

## Skeleton-Driven 3D Model Segmentation Algorithm

Menglu Gao<sup>1, a</sup>, Jingbin Hao<sup>1, b</sup>, YINUO Zhou<sup>1, c</sup>, Zewen Cao<sup>1, d</sup>, Xin Li<sup>1, e</sup>

<sup>1</sup>School of Mechatronic Engineering, China University of Mining and Technology, Jiangsu 221116, China.

<sup>a</sup>supoppy@126.com, <sup>b</sup>jingbinhao@cumt.edu.cn, <sup>c</sup>909076304@qq.com, <sup>d</sup>18779393872@163.com, <sup>e</sup>15651358951@163.com

### Abstract

CNC machines and 3D printers are commonly used machining equipment, which are both limited to platform size and tool shape. A large complex part must first be divided into a group of smaller and simpler sub parts, which can be machined separately for splicing. The model segmentation is a key influent factor for machining efficiency and splicing quality of large complex models. In this paper, a skeleton-based algorithm is proposed to segment a large complex model. Firstly, the skeleton of a 3D model is obtained by using the curve skeleton extraction method based on the pseudo normal vector field. Then, the priority access order of each branch is determined in the skeleton, and the key segmentation points are found to construct the segmentation boundary for adjacent branches. Based on the minimization principle, the meaningful segmentation can be generated by the segmentation boundaries. Several experimental models are used to illustrate the efficiency of the proposed algorithm. This research work provides an effective solution to improve the machining capacity and reduce the difficulty of large complex parts machining.

### Keywords

3D model segmentation; curve skeleton extraction; branch access order; segmentation boundary; large complex parts.

### 1. Introduction

The advantage of composite manufacturing technology is that any complex 3D model can be transformed into a high-precision and high-quality physical model in a short time. However, when the structure is too complex and the volume is too large (beyond the material size or machine tool processing range), it will inevitably cause overall processing difficulties or processing failure. At present, a good way to solve this kind of problems is to decompose a closed 3D mesh polyhedron or a directional 2D manifold into a set of 3D model sub-blocks with limited number, simple shape meanings and respective connectivity according to its surface geometry and topological characteristics [1]. In short, it is divided into smaller and simpler sub models according to the surface characteristics of the models.

Model segmentation technology can solve the problems of complex structure, large volume (beyond material size or machine tool processing range), low material utilization, and can effectively reduce the requirements of processing equipment and improve productive efficiency [2]. As model segmentation has important applications in point cloud mesh surface reconstruction, mesh subdivision and simplification, geometric compression, interactive editing, texture mapping, geometric deformation, local area parameterization and so on, it has become a hot topic in academia and industry in recent years. For the rapid manufacturing of large-scale or 3D models, model segmentation is not only to divide large models into a group of small models whose size meets the processing requirements, but also to divide them into block

models with simple shape, convenient processing and assembly, so as to improve the ability and efficiency in the processing of rapid prototyping equipment.

In this paper, aiming at the requirement of meaningful model segmentation, we have proposed a skeleton-driven 3D model segmentation algorithm. In this study, our purpose is to segment complex models that cannot be precisely fabricated directly into several sub-models that is simpler and smaller. For this purpose, the major considerations for realizing the model segmentation algorithm are presented with the help of the skeleton of model, including skeleton extraction, determination of the priority access order for each skeleton branch, identification of the key segmentation points for segmentation and the final segmentation boundary extraction. At last, seven different 3D models that have complex features are partitioned into a series of sub-models. The experimental results demonstrate the feasibility of the proposed algorithm. Our research work has the potential applicability to precision fabrication of complex or large-scale parts.

The rest of this paper is organized as follows: Section 2 provide a literature review related to 3D model segmentation method. Section 3 offers a brief introduction of the proposed algorithm and introduces the skeleton extraction method. Section 4 develops methods to find the priority access order for each skeleton branch and to identify the key segmentation points for segmentation. Section 5 presents the extraction of final segmentation boundary. Section 6 shows model segmentation experiments and experimental results, and finally, Section 7 concludes and depicts future directions.

## 2. Organization of the Text

### 2.1. Related Work

In this section, we review some approaches on 3D model segmentation. Nowadays, the common segmentation methods are mainly divided into watershed-based segmentation algorithm, clustering-based segmentation algorithm and skeleton extraction-based segmentation algorithm.

Serra [3] proposed a watershed method for image segmentation. Its basic idea is to treat the image as a topographic map. The gray value of the pixels in the image represents the "altitude" of the point. The local minimum and its affected area in the image are called catchment basins, while the local maximum and its influence are called peaks or highland, and the boundaries between them form watersheds, so that the image is divided into interconnected independent regions by these boundaries. Mangan [4] proposed a watershed segmentation algorithm for triangular mesh surfaces. Page et al. [5] created a robust greedy watershed algorithm for mesh model segmentation, FMW (Fast Marching Watersheds). With the local principal curvature as the height function, a directional height map is obtained, which follows the minimum rule and improves the initial identification set of the watershed algorithm. Garland et al. [6] proposed a hierarchical patch clustering method, which uses the dual relationship between patches to iteratively merge surface meshes, thus separating flat model surfaces. Israeli scholar Katz et al. [7] proposed a hierarchical segmentation algorithm based on fuzzy clustering, which can divide a given mesh into meaningful components, solve the problem of over-segmentation and boundary sawtooth, and add different distance functions, different capacity functions and non-geometric feature factors into the algorithm to expand the application scope of the algorithm.

The skeleton structure shape description method based on geometric and topological information has been well applied in many places [8]. Hoppe [9] extracted the skeleton tree of the model based on progressive mesh algorithm effectively. Lazarus [10] and Verroust [11] proposed a level set algorithm to extract the skeletal axis from the vertex set of the polygon model and segment the model at the skeletal joints. Xiao et al. [12] presented a method of topological segmentation of the 3D human body scanning model based on the discrete Reeb

map of the point cloud of the three-dimensional human body scanning. Shi et al. [13] constructed Reeb diagrams from the characteristic functions of Laplace transform to calculate the skeleton of the topological structure. Luca et al. [14] proposed a skeleton-based model segmentation algorithm. By using the concept of influence region, the skeleton of a curve can be divided into meaningful branches and meaningless branches. Only meaningful skeleton branches can be used for model segmentation, and then meaningful model components can be obtained.

Nevertheless, most of the above work primarily emphasize on developing segmentation algorithm itself, not whether the segmented sub-models are suitable for fabrication. Or they focus more on achieving good segmentation results, not low computational complexity. Or they can develop segmentation method with short computational time, but final segmentation is not effective. In this paper, we propose a skeleton-driven segmentation algorithm for 3D model to effectively achieve model segmentation in a low computational complexity.

## 2.2. Overview of the Segmentation Algorithm

The skeleton diagram can measure the geometric and topological features of the 3D model, which can be good used for model segmentation. The proposed algorithm mainly contains the following four steps.

Step1 The method of curve skeleton extraction based on pseudo - law vector field is used to obtain the skeleton of 3D model.

Step2 Determine the priority access order for each branch in the skeleton.

Step3 Find the key segmentation points for segmentation.

Step4 Extract the final segmentation boundary.

In the whole algorithm, Step1 is the robust skeleton with good single connectivity to extract 3D models, which is also the basis of driving 3D model segmentation. In addition, skeleton extraction can be realized by curve skeleton extraction algorithm based on pseudo-method vector field; Step2 is to regard a complete skeleton as a set consisting of a series of skeleton branches, and the priority access sequence of skeleton branches is determined to ensure that the small but important features in the 3D model can be accessed and segmented first; Step3 is based on the topological and geometric properties carried by skeleton branches to find the key points that can be separated from two different sub-models. In this step, the skeleton branch is discretized into a set of skeleton points; Step4 is to extract the segmentation boundary in the concave area of the model based on the minimization principle, and finally produce meaningful segmentation.

In step1, in order to accurately extract the curve skeleton of the 3D model, a curve skeleton extraction algorithm based on pseudo-normal vector field is used [15].

Curve skeleton extraction based on pseudo-normal vector field can be divided into three steps:

Step-A Creating a pseudo-normal vector field inside the three-dimensional voxel model;

Step-B In the created pseudo-normal vector field, three types of growth points are calculated: key points, low-dispersion control points, and high-curvature boundary points;

Step-C Based on the above three types of growth points, the skeleton growth algorithm is used to extract the curve skeleton.

Step-C.1: Tracking the pseudo-vector field at a high curvature boundary point until the low-dispersion control point or the previous access position, the tracking stops, and the generated tracking path captures the skeleton branch of the model prominently;

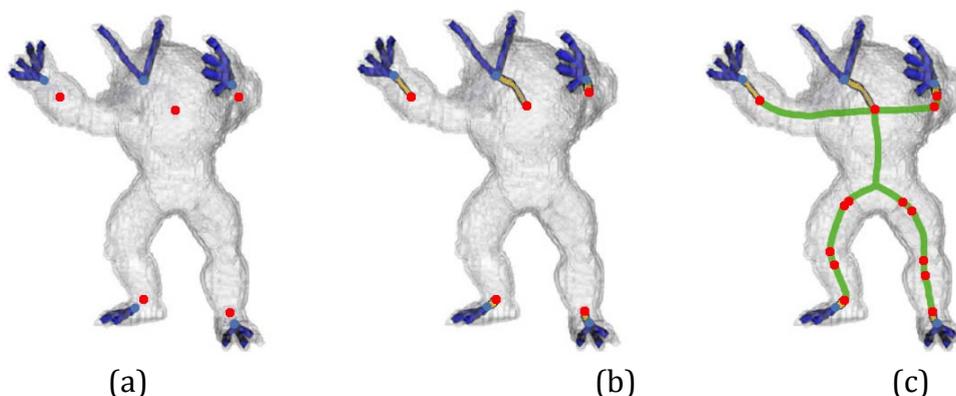
Step-C.2: Starting to track the pseudo-vector field with a low divergence control point until the key point or the previously accessed position, the tracking stops, and the generated tracking path captures the skeleton branch of the sub-center portion of the voxel model;

Step-C.3: Start tracking the pseudo-vector field with key points until another key point, the tracking stops, and the resulting tracking path captures the skeleton branch of the central part of the three-dimensional voxel model.

The entire step-C process will eventually produce a correct, complete curve skeleton. Among them, the skeleton branch generated by step-C.3 is the core skeleton of the three-dimensional voxel model, which can be regarded as composed of all key points, and can be regarded as the first layer of the curve skeleton; Second, the skeleton branches generated by step-C.3 and step-C.2 have a higher complexity skeleton than the first layer skeleton, and the skeleton can be regarded as composed of all key points and low divergence control points, that is the second layer of the curve skeleton; At last, the skeleton branches generated by step-C.1, step-C.2, and step-C.3 have a higher complexity skeleton than the second layer skeleton. The skeleton can be regarded as all key points, low divergence control points, and high curvature. The boundary point is formed, that is, the third layer of the curve skeleton.

In the whole algorithm, Step-A is the pseudo-vector vector field inside the 3D model. Let  $O \subset E^3$  be a three-dimensional voxel mesh model,  $v: E^3 \rightarrow R^3$  is a vector function, which is used to calculate the vector  $\vec{v} = v(p)$  of each mesh voxel. The pseudo-vector field of the three-dimensional model is from the voxel model voxel (or its subset) pseudo-normal vector composition. Step-B is to determine three types of growth points (key points, low divergence points, and high curvature boundary points) that affect the complexity of the skeleton. Step-C based on the above three types of growth points, according to the skeleton growth algorithm, sequentially captures the skeleton branches of the 3rd, 2nd and 1st layers of the 3D voxel model, and its complexity gradually increases.

With the help of the curve skeleton extraction algorithm, the skeleton of one example is extracted effectively and precisely, as shown in Fig. 1.



**Fig 1.** The extraction of one example. (a) Extracted skeleton branches (blue lines) using high curve boundary points. (b) Extracted skeleton branches (yellow lines) using low divergence control points. (c) Extracted skeleton branches (green lines) using critical points.

### 2.3. Priority Access Sequence of Skeleton Branch and Critical Segmentation Points

In this section, priority access sequence of skeleton branch is first determined by defined priority function, which can ensure that the small but important features in the 3D model can be accessed and segmented first. Next, the critical segmentation points are identified with the help of topology and geometry of skeleton.

#### 2.3.1. Computation of Priority Access Sequence of Skeleton Branch

According to the number of adjacent skeleton points, a skeleton point can be divided into the following three types:

- (1) Termination point, there is only one adjacent skeleton point;
- (2) Common points, there are only two adjacent skeleton points;
- (3) Branch points, there are more than two adjacent skeleton points.

In this article, termination point and branch point are considered as feature points, and any subset of skeleton formed by feature points at both ends is called skeleton branch.

The segmentation algorithm proposed in this paper detects the initial critical segmentation points by measuring the related geometric and topological properties, and this algorithm does not consider the sub-model which has been divided in the subsequent calculation of key segmentation points. By determining the access order of skeleton branches, the small but important sub-models must be isolated, rather than relegated to a large sub-model. This sequence allows to extract small but important sub-models first. And there are three criteria for finding the best skeleton branch: Branch type, Branch length, and Branch center.

Each skeleton branch can be thought of as consisting of a set of skeleton points located above it. In other words, the skeleton branches are separated into a series of skeleton points. The priority  $P(b)$  for defining skeleton branches is as follows:

$$P(b) = Type(b) + \frac{1}{Length(b)} \sum_{i \in b} C(p) \quad (1)$$

Where,  $Type(b)$  is the type value of the skeleton branch;  $Length(b)$  is the length of the skeleton branch;  $\sum_{i \in b} C(p)$  is the sum of the normalization centers of all skeleton points on the skeleton branch.

Among them, the type value of skeleton branch  $b$  is determined by the type of its two endpoints:

- (1) The skeleton branch with two branching points has a low weight value (the weight value is 0) and is called a low-type skeleton branch.
- (2) The type weights of skeleton branches with one endpoint and one branch point belong to medium weight (weight value is 1), and are called medium type skeleton branches.
- (3) A skeleton branch with two endpoints has a high type weight (the weight value is 2), which is called a high-type skeleton branch.

The length of skeleton branch  $b$  is defined as the total number of skeleton points above it.

The central value of the skeleton point  $p$  is defined as the average hop from the skeleton point  $p$  to all points on the skeleton. In the grid, let  $\max H$  represent the maximum average hop of all skeleton points, namely:

$$\max H = \max_p (avgH(p)) \quad (2)$$

The following formula is used to the normalize the central value of the skeleton point  $p$ :

$$C(p) = avgH(p) / \max H \quad (3)$$

After the grid is split using the critical cut-point in a selected skeleton branch, then the center value of all currently computed skeleton points will no longer be valid for subsequent

segmentation. That is, when other skeleton branches are processed later, the center value of skeleton points will be recalculated.

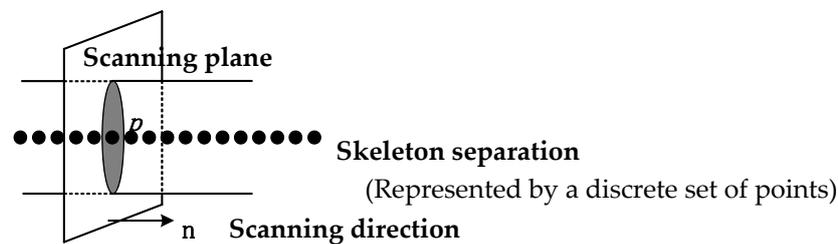
The central value of skeleton branch  $b$  is defined as the sum of the central value of all skeleton points  $P$  on the segment branch, and the central value of all skeleton points  $P$  on the segment branch is  $C(p)$ . By calculating the priority  $P(b)$  of the skeleton branch, it can be guaranteed that the small but important sub-features can be divided first to achieve meaningful segmentation.

### 2.3.2. Identification of Critical Segmentation Point

The critical segmentation point refers to the important point that can divide two sub-models with obviously different shapes in the model. Moreover, the geometric and topological properties of grid model can distinguish one sub-model from other sub-models in grid segmentation. Therefore, with the help of these two properties, key segmentation points can be found. The search process is roughly divided into two steps: Firstly the topology properties are used to scan the given grid model; Secondly, the candidate key segmentation points are determined by geometric properties.

#### (1) Space Scanning

The spatial scanning method in computational geometry is used to scan the given grid model along the topology path represented by the skeleton. Where, the spatial scanning plane is perpendicular to the skeleton branch, and the scanning movement direction is pointed from the starting point to the ending point. The schematic diagram of spatial scanning is shown in Fig. 2.



**Fig 2.** The schematic of space sweep

Let  $b$  be the selected skeleton branch, and perform different scanning operations on different types of skeleton branch  $b$ . Details are as follows:

If  $b$  is a medium-sized skeleton branch, scan it from the endpoint  $P_{start}$  to another point  $P_{end}$ . Some points in the region near the point  $P_{end}$  are excluded from the scan because there are no valid key points in the region. The area near the point  $P_{end}$  is defined as a spherical space with its center as the point  $P_{end}$  and the radius as the minimum distance from the point  $P_{end}$  to the nearest vertex on the surface;

If  $b$  is a high or low type skeleton branch, scan from a point with a smaller cross-sectional area.

#### (2) Geometric Function

For the selected skeleton branch  $b$ , the cross-sectional area at each skeleton point  $P$  on the scanning path from  $P_{start}$  to  $P_{end-1}$  is calculated, and then the cross-sectional area is used to define the geometric function  $G(p)$ .

$$G(p) = \frac{AreaCS(p+1) - AreaCS(p)}{AreaCS(p)}, p \in [p_{start}, p_{end-1}] \tag{4}$$

In the above formula,  $AreaCS(p)$  is the cross-sectional area at the skeleton point  $P$ .  $AreaCS(p+1)$  is the cross-sectional area at the point  $(p+1)$  of the skeleton.

The cross section area at the skeleton point  $P$  is defined as the sum of the number of voxel intersections with the spatial scanning plane at the current skeleton point  $P$ . This definition can accelerate the calculation of the cross section area at each skeleton point  $P$ .

Connecting the calculated  $AreaCS(p)$  value and  $G(p)$  value respectively, a polygon curve can be obtained, which is presented in the form of pulse fluctuation. The three sample distribution curves of  $AreaCS(p)$  and  $G(p)$  are shown in Fig. 3. In the  $G(p)$  curve, the following two intersections are identified as candidate critical segmentation points:

- (1) If the rising edge of an impulse passes through the  $P$  axis and its trough is less than -0.15 (as shown in Fig. 3(a) or Fig. 3(c)), the intersection point  $t$  will be used as the candidate key segmentation point for the grid model segmentation;
- (2) If the peak value of a positive pulse is higher than the threshold value of 0.4 (as shown in Fig. 3(b)), the intersection point  $t$  will also be selected as the candidate critical segmentation point.

Among them, the intersection point  $t$  corresponds to the skeleton point  $P$  in the skeleton branch.

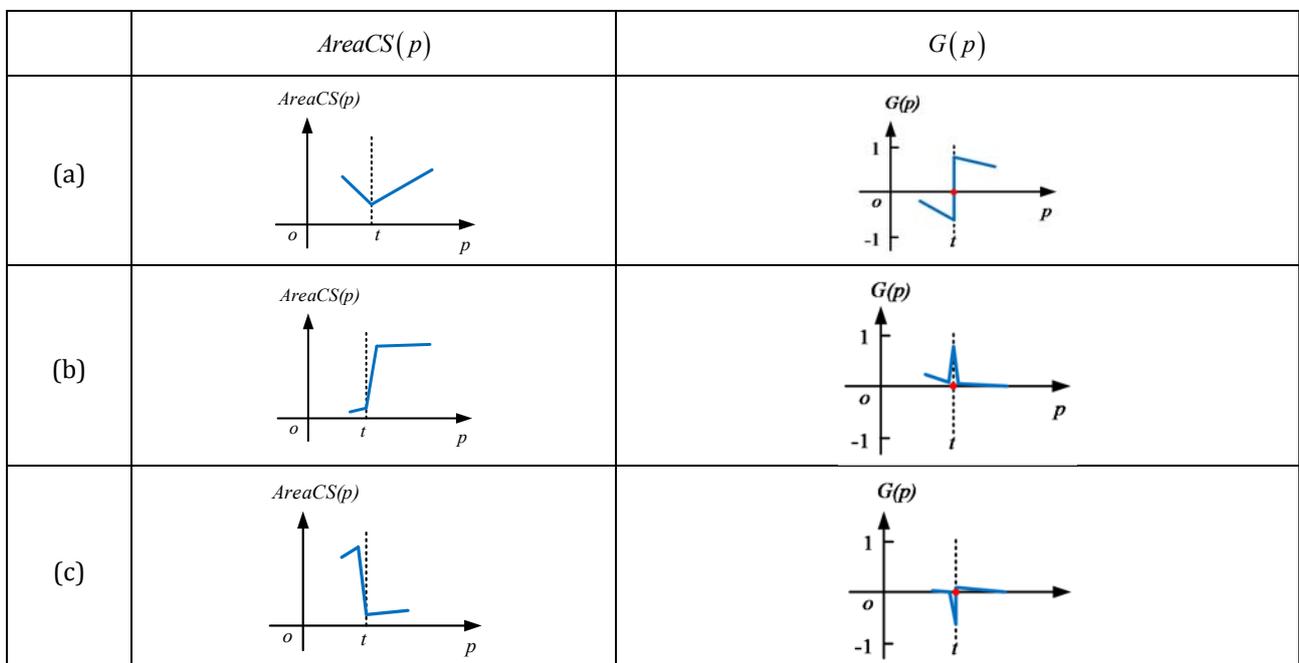


Fig 3. Three kind sample profiles of  $AreaCS(p)$  and  $G(p)$

### 2.4. Segmentation Boundary

When the 3D grid model is segmented only with the key segmentation points, the smooth segmentation boundary may not be obtained. In order to solve this problem, the feature contour in the concave region is used as the final segmentation boundary for the model segmentation.

### 2.4.1. Split Restricted Zone Construction

For each critical segmentation point, the corresponding segmentation restricted region is constructed. Split restricted area construction is shown in Fig. 4. The segmentation restricted region is composed of two parallel planes, the normal vector of the plane is the same direction as the corresponding key segmentation point, and the two planes keep the same distance from the key segmentation point, with the distance Threshold  $d$ .

$$d = \begin{cases} 2 * \sigma, & \text{if } \sigma > LNG_{edge} \\ 2 * LNG_{edge}, & \text{else} \end{cases} \tag{5}$$

In the above formula,  $\sigma$  is the distance between two adjacent skeleton points;  $LNG_{edge}$  is the average length of the grid.

The segmentation restricted region constrains the extraction range of the final segmentation boundary. Outside the segmentation restricted region, no segmentation boundary is considered to be established, which accelerates the search of the rough region of the segmentation boundary, so as to improve the efficiency of mesh model segmentation.

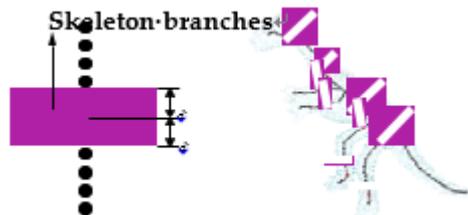


Fig 4. The restricting zone of segmentation construction

### 2.4.2. Concave Region Extraction

The minimization rule in cognitive theory indicates that human perception usually divides the model into a series of sub-models along the concave and discontinuous tangent plane [16]. Inspired by this rule, the algorithm proposed in this paper tends to segment the given grid along the concave region, and the final segmentation boundary is located in the concave region.

#### (A) Estimation of Vertex Discrete Curvature in Grid Model

The local concavo-convex or bended condition of 3D surface can be represented by its discrete curvature. The Gaussian curvature of any triangular grid vertex can be estimated flexibly by using Voronoi graph method. The basic idea is to take each vertex of triangular grid as a small neighborhood and the average curvature of the neighborhood as the curvature of the vertex.

Firstly, any edge  $e_j$  in the triangular grid is considered to be expressed by its two endpoints,  $p_i$  and  $p_j$ , namely,  $e_j = p_i p_j$ ; Secondly, the Angle between  $e_j$  and  $e_{j+1}$  is called  $\alpha_j$ , namely,  $\alpha_j = \angle(e_j, e_{j+1})$ . Finally,  $\Delta p_i p_j p_{j+1}$  of the triangle formed by  $e_j$  and  $e_{j+1}$  is denoted as  $T_j$ , and the normal vector  $n_j$  of this triangle can be calculated by the following formula:

$$n_j = \frac{e_j \times e_{j+1}}{\|e_j \times e_{j+1}\|} \tag{6}$$

Look for two adjacent triangular faces  $T_j$  and  $T_{j+1}$ , and determine their normal vectors  $n_j$  and  $n_{j+1}$ , and write the Angle between them as  $\beta_j$ , namely,  $\beta_j = \angle(n_j, n_{j+1})$ ; Next, the projected area of vertex  $p_i$  is denoted as  $A_M$ . The definition of  $\alpha_j, \gamma_j, \lambda_j$  and  $A_M$  are shown in Fig. 5.

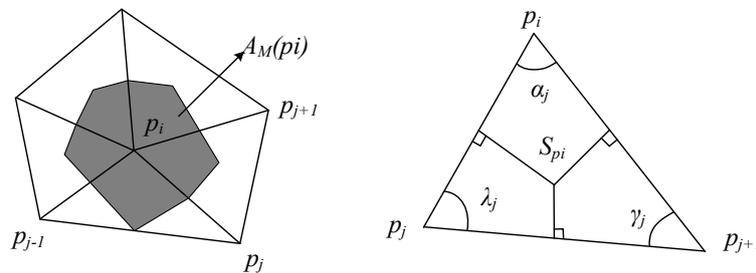


Fig 5. Definition of  $\alpha_j, \gamma_j, \lambda_j$  and  $A_M$

The Gaussian curvature  $K$  of any triangular grid vertex can be calculated by the following formula:

$$K(p_i) = \frac{2\pi - \sum_j \alpha_j}{A_M} \tag{7}$$

The formula for calculating the mean curvature  $H$  is as follows:

$$H(p_i) = \frac{1}{4A_M} \sum_{j=1}^n \|e_j\| \beta_j \tag{8}$$

(B) The Division of Vertex and Surface Attributes Based on Curvature Characteristics

Gaussian curvature  $K$  and average curvature  $H$  can determine whether the surface is concave or convex at some point. The types of points on the surface can be divided into three categories: when  $K < 0$ , this point is a hyperbolic point; When  $K = 0$ , it's a parabolic point; When  $K > 0$ , this point is an ellipse point. Furthermore, the type of face plate can be determined by the positive negative values of  $K$  and  $H$  and normal vectors of vertex and first-order neighborhood vertex. The attribute division of face slice is shown in Fig. 6.



(a) Convex and concave vertex (b) Convex and concave facet (c) Saddle facet

Fig 6. Facet property classification

(1) Convex surface area. If the Gaussian curvature of the vertex and its neighboring vertex are both positive, the point is convex. In Fig. 6(a), the point marked by blue is ellipse point; In Fig. 6(b), the area marked blue-green is the convex surface area.

(2) Concave surface area. If the Gaussian curvature of the vertex and its neighboring vertex are both negative, this point is a concave point. In Fig. 6(a), the point marked red is a hyperbolic point; In Fig. 6(b), the area marked pink is the area of concave surface.

(3) Ridged surface area. If one side of the vertex and its neighboring vertex is the zero value point of Gaussian curvature and one side is the positive value point of average curvature, and the Angle of the triangular normal vector of the Shared edge exceeds a certain threshold, this point is convex.

(4) Saddle surface area. If the Gaussian curvature and average curvature of the vertex and its neighboring vertex are negative, the point is concave, and the local area is mostly concave and a few convex. The faces marked pink in Fig. 6(c) are the saddle-facing areas.

(5) Flat area. If the Gaussian curvature and average curvature of the vertex and its neighboring vertex are 0, the point is flat.

### (C) Concave Region Extraction Method

The extracted segmentation boundary should be in the concave region, and the concave region of the grid model can be found by the Gaussian curvature  $K$  and the average curvature  $H$  of the grid vertex. In addition, because discrete curvature is not robust to noise, it is easy to produce minute segmentation results when concave surface is partitioned. Concave regions are optimized by virtue of neighborhood area of grid vertex and curvature distribution, and concave sets with less data are deleted. The main steps of the concave region extraction algorithm include the following three steps:

#### (1) Calculating the Set of Pits

The vertex set  $3DVertex$  of the triangular mesh model is input, and the Gaussian curvature  $K$  and average curvature  $H$  of each vertex  $v_i$  are calculated through the discrete curvature estimation method of the grid vertex, and the first-order neighborhood vertex  $FirstNeighborhoodVertex$  of each vertex  $v_i$  is calculated; If the Gaussian curvature  $K_{FirstNeighborhoodVertex}$  of all vertices in the first-order neighborhood vertex  $FirstNeighborhoodVertex$  of a vertex  $v_i$  and the average curvature  $H$  of vertex  $v_i$  satisfy:

$$\left(K_{FirstNeighborhoodVertex} < 0\right) \text{ and } (H < 0) \quad (9)$$

So add vertex  $v_i$  to  $ConcaveRegion$ .

#### (2) Regional Extension

To begin with, the area  $A_{ConcaveRegion}^{v_i}$  of concave triangular facets formed by vertex  $FirstNeighborhoodVertex$  in concave region  $ConcaveRegion$ . Secondly, the area  $A_{FirstNeighborhoodVertex}^{v_i}$  formed by all vertices in the first-order neighborhood of vertex  $v_i$  is calculated. If

$$\frac{A_{ConcaveRegion}^{v_i}}{A_{FirstNeighborhoodVertex}^{v_i}} \geq threshold1 \quad (10)$$

So add the first-order adjacency vertex  $FirstNeighborhoodVertex$  of vertex  $v_i$  to  $ConcaveRegion$ .

#### (3) Regional Delete

The region deletion is to calculate all the connected branches  $ConnectedBranch$  in  $ConcaveRegion$ , and calculate the number of vertices of each connected branch  $NumberOfConnectedBranchVertex$  and the average Gaussian curvature  $K_{ConnectedBranch}$  of the connected branches  $ConnectedBranch$ . If

$$\begin{cases} NumberOfConnectedBranchVertex < threshold2 \\ K_{ConnectedBranch} > threshold3 \end{cases} \quad (11)$$

So remove the connected branch from  $ConcaveRegion$ .

The goal of region extension and region deletion is to optimize the concave set, which can better reflect the regional characteristics of the model. Next, the segmentation boundary is extracted in the optimal concave set and used for model segmentation. Note that if the set of concave points is not found in the segmentation restricted region, the key segmentation points in this region are deleted and the segmentation operation does not occur.

### 2.4.3. Split Boundary Extraction

By connecting the vertex in the concave set according to the topological relation, the connected edge can be used as the partition boundary of the model. However, due to too many vertices, the resulting connections may be irregular and not unique. Therefore, in this article, these concave join edges are refined to make them a single segmentation ring boundary, which passes through the model surface and has the shortest path.

The segmentation boundary extraction mainly consists of three steps:

- (1) According to the topological relation, the set of points in the concave region is constructed as the region expressed by the triangular faces, and these faces are adjacent to each other without crossing or overlapping;
- (2) Classify each connected edge: Boundary-Edge and Interior-Edge;
- (3) The edge of BoundaryEdge is deleted iteratively, and the concave region is finally transformed into a single partition boundary by refinement method, which is used for model segmentation.

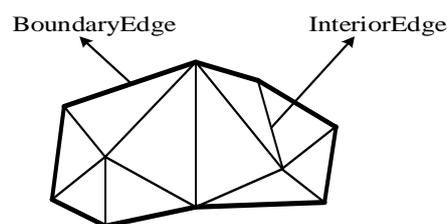


Fig 7. The edge classification

Divide all connected edges into two categories: one is  $BoundaryEdge$ ; The other one is  $InteriorEdge$ . The detection method of  $BoundaryEdge$  is: Take a connecting edge  $e$ , which has 2 adjacent triangles,  $t_1$  and  $t_2$ . If all the sides of  $t_1$  and  $t_2$  are in the concave region, then  $e \notin BoundaryEdge$ ; Otherwise, if there is an edge in  $t_1$  and  $t_2$  that is not in the concave region,  $e \in BoundaryEdge$ . The edge classification is shown in Fig. 7. In the figure, the roughened edge is the  $BoundaryEdge$  edge in the concave region, while the unbolded edge is the  $InteriorEdge$  edge in the concave region.

After classifying all the edges in the concave region, the concave region is refined. The goal is to obtain a single connected feature line, which can express the concave region. The refinement

idea is to delete the **BoundaryEdge** edge from the concave region. Fig. 9 shows the edge removal operation. If the edge in **BoundaryEdge** is deleted, it will change the edge originally located in **InteriorEdge** to **BoundaryEdge** edge, as shown in Fig. 8(a). According to this method, the iterative refining operation is performed until the edge in **InteriorEdge** is empty, that is, all remaining edges are inside **BoundaryEdge**, and the refinement ends. In order to ensure that the resulting edge chain is connected, the thickened edge in Fig. 8(b) cannot be deleted, because deleting it will result in no connectivity. The thickened edges in Fig. 8(c) can be removed.



(a)BoundaryEdge deletion (b)The edge that cannot be deleted (c)The edge that can be deleted

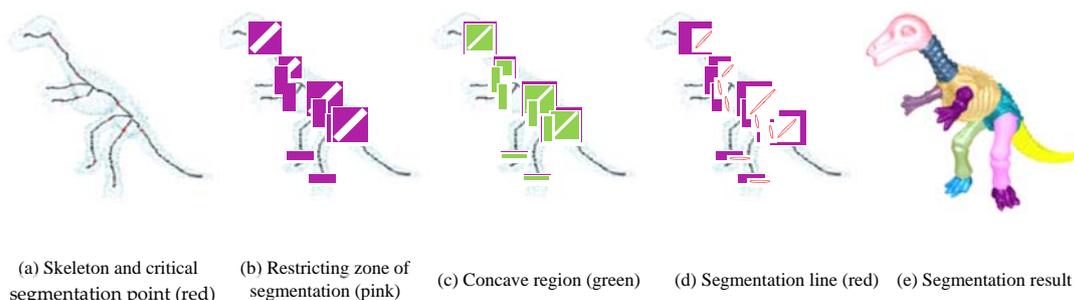
**Fig 8.** The edge deletion operation

The refining process of concave region is divided into the following three steps:

- (1) All edges in **ConcaveRegion** are classified into **BoundaryEdge** edge and **InteriorEdge** edge;
- (2) Delete **BoundaryEdge** edge from **ConcaveRegion** until **InteriorEdge** is empty. If there is edge  $e$  in **InteriorEdge**, if  $e$  is in **BoundaryEdge**, it can **BoundaryEdge** judged whether there will **BoundaryEdge** no connectivity after directly deleting  $e$ ; if not, it will **BE** directly deleted. If so, do nothing with it. Let's do the same thing for the rest of the sides;
- (3) After deleting the edges in **BoundaryEdge**, find out which edges in **InteriorEdge** become **BoundaryEdge** edges and modify the **BoundaryEdge** and **InteriorEdge** sets.

### 3. Literature References

Doing experiment on 3D model segmentation algorithm based on skeleton driver for models of different shapes. The platform is Windows and the basic configuration is Intel Core i3 3.3GHZ CPU, 4GB memory. The final segmentation boundary of the model is extracted after determining the key segmentation points, constructing the segmentation restricted region, extracting the concave region and refining the concave region.

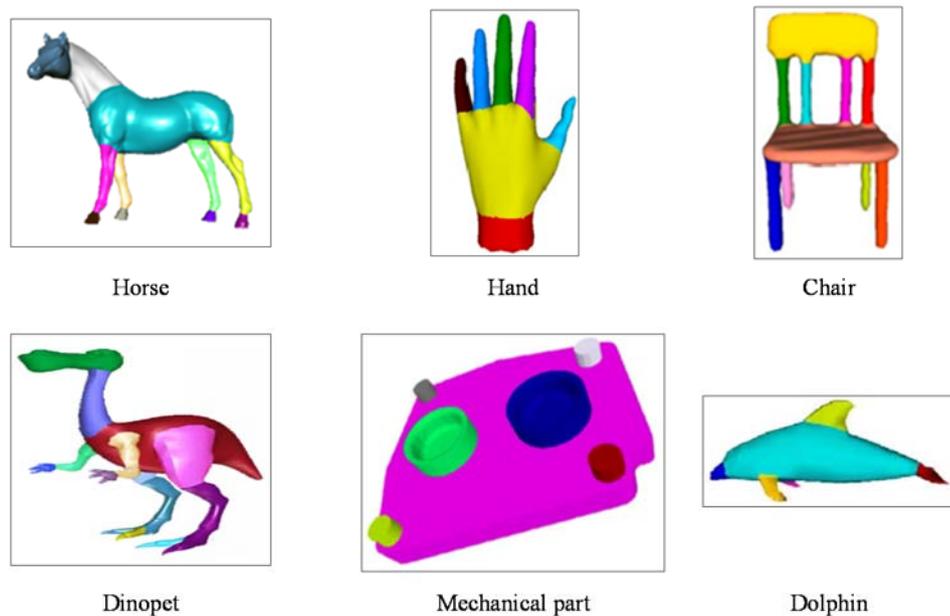


(a) Skeleton and critical segmentation point (red) (b) Restricting zone of segmentation (pink) (c) Concave region (green) (d) Segmentation line (red) (e) Segmentation result

**Fig 9.** Dinosaur model segmentation

Dinosaur models, for example, application of Dinosaur model based on the method of pseudo vector field of skeleton extraction algorithm to obtain its skeleton, and then through the space scanning method and geometric function to determine the key point of division, again from the point of division on both sides of the same distance to construct consists of two parallel plane segmentation restricted areas, then through discrete curvature method to calculate concave area. If there is a concave region in the segmentation restricted region, the final segmentation boundary is extracted by concave region refinement method, which is used for the final segmentation of the model. If there is no concave region in the restricted region, delete all key segmentation points in the restricted region and do not do any operation thereafter. Fig. 9 shows Dinosaur model segmentation, which produces good segmentation results.

Different models were selected for segmentation experiments to verify the feasibility of the algorithm. Fig. 10 shows the segmentation results of different models. It can be seen from the figure that the extracted segmentation boundary can well distinguish the parts with different shapes from each other, and the small but important parts are also significantly divided. In addition, the segmentation boundaries are located in the concave region of the model, which satisfies the minimization principle, and the final segmentation results conform to human visual perception.



**Fig 10.** Different model segmentation results

The relevant data of the experimental model are shown in table 1

**Table 1.** The relevant data of part of the experiment model

Model	Number of Fixed-point	Number of facts	Time of Concave region processing/s	Refinement time of concave region/s
Dinopet	4500	8996	5.135	0.537
Chair	13463	26926	15.989	1.728
Hand	4696	9388	5.383	0.566
Dolphin	418	836	0.435	0.052
Mechanical part	2660	5318	3.038	0.318
Horse	6428	12850	7.336	0.767

The proposed algorithm belongs to the non-learning segmentation line extraction algorithm [17], and the discrete curvature of vertex of 3D model needs to be calculated. The computational complexity of algorithm time is  $O(n^2)$ , that  $n$  is the total number of model vertices. In the process of performing concave region refinement, the time calculation complexity is  $O(C^2)$ , where  $C$  is the total number of inner edges of concave region. Since  $C \ll n$ , the time computational complexity of the proposed segmentation algorithm is regarded as  $O(n^2)$ . According to the relevant data of the experimental model shown in table 1, the computational speed of the segmentation algorithm in this paper is acceptable.

## 4. Conclusions

In this paper, a skeleton-driven 3D model segmentation algorithm is proposed. Firstly, the priority of the access order of skeleton branches is calculated to ensure that the small but important features can be first accessed; Next, the geometric and topological properties of the model are searched for the critical segmentation points to segment the two sub-models with obviously different shapes in the model; Then, based on the minimization principle, the concave region is calculated in the segmentation restricted region constructed at the key segmentation points; Then the concave region refinement method is proposed, and the boundary edge in the concave region is iteratively deleted to obtain a single partition boundary for the final model segmentation; At last, the segmentation experiments of different 3D models based on skeleton2 are carried out. The results show that the extracted segmentation boundary can be used to segment the model meaningfully, and the boundary appears at the minimum negative curvature.

As the part segmentation makes it possible to fabricate the parts with complex geometry or large scale size, the proposed algorithm can be applied to part segmentation. Several enhancements can be added to our algorithm:

The final segmentation boundary can be quickly and effectively extracted by mesh simplification. The efficiency of overall segmentation can be improved further.

The applicability of the proposed algorithm can be extended by taking the process planning for each sub-models, including additive manufacturing, subtractive machining and inspection operation, into consideration.

We believe that this skeleton-driven segmentation algorithm for 3D model will expand the application and capacity of part segmentation fabrication.

## Acknowledgments

This research was partially supported by National Natural Science Foundation of China (51305443, 51475459), Natural Science Foundation of Jiangsu Province (bk20130184, bk20160258), State Scholarship Fund of the China Scholarship Council (201706425017), Science and Technology Planning Project of Xuzhou City (KC18073), National College Student Innovation and Entrepreneurship Training Program (201710290015), and A Project Funded by Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD). The authors would also like to thank Mr. Jason Benoit for his proofreading of this paper.

## References

- [1] X.P. Sun, H. Li. The segmentation and application of 3d mesh model are reviewed. Journal of Computer Aided Design and Graphics, Vol. 8 (2005) No. 17, p.1648-1655.

- 
- [2] X. Li, T.W. Woon, T.S. Tan, Z. Huang. Decomposing polygon meshes for interactive applications. Symposium on Interactive 3d Graphics, Vol. 4 (2001) No. 6, p.35-42.
- [3] J.P. Serra. Image Analysis and Mathematical Morphology (Academic Press, America 1982).
- [4] A. Mangan, R. Whitaker. Partitioning 3D surface meshes using watershed segmentation. IEEE Transactions on Visualization and Computer Graphics, Vol. 4 (1999) No. 5, p.308-321.
- [5] D.L. Page, A.F. Koschan, M.A. Abidi. Perception-based 3D triangles mesh segmentation using fast marching Watersheds. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Madison, (2003), p.27-32.
- [6] M. Garland, A. Willmott, P.S. Heckbert. Hierarchical face clustering on polygonal surfaces. Symposium on Interactive 3d Graphics, (2001), p.49-58.
- [7] S. Katz, A. Tal. Hierarchical mesh decomposition using fuzzy clustering and cuts. ACM Transaction on Graphics, (2003) No. 22, p.954-961.
- [8] G.D. Chen, J.W. Li, L. Pan, et al. Skeleton extraction algorithm based on 3d human body model. Computer science, Vol. 7 (2009) No. 36, p.295-297.
- [9] H. Hoppe. Progressive meshes. Conference on Computer Graphics & Interactive Techniques, (1996), p.99-108.
- [10] F. Lazarus, A. Verroust. Level set diagrams of polyhedral objects. Fifth Symposium on Solid Modeling and Applications, (1999), p.130-140.
- [11] A. Verroust, F. Lazarus. Extracting skeletal curves from 3D scattered data. The Visual Computer, Vol.1 (2000) No. 16, p.15-25.
- [12] Y. Xiao, P. Siebert, N. Werghi. A discrete Reeb graph approach for the segmentation of human body scans. The 4th International Conference on 3-D Digital Imaging and Modeling, (2003), p.378-385.
- [13] Y. Shi, R. Lai, S. Krishna, N. Sicotte, et al. Anisotropic Laplace-Beltrami Eigenmaps: Bridging Reeb Graphs and Skeletons. Proceedings, (2008), p1-7.
- [14] L. Serino, C. Arcelli, G.S.D Baja. From skeleton branches to object parts. Computer Vision and Image Understanding, (2014) No. 129, p.42-51.
- [15] N. Pantuwong, M. Sugimoto. Skeleton growing: an algorithm to extract a curve skeleton from a pseudonormal vector field. The Visual Computer, Vol. 3 (2013) No. 29, p.203-216.
- [16] D.D. Hoffman, W.A. Richards. Parts of recognition. Cognition, (1984) No. 18, p.65-96.
- [17] Benhabiles, H., Lavoué, G., Vandeborre, J.P., Daoudi, M. Learning Boundary Edges for 3D-Mesh Segmentation. Computer Graphics Forum, Vol. 8 (2011) No.30, p.2170-2182.