sucar, "Partial shape matching and retrieval under occlusion and noise," in *Proc. Iberoamer. Congr. Pattern Recognit.*, 2014, pp. 151–158.

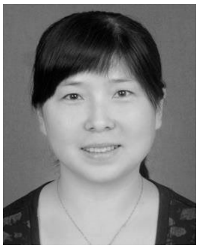
[62] Y. Lipman, D. Cohen-Or, D. Levin, and H. Tal-Ezer, "Parameterization-free projection for geometry reconstruction," *ACM Trans. Graph.*, vol. 26, no. 3, p. 22, Jul. 2007.

[63] H. J. Wolfson, "On curve matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 5, pp. 483–489, May 1990.

[64] L. J. Latecki, R. Lakamper, and T. Eckhardt, "Shape descriptors for non-rigid shapes with a single closed contour," in *Proc. CVPR*, Jun. 2000, pp. 424–429.



YUHE ZHANG received the B.S. and Ph.D. degrees in computer applied technology from Northwest University, Xi'an, China, in 2012 and 2017, respectively. Since 2017, she has been a Lecturer with the School of Information Science and Technology, Northwest University. Her research interests include computer graphics, image processing, intelligent information processing, and the digital restoration of cultural heritage.



XIAONING LIU was born in Mei, Shaanxi, China, in 1978. She received the B.S., M.S., and Ph.D. degrees from Northwest University of China, in 2000, 2003, and 2006, respectively.

From 2003 to 2006, she was an Assistant of Data Structure with the Department of Computer, Northwest University. From 2007 to 2009, she was a Lecturer. From 2016 to 2017, she was a Researcher with The University of Sheffield, U.K. Since 2010, she has been an Associate Professor with the School of Information Science and Technology. She is the author of more than 50 articles and three inventions. Her research interests include point cloud data processing, digital protection of cultural heritage, and machine learning.

Dr. Liu's awards and honors include the second award of Science and Technology Award in Colleges and Universities of China, the second award of National Teaching Achievement Award of China.

CHUNHUI LI will receive the B.S. degree in software engineering from Northwest University, in 2020. From 2018 to 2019, he was with the Visualization Technology Institute of Northwest University. His research interests include computer graphics, image processing, intelligent information processing, and the digital restoration of cultural heritage. In 2019, he won the Meritorious Winner Award in the MCM/ICM Competition.

JIABEI HU was born in Xianyang, Shaanxi, China. He received the B.S. degree in computer science and technology from Yan'an University, in 2017. He is currently pursuing the master's degree in computer application technology with the Northwest University of China.

Since 2017, he has been a Graduate Student with the Visualization Technology Institute of Northwest University. He has published four articles that are indexed by EI. His research interests include computer graphics and image processing.

GUOHUA GENG received the Ph.D. degree in computer software and theory from Northwest University, Xi'an, China, in 2004. Since 2002, she has been a Professor with the School of Information Science and Technology, Northwest University. Her research interests include image processing, intelligent information processing, and the digital restoration of cultural heritage.

SHUNLI ZHANG received the B.S. degree in applied mathematics from Xidian University, Xi'an, China, in 1997, and the M.S. and Ph.D. degrees in aeronautical and astronautical manufacturing engineering from Northwestern Polytechnical University, Xi'an, in 2004 and 2010, respectively.

From 2011 to 2014, he was a Postdoctoral Researcher with Northwestern Polytechnical University. Since 2014, he has been a Professor with the School of Information Science and Technology, Northwest University, Xi'an. His research interests include computed tomography, image processing, biomedical imaging, and parallel computing.