www.cadanda.com
Computer-Aided Design™
AND APPLICATIONS

Surface Segmentation based on Concave Region and Flattenability

Rui Li[1] and Qingjin Peng[2]

[1]University of Manitoba, lir34512@myumanitoba.ca
[2]University of Manitoba, Qingjin.Peng@umanitoba.ca

Corresponding author: Qingjin Peng, Qingjin.Peng@umanitoba.ca

Abstract. Segmentation in a surface unfolding process can increase the unfolding accuracy and efficiency. Existing methods of the surface segmentation may not be effective for some surfaces as concave regions are ignored. This paper proposes a surface segmentation method based on the concave region and flattenability. Vertices in concave regions with small flattenability values are extracted as feature vertices that are then used to form segmentation regions. A curve fitting method is developed to form cutting lines in dividing the surface into parts by a least-squares fitting method. As cutting lines may destroy topology of an original mesh surface, a topology reconstruction method is proposed to build new segments. A 3D surface can finally be divided into segments with less unfolding deformation. The proposed method is applied in case studies to evaluate its performance. Results show that the proposed method improves the flattenability and unfolding accuracy of segmented surfaces.

Keywords: Surface segmentation; Concave region; Flattenability; Mesh surface;
DOI: https://doi.org/10.14733/cadaps.2022.561-574

1 INTRODUCTION

Surface segmentation is widely applied in reverse engineering, digital modeling, 3D model construction and identification [8]. In reverse engineering, product modeling starts at forming a 3D geometry model from point cloud data. The model is then segmented into some known surfaces such as plane, spherical and cylindrical [17]. In addition, the model segmentation is also applied in terrain and architecture fields to distinguish objective and environment [2] and divide structure into archaeological elements [5]. In the model reconstruction, 3D models can be segmented to build new models for improving the design process and 3D animation.

Unfolding is an important process in 3D product shaping, such as garment products, where a 3D-2D-3D process is used to unfold a 3D surface into 2D patterns for cutting material into patches before reshaping patches into the 3D shape [4, 25]. For example, Cai et al proposed a simplified surface unfolding algorithm to generate 2D planes with the minimum deformation by solving a constrained nonlinear programming problem and applied the algorithm to the blank design of sheet metal parts [3]. Liu et al developed an unfolding system for CAD models of the sheet metal using a free-form surface based on the mass-spring model [16]. Accuracy is essential to unfold a 3D surface into 2D planes that will be folded back into the 3D shape. Based on the computational geometry, only developable surfaces such as cylindrical and cone can be unfolded into a 2D plane without deformation [22]. As most 3D free-form surfaces are non-developable, it is a challenge to unfold non-developable 3D surfaces into 2D planes with less deformation.

Segmentation can add some cutting lines to divide a non-developable 3D surface into several parts to increase developability for reducing unfolding deformation. Thus, the problem of non-developable 3D surface unfolding becomes the surface segmentation to select a suitable position of cutting lines.

To meet the objective, this paper combines concave regions and flattenability to propose a surface segmentation method based on surface features including the vertex-level concave region, triangle-level concave region, curvature-level concave region and flattenability. Flattenability defined in [22] is used to measure a surface which is able to be unfolded into 2D patterns based on the summed inner angle of each vertex. Cutting lines are then fitted based on feature vertices by curve fitting. Contributions of this paper are as follows.

1) A surface segmentation method is proposed to improve surface flattenability. The method is easy applied with acceptable accuracy.

2) Feature vertices are defined by combining concave regions and flattenability to segment a surface into parts with improved developability.

3) Cutting lines are fitted based on feature vertices for the surface segmentation.

Following sections are organized as follows. Section 2 reviews methods of the surface segmentation and feature definition. Topological structure of the mesh surface and features of the surface segmentation are discussed in Sections 3 and 4, respectively. Section 5 introduces the proposed surface segmentation including the cutting line fitting and new topological data structure reconstruction, followed by case studies and conclusions in Sections 6 and 7.

2 LITERATURE REVIEW

2.1 Surface Segmentation

Based on the purpose, the surface segmentation can be classified into surface-type segmentation to separate a surface into different types of surfaces such as plane, sphere, and cylinder, and part-type segmentation to divide a surface into parts with special meanings [21]. The surface-type segmentation considers geometric features of a surface such as geodesic distances and curvatures, which is normally applied in representing distinct surfaces of 3D models for reverse engineering [26]. The part-type segmentation simulates the human cognition to segment a 3D model into meaningful parts that can be identified by human, which is normally applied in the model recognition and reconstruction.

Based on the process, the surface segmentation can be classified into human-computer interactive segmentation and automatic segmentation. The user involvement in the interactive segmentation includes generating segmentation regions, assigning the number of clusters, and labeling training data set for machine learning. For example, Zheng et al proposed an interactive segmentation interface to select potential regions by using a part-brush and a patch-brush [28]. They also proposed a simple interactive interface called Dot Scissor to select feature points for potential segmentation boundary. The segmentation boundary can be optimized based on a concavity-aware harmonic field [29]. Pal segmented 3D cloud point models manually into small regions for fitting large B-spline topological surfaces [19]. Wu et al drew a few freehand sketches on the model to determine segmentation regions [24]. Clustering methods such as K-means and spectral clustering require assigning the number of clustering [15]. Supervised machine learning in the model segmentation requires labeling training data. The automatic segmentation can separate a 3D model automatically without the user involvement. However, only a few methods can fully complete the automatic segmentation in the surface segmentation. Most of existing methods require users defining some parameters.

2.2 Features for Surface Segmentation

Features such as curvature, convex and concave regions can be extracted for guiding the surface segmentation to meet different requirements.

Curvature of a surface is the most common feature for the surface segmentation to measure the surface deviated from a plane [11]. For example, Lee et al proposed a 3D triangular mesh surface segmentation method based on discrete curvatures for an accurate segmentation and then presented a vertices clustering method using a shift algorithm [13]. Jagannathan and Miller proposed a model segmentation algorithm by searching curvedness values of vertices for each segment containing a set of vertices with similar curvedness values, and vertices in disjoint segments with different curvedness values [6]. Mizoguchi et al proposed a mesh segmentation method by estimating the mesh curvature with the sharp edge recognition and extracting regions using a region growing method [18].

A semantic part in 3D models may be composed of several weakly convex sub-parts [9]. Thus, convex regions can be used to indicate semantic parts and find feature points. For example, Kaick et al proposed a shape segmentation method by first decomposing the shape into approximate convex components, and then merging them into consistent parts based on a nonlocal geometric signature [9]. Kreavoy et al proposed a segmentation method to divide input models into meaningful, interchangeable parts based on metrics of the part convexity and compactness [12].

Concave regions normally connect two semantic parts of a 3D model. Three popular concave regions are vertices-level concave region, face-level concave region and curvature-level concave region. Au et al proposed an automatic mesh segmentation method based on shape concavity information and score-based greedy algorithm. The concavity-aware fields were selected based on vertices-level concave regions [1]. Wang et al proposed an automatic mesh segmentation method based on hierarchical spectral eigenvectors in analyzing vertices-level concaves [23]. Jiao et al proposed a concave region-based segmentation method which combines three concave regions including vertices-level concave region, face-level concave region and curvature-level concave region [7].

Feature points can guide the segmentation. They are normally located on tips of components of a model or segmented boundaries. If a feature point is on the model tip, it can be extracted by analyzing geodesic distances and convex regions. For example, Katz et al proposed a hierarchical mesh segmentation algorithm based on the prominent feature point and core extraction [10]. Feature points on segmented boundaries are normally extracted manually. Zheng et al proposed an interactive tool to define feature points for the mesh segmentation [29].

Flattenability measures a surface which is able to be unfolded into a plane [22]. Li et al proposed a surface segmentation method by combining the geometric distance and flattenability using a spectral clustering method to generate surfaces with the good flattenability [15].

The objective of this paper is to segment a surface into parts with less unfolding deformation. A key step is to search for the best position of the cutting line. In literature, concave regions are mainly used to form boundaries. Flattenability has been applied in the model segmentation [15]. However, the influence of concave regions was not considered. This paper combines concave regions and flattenability to search suitable positions of cutting lines for the model segmentation.

3 TOPOLOGICAL DATA STRUCTURE OF MESH SURFACE

Topological data structure plays an important role in the surface analysis. It should be built before the feature extraction [27]. The topological data structure of a mesh surface M is defined as a set

$\{V,E,F\}$ including vertices, edges and triangles. $V=\{v_i \mid v_i \in R^3, 1\le i\le m\}$ is a vertex set with *m* vertices. $E=\{e_{ij}=(v_i,v_j) \mid v_i,v_j \in V, i\ne j\}$ is an edge set, $F=\{f_{ijk}=(v_i,v_j,v_k) \mid v_i,v_j,v_k \in V, i\ne j, j\ne k, i\ne k\}$ is a triangle set.

One ring neighborhood vertices of vertex v_i are the vertices with common edges with v_i, and the set of one-ring neighborhood vertices of vertex v_i is defined as N^i. The one-ring neighborhood triangles of vertex v_i are the triangles composed by v_i and their one-ring neighborhood vertices. A set of one ring neighborhood triangles of vertex v_i is defined as F^i.

The normal vector of triangle f_i can be calculated as follows.

$$N(f_i)=\frac{e_{ij-1}\times e_{ij}}{\left\|e_{ij-1}\times e_{ij}\right\|}=\frac{(v_i-v_{j-1})\times(v_i-v_j)}{\left\|(v_i-v_{j-1})\times(v_i-v_j)\right\|} \tag{1}$$

where, e_{ij-1} is the edge vector from v_{j-1} to v_i, e_{ij} is the edge vector from v_j to v_i.

Normal vector $N(v_i)$ can be decided as follows [20].

$$N(v_i)=\frac{\sum_{f_k\in F^i}|f_k|N(f_k)}{\left\|\sum_{f_k\in F^i}|f_k|N(f_k)\right\|} \tag{2}$$

where, F^i is the set of one–ring neighborhood triangles of v_i. $|f_k|$ is the area of triangle f_k.

Based on mesh surface *M*, one-ring neighborhood vertices of each vertex, one-ring neighborhood triangles of each vertex, boundary edges of mesh surface, and triangles with the common edge can be searched. In addition, the length of each edge, area of each triangle, centroid of each triangle, three inner angles of each triangle, normal vector of each triangle, and normal vector of each vertex are calculated.

Based on the built topological data structure, the proposed method is described as shown in Figure 1. The process input is a triangular mesh surface model with the topological data structure. Features for concave regions and flattenability are selected to search segmented regions for cutting lines. Vertices in concave regions include the vertex-level concave region, triangle-level concave region and curvature-level concave region. They can be extracted as feature vertices as well as vertices with small flattenability values (FLV). Feature vertices that satisfy all the features are used to form segmentation regions. All the vertices are projected on a plane to design cutting lines for dividing a surface into parts using a curve fitting method. As cutting lines may destroy the topology of the original mesh surface, a topology reconstruction method is applied to build new segments. Then, the 3D surface is segmented into parts to be unfolded into 2D patterns with less deformation.

4 FEATURES OF SURFACE SEGMENTATION

4.1 Concave Regions

Three concave regions are vertices-level concave region, triangle-level concave region and curvature-level concave region. To search the vertices-level concave region, vertex v_i is considered as a concave vertex if at least one adjacent vertex $v_j \in N^i$ satisfies Equation (4) [1].

$$P(v_i)=\left\langle \frac{v_i-v_j}{\|v_i-v_j\|}, N(v_j)-N(v_i)\right\rangle \tag{3}$$

$$p(v_i)>\varepsilon_1 \tag{4}$$

where, $N(v_i)$ and $N(v_j)$ are unit normal vectors of v_i and v_j, respectively.

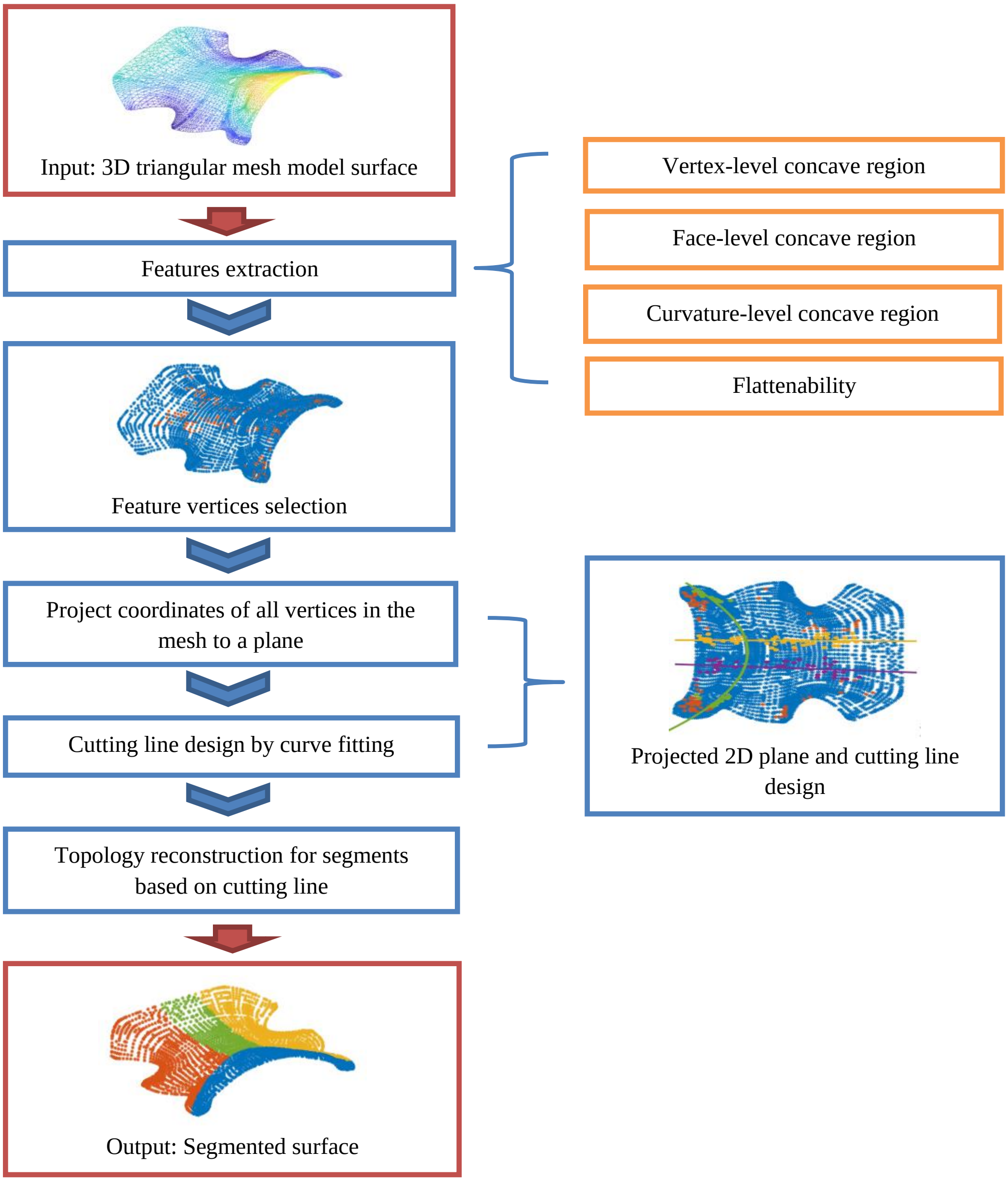

Input: 3D triangular mesh model surface
Features extraction
Vertex-level concave region
Face-level concave region
Curvature-level concave region
Flattenability
Feature vertices selection
Project coordinates of all vertices in the mesh to a plane
Cutting line design by curve fitting
Projected 2D plane and cutting line design
Topology reconstruction for segments based on cutting line
Output: Segmented surface

Figure 1: Flowchart of the proposed surface segmentation method.

The triangle-level concave region is searched as follows. For triangle f_i and its adjacent triangle f_j, vertices on common edges of f_i and f_j are concave vertices if the relation of f_i and f_j satisfies Equation (6).

$$T(f_i f_j) = \frac{\cos\alpha_i + \cos\alpha_j}{2} \tag{5}$$

$$T(f_i f_j) > \varepsilon_2 \tag{6}$$

where, α_i is an angle between the normal vector of triangle f_i and line by connected centroids of f_i and f_j. α_j is the angle between the normal vector of triangle f_j and their centroids as shown in Figure 2. $\cos\alpha_i$ is calculated as follows.

$$\cos\alpha_i = \frac{\langle N(f_i),(c_i - c_j)\rangle}{\| N(f_i)\| \cdot \| c_i - c_j \|} \tag{7}$$

where c_i and c_j are centroids of f_i and f_j respectively. $N(f_i)$ is normal vectors of f_i [7].

For searching the curvature-level concave region, vertex v_i is considered as a concave vertex if the mean curvature in vertex v_i satisfies Equation (9).

$$H(v_i) = \frac{1}{4}\sum_{j=1}^{n} \| e_{ij} \| \beta_j \tag{8}$$

$$H(v_i) > \varepsilon_3 \tag{9}$$

where $\| e_{ij} \|$ is the length of edge e_{ij}, β_j is a dihedral angle for triangles f_i and f_j calculated by Equation (10) as shown in Figure 2.

$$\cos\beta_j = \frac{\langle N(f_i), N(f_j)\rangle}{\| N(f_i)\| \cdot \| N(f_j)\|} \tag{10}$$

4.2 Flattenability

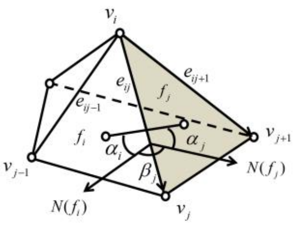
Flattenability ϖ measures a surface which is able to be unfolded into 2D patterns based on the summed inner angle in each vertex [22]. $\varpi(v_i)$ at v_i is defined in Equation (11). A lower value of $\varpi(v_i)$ means that when unfolding one-ring neighborhood triangles of v_i into a 2D plane, the unfolded 2D plane has less deformation and distortion [22].

$$\varpi(v_i) = \left|\theta(v_i) - 2\pi\right| \tag{11}$$

where $\theta(v_i) = \sum_j \theta_j$ is the summed inner angle in vertex v_i as shown in Figure 3. If $\theta(v_i) = 2\pi$, one-ring neighborhood triangles of v_i can be unfolded into a 2D plane without distortion and deformation. If $\theta(v_i) - 2\pi > 0$, unfolding one-ring neighborhood triangles of v_i will cause overlaps. If $\theta(v_i) - 2\pi < 0$, unfolding one-ring neighborhood of triangles of v_i will cause gaps.

Thus, if $\varpi(v_i)$ satisfies Equation (12), v_i is a non-flattenable vertex.

$$\varpi(v_i) > \varepsilon_4 \tag{12}$$



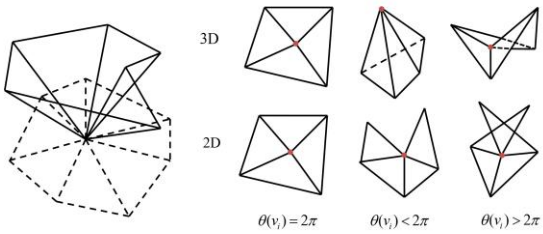


Figure 2: Schematic diagram mesh surface. **Figure 3:** $\theta(v_i)$ with different inner angles [22].

$\sum \varpi$ is the summarized value of ϖ of a model as follows:

$$\sum \varpi = \sum_{i=1}^{m} \varpi(v_i) \tag{13}$$

Flattenablility value (FLV) is applied to measure the flattenablility of the surface as follows [14].

$$\mathrm{FLV} = 10 \times 0.5^{\sum_{i=1}^{m} \varpi(v_i)} \tag{14}$$

Feature vertices are vertices that satisfy Equations (4, 6, 9, and 12) simultaneously. ε1 to ε4 in Equations (4, 6, 9, and 12) are thresholds defined based on the model shape to generate enough feature vertices. After preparing the data structure of a mesh surface, three concave regions and flattenability are searched for feature vertices. The process of the feature vertices extraction is as follows. If $p(v_i) > \varepsilon_1$, $w_1(v_i) = 1$. If $T(f_i f_j) > \varepsilon_2$, $w_2(v_i) = 1$ and $w_2(v_j) = 1$. if $H(v_i) > \varepsilon_3$, $w_3(v_i) = 1$. If $\varpi(v_i) > \varepsilon_4$, $w_4(v_i) = 1$. For v_i , if $w_1(v_i) = 1 \cap w_2(v_i) = 1 \cap w_3(v_i) = 1 \cap w_4(v_i) = 1$, v_i is selected as feature vertex.

5 SURFACE SEGMENTATION

5.1 Cutting Lines Forming

Cutting lines closed by feature vertices can segment the model with a large FLV. As the polyline connected by feature vertices is not smooth and coherent, it cannot be used as a cutting line directly. Thus, a cutting line is formed based on fitting feature vertices into a curve to segment the surface into parts. However, it is difficult to fit a smooth cutting line on the input surface by feature vertices in the 3D space directly. Thus, a 3D surface *M* is first projected onto a 2D plane *P*, and feature vertices are fitted into a curve on the 2D plane *P*. A surface *S* is designed based on the fitted curve to be vertical with the 2D plane. The intersection line between surface *S* and input 3D surface *M* forms the cutting line for the surface segmentation.

The 3D surface is first projected into a 2D plane. Plane *P* is represented as $p_1 x + p_2 y + p_3 z + 1 = 0$ with the normal vector $d_P = (p_1, p_2, p_3)$. The plane equation can be rewritten as follows.

$$z = -\frac{p_1}{p_3} x - \frac{p_2}{p_3} y - \frac{1}{p_3} \tag{15}$$

To calculate p_1, p_2 and p_3, a least-squares fitting method is applied to search the optimal solution by minimizing the objective function as follows.

$$\min \sum_{i=0}^{m-1} \left(-\frac{p_1}{p_3} x - \frac{p_2}{p_3} y - \frac{1}{p_3} - z\right)^2 \tag{16}$$

A new coordinate is rebuilt based on plane *P*, where plane *P* is an *x-o-y* coordinate plane in the new coordinate. The original coordinate of mesh surface is transferred to the new coordinate as follows. Point $(0,0,-\frac{1}{p_3})$ in plane *P* is the origin of the new coordinate. The original *y*-axis rotates γ degrees clockwise around *z*-axis to generate new *y*′-axis and *x*′-axis. Then, the original *z*-axis rotates α degrees clockwise around *x*′-axis to generate a new *z*′-axis, as shown in Figure 4.

$$\begin{bmatrix} v^{x'} \\ v^{y'} \\ v^{z'} \end{bmatrix} = R(\alpha)R(\gamma)\begin{bmatrix} v^{x} \\ v^{y} \\ v^{z} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix}\begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} v^{x} \\ v^{y} \\ v^{z} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ -1/p_3 \end{bmatrix} \tag{17}$$

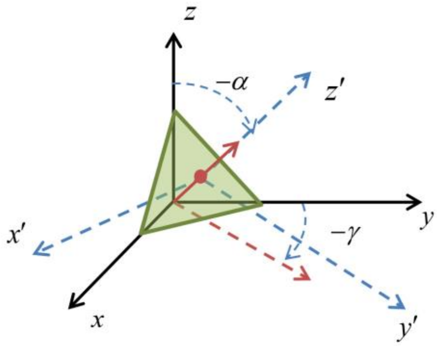


Figure 4: Candidate transformation.

$$\alpha = -\arccos\left(\frac{p_3}{\sqrt{p_1^2 + p_2^2 + p_3^2}}\right) \tag{18}$$

$$\gamma = -\arccos\left(\frac{p_2}{\sqrt{p_1^2 + p_2^2}}\right) \tag{19}$$

where, the rotation direction is counter-clockwise in Equation (17). If the rotation direction is clockwise, α and γ should be changed to $-\alpha$ and $-\gamma$, respectively.

Feature vertices are projected onto plane *P*, because plane *P* is the *x*-*o*-*y* coordinate plane, thus, only *z* coordinate is change to 0.

A curve is fitted based on the projected feature vertices in plane *P* by the least square method. To make sure part boundaries are easy to be connected for a whole surface, a polynomial function is applied as follows.

$$v^y = f(v^x) = a_1 v_x^n + a_2 v_x^{n-1} + \cdots + a_n v_x + a_{n+1} \tag{20}$$

The least-squares fitting method minimizes the objective function as follows.

$$\min \sum_{i=1}^{N} [f(v^x; a_1, a_2, \cdots a_{n+1}) - v^y]^2 \tag{21}$$

where, *N* is the number of feature vertices.

A surface *S* perpendicular with plane *P* and intersected to plane *P* with the fitted curve is built to calculate the cutting line between two segments. The equation of surface *S* is as follows.

$$S = \begin{cases} x = a_1 t^n + a_2 t^{n-1} + \cdots + a_n t + a_{n+1} \\ y = t \\ z = s \end{cases} \tag{22}$$

Surface *S* is intersected with edges in 3D mesh surface *M*. Intersections are calculated as follows.

$$\begin{cases} a_1 t^n + a_2 t^{n-1} + \cdots + a_n t + a_{n+1} - ((e_{ij}(v_j^x) - e_{ij}(v_i^x)) \cdot k + e_{ij}(v_i^x))=0 \\ t - ((e_{ij}(v_j^y) - e_{ij}(v_i^y)) \cdot k + e_{ij}(v_i^y))=0 \\ s - ((e_{ij}(v_j^z) - e_{ij}(v_i^z)) \cdot k + e_{ij}(v_i^z))=0 \end{cases} \tag{23}$$

Intersections of surface *S* and boundary of mesh surface *M* are first searched. The boundary is then searched for other edges of the triangle having intersections of surface *S*. Other triangles with the common edge are searched for the next intersection. The process continues until finding another intersection of surface *S* and boundary of the 3D mesh. The connection order of intersections forms the cutting line to segment mesh surface *M* into parts.

When surface *S* intersects with mesh surface *M*, there are normally two situations. The first one is shown in Figure 5(a), in triangle *A*, two edges without a common edge have one intersection with surface *S*, thus, the cutting line is formed by the connection of intersections of common and uncommon edges. For the second situation in Figure 5(b), there are three intersections between surface *S* and two edges that are uncommon edges. The simplified cutting line is the connection by intersections of the common edge and the edge with only one intersection. The cutting line generation process is presented in Algorithm 1.

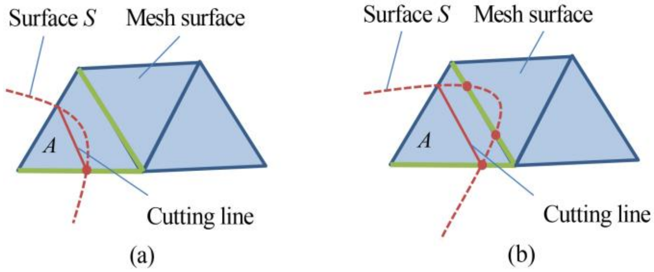
5.2 New Topological Data Structure Reconstruction

The model segmentation will form new vertices, edges and triangles. The topological data structure of the model should be reconstructed. A triangle in the mesh model can be segmented into a triangle and a quadrilateral [15]. For example, intersections between mesh surface *M* and cutting line is new vertices in two adjacent segments M_1 and M_2 in the new topological data structure, as shown in Figure 6, vertices *D* and *E*. Edge *DE* is a new edge of two adjacent segments M_1 and M_2. Triangle *CDE* is a new triangle of segment M_1. Quadrilateral *ABED* is segmented into two triangles *ABE* and *ADE* by adding a new edge AE. Thus, for segment M_1, two new vertices *D* and *E*, three new edges *CD*, *CE*, and *DE* and a new triangle CDE are added in the new topological data structure. For segment M_2, two new vertices *D* and *E*, four new edges *AD*, *BE*, *AE*, and *DE* and two new triangles *ABE* and *ADE* are added in the new topological data structure.

Algorithm 1: Cutting line forming

```
Input: Mesh surface $M$ after coordinate transformation, topological data structure of mesh surface $M$.
Output: Cutting line
for $i \leftarrow 1$ to number of boundary edges
    Search boundary intersections using Equation(23)
end for
Fix one boundary intersections and the boundary is labeled as $e_{FIX}$
Search a triangle with $e_{FIX}$ and labeled as $f_{LAB}$
Calculate intersections on two additional edges in $f_{LAB}$ using Equation (23)
Label $e_{FIX}$ as $e_{UFIX}$
Label the edge with intersection as $e_{FIX}$ and put the intersection in set of cutting line vertices
Label $f_{LAB}$ as $f_{ULAB}$
for $i \leftarrow 1$ to 1000
    Search triangles with $e_{FIX}$
    for $j \leftarrow 1$ to 2
        if The triangle does not have label $f_{ULAB}$
            The triangle is labeled as $f_{LAB}$
        end if
    end for
    if The $f_{LAB}$ not contain a boundary edge of surface $M$
        Calculate intersections on two additional edges in $f_{LAB}$ using Equation(23)
        if The number of intersections=1
            Label $e_{FIX}$ as $e_{UFIX}$
              Label the edge with intersection as $e_{FIX}$ and put the intersection in set of cutting
```

```
line vertices
22:             Label $f_{LAB}$ as $f_{ULAB}$
23:         else if The number of intersections=3
24:             Label $e_{FIX}$ as $e_{UFIX}$
25:             Label the edge with 1 intersection as $e_{FIX}$ and put the intersection in set of cutting
line vertices
26:             Label $f_{LAB}$ as $f_{ULAB}$
27:         end if
28:     else
29:         Break from loop line 10
30:     end if
31: end for
32: Connect vertices in the set of cutting line vertices as a cutting line
```



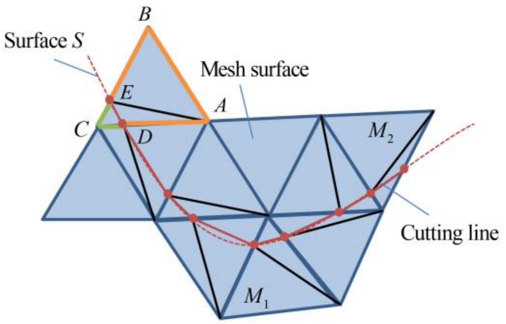


Figure 5: Two situations of surface S-mesh intersection. **Figure 6:** New topological data structure.

6 CASE STUDY

6.1 Human Nose Model

A human nose model and a car shell model are segmented in this study to reduce deformation when unfolding them into 2D planes. Both of them are non-developable surfaces. The nose model contains 3039 vertices, 17883 edges and 8999 triangles as shown in Figure 7(a). The value of $\sum\varpi$ for the model is 12.3425 and the value of FLV is 0.0019.

Red areas in Figure 7(b) are regions with the large value of ϖ. Red areas in Figures 7(c) (d) and (e) are the vertex-level concave region, triangle-level concave region, and curvature-level concave region, respectively. Red areas in Figure 7(f) are feature vertices selected from vertices in red areas in Figures 7(b) (c) (d) and (e). Curve fitting is applied to generate cutting lines for segmenting the nose model as shown in Figure 7(g). The segmented result of the model is shown in Figure 7(h) and Table 1. FLV is applied to measure flattenability of the model, where a larger value of FLV means the surface is more developable. Table 1 lists the number of vertices, number of edges and number of triangles for the whole surface and segmented results. Values of $\sum\varpi$ decrease and values of FLV increase after the segmentation. FLV of the whole model is 0.0019. The model is segmented into 4 pieces with FLV 0.1489, 3.1350, 5.0404 and 0.8181, respectively. The segmented results of the nose model are shown in Figure 7(i).

6.2 Car Shell Model

A car shell model is shown in Figure 8(a) with 3775 vertices, 10981 edges and 7200 triangles. The value of $\sum\varpi$ for the model is 46.6140 and the value of FLV is 9.3×10^{-14}.

Red areas in Figure 8(b) are regions with larger values of ϖ . Red areas in Figures 8(c) (d) and (e) are the vertex-level concave region, triangle-level concave region, and curvature-level concave region, respectively. Red areas in Figure 8(f) are feature vertices and cutting lines of the model shown in Figure 8(g). The segmented result of the model is shown in Figure 8(h) and Table 2. After the segmentation, values of $\sum\varpi$ decrease and values of FLV increase. FLV of the whole model is 9.3×10^{-14}. The model is segmented into 4 pieces with FLV 3.6×10^{-6}, 0.0071, 0.2127 and 0.0194, respectively. The segmented results of the car model are shown in Figure 8(i).

For both human nose and car shell models, values of FLV for each segment increase significantly compared to the model without segmentation, which shows that segmented surfaces can be unfolded into 2D patterns with less deformation and can also maintain accuracy when 2D patches are formed back to the 3D shape.

Comparing with an existing surface segmentation method in [15], this paper proposes a simple surface segmentation method to meet requirements of increasing flattenability for non-developable surfaces and the unfolding accuracy. In addition, the proposed method combines concave regions and flattenability to search for feature vertices and generate smooth cutting lines based on feature vertices by projecting 3D model into 2D surface and a curve fitting method. The method can generate cutting lines directly based on feature vertices rather than dividing the surface into several parts by a spectral clustering method used in [15]. Thus, our method is simple and easy in applications with less processing steps.

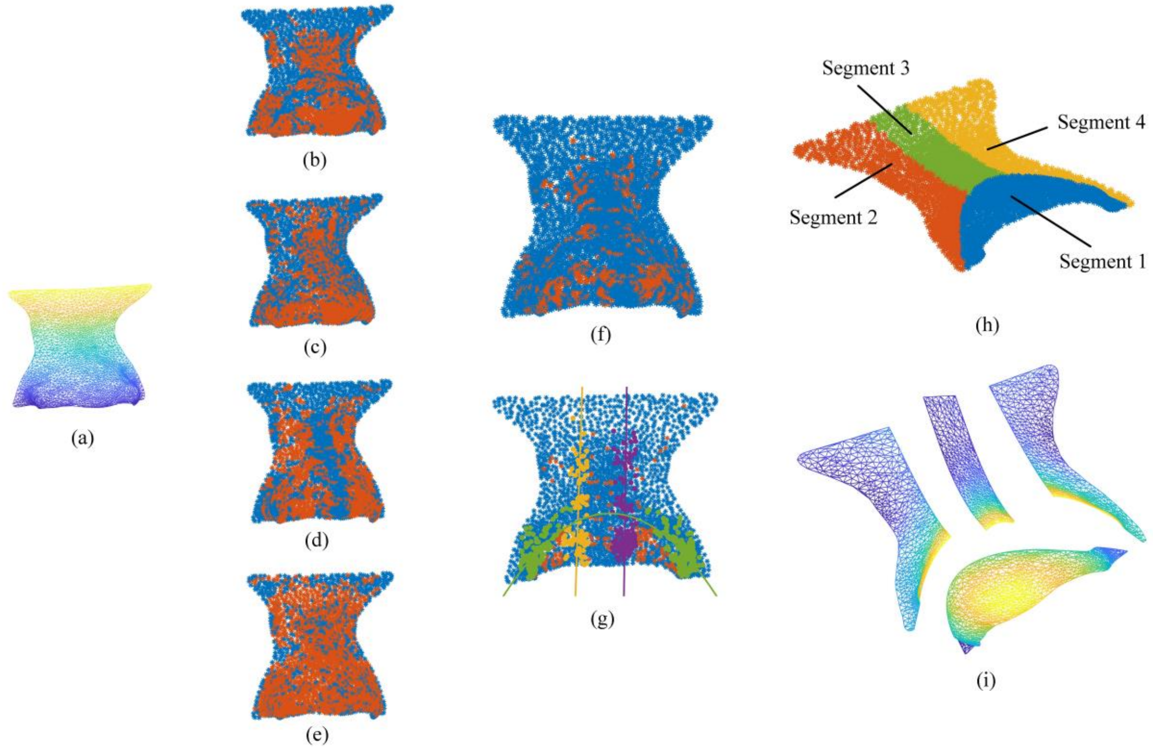


Figure 7: Segmentation of the human nose model.

Human nose model		*No. vertices*	*No. edges*	*No. triangles*	$\sum\varpi$	*FLV*
Whole model		3039	17883	8999	12.3425	0.0019
Segmented model	*Segment 1*	1248	3512	2265	6.0692	0.1489
	Segment 2	900	2487	1588	1.6734	3.1350
	Segment 3	695	1872	1178	0.9884	5.0404
	Segment 4	942	2611	1670	3.6115	0.8181

Table 1: Segmentation results of the human nose model.

Car shell model		*No. vertices*	*No. edges*	*No. triangles*	$\sum\varpi$	*FLV*
Whole model		3775	10981	7200	46.6140	9.3×10^{-14}
Segmented model	*Segment 1*	1506	4266	2753	21.4210	3.6×10^{-6}
	Segment 2	901	2505	1605	10.4642	0.0071
	Segment 3	1018	2797	1779	5.5552	0.2127
	Segment 4	908	2461	1556	9.0105	0.0194

Table 2: Segmentation results of the car shell model.

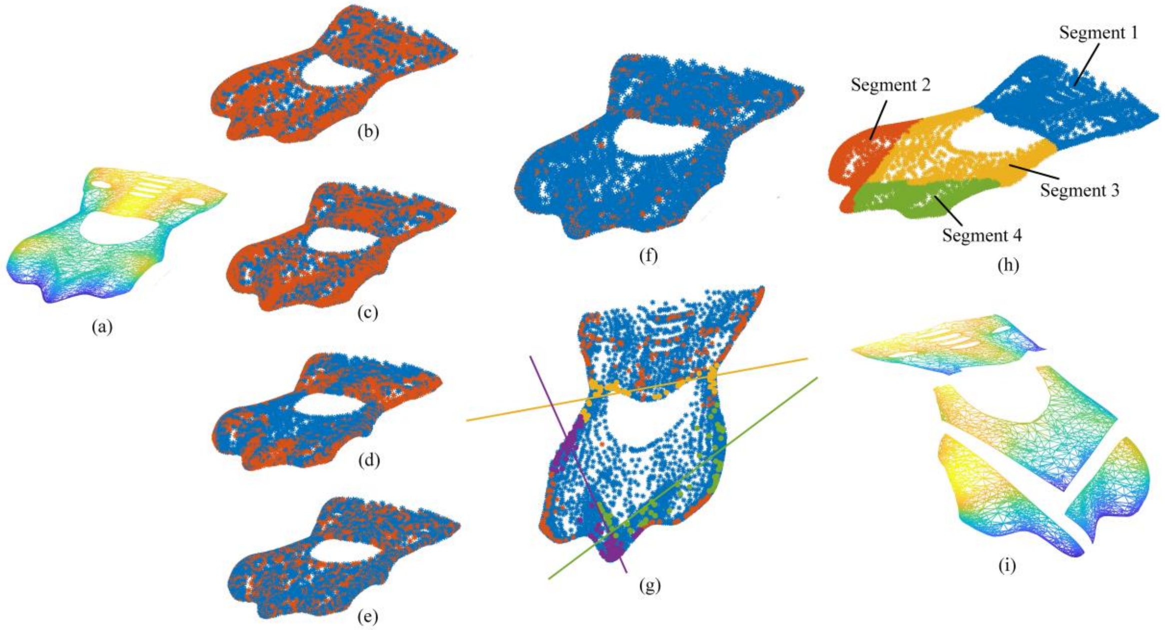


Figure 8: Segmentation of the car shell model.

7 CONCLUSIONS

This paper proposed a model segmentation method by combining concave regions and flattenability to segment a surface model from a 3D model to 2D patterns with less deformation. Features of vertex-level, triangle-level and curvature-level concave regions, and flattenability are applied in selecting segmenting regions. Vertices in these concave regions with the small FLV are selected as feature vertices. Curve fitting is proposed to find cutting lines for the final segmentation. Two case studies verified performance of the proposed method. FLV of surfaces is increased significantly after the segmentation. The proposed method will be tested in different applications to improve the method in the further work.

ACKNOWLEDGMENTS

The authors wish to acknowledge that this research has been supported by the Discovery Grants from the Natural Sciences and Engineering Research Council (NSERC) of Canada, and the Graduate Enhancement of Tri-Council Stipends (GETS) program from the University of Manitoba.

Rui Li, http://orcid.org/0000-0002-3277-2772
Qingjin Peng, http://orcid.org/0000-0002-9664-5326

REFERENCES

[1] Au, O.-K.-C.; Zheng, Y.; Chen, M.; Xu, P.; Tai, C. L.: Mesh segmentation with concavity-aware fields, IEEE Transactions on Visualization and Computer Graphics, 18(7), 2011, 1125-1134. https://doi.org/10.1109/TVCG.2011.131

[2] Brodu, N.; Lague, D.: 3D terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: Applications in geomorphology, ISPRS Journal of Photogrammetry and Remote Sensing, 68, 2012, 121-134. https://doi.org/10.1016/j.isprsjprs.2012.01.006

[3] Cai, Z.-Y.; Li, M.-Z.; Zhang, H.-M.: A simplified algorithm for planar development of 3D surface and its application in the blank design of sheet metal forming, Finite elements in analysis and design, 43(4), 2007, 301-310. https://doi.org/10.1016/j.finel.2006.10.005

[4] Decaudin, P.; Julius, D.; Wither, J.; Boissieux, L.; Sheffer, A.; Cani, M.-P.: Virtual garments: A fully geometric approach for clothing design, Computer Graphics Forum, Wiley Online Library, 2006, 625–634. https://doi.org/10.1111/j.1467-8659.2006.00982.x

[5] Fiorillo, F.; Jiménez Fernández-Palacios, B.; Remondino, F.; Barba, S.: 3D Surveying and modelling of the Archaeological Area of Paestum, Italy, Virtual Archaeology Review, 4(8), 2015, 55-60. https://doi.org/10.4995/var.2013.4306

[6] Jagannathan, A.; Miller, E.-L.: Three-dimensional surface mesh segmentation using curvedness-based region growing approach, IEEE Transactions on pattern analysis and machine intelligence, 29(12), 2007, 2195-2204. https://doi.org/10.1109/TPAMI.2007.1125

[7] Jiao, X.; Wu, T.; Qin, X.: Mesh segmentation by combining mesh saliency with spectral clustering, Journal of Computational and Applied Mathematics, 329, 2018, 134-146. https://doi.org/10.1016/j.cam.2017.05.007

[8] Joshi, K.; Bhatt, A.-D.: Development of Multi-branch T-spline Templates and its Applications in Reverse Engineering, Computer-Aided Design & Applications, 17(3), 2020, 487-501. https://doi.org/10.14733/cadaps.2020.487-501.

[9] Kaick, O.-V.; Fish, N.; Kleiman, Y.; Asafi, S.; Cohen-Or, D.: Shape segmentation by approximate convexity analysis, ACM Transactions on Graphics (TOG), 34(1), 2014, 1-11. https://doi.org/10.1145/2611811

[10] Katz, S.; Leifman, G.; Tal, A.: Mesh segmentation using feature point and core extraction, The Visual Computer, 21(8–10), 2005, 649–658. https://doi.org/10.1007/s00371-005-0344-9

[11] Koenderink, J.-J.; Van Doorn, A.-J.: Surface shape and curvature scales, Image and vision computing, 10(8), 1992, 557-564. https://doi.org/10.1016/0262-8856(92)90076-F

[12] Kreavoy, V.; Julius, D.; Sheffer, A.: Model composition from interchangeable components, In 15th Pacific Conference on Computer Graphics and Applications (PG'07), IEEE, 2007, October, 129-138. https://doi.org/10.1109/PG.2007.40

[13] Lee, J.; Kim, S.; Kim, S.-J.: Mesh segmentation based on curvatures using the GPU, Multimedia Tools and Applications, 74(10), 2015, 3401-3412. https://doi.org/10.1007/s11042-014-2104-1

[14] Li, R.; Peng, Q.; Ingleby, H.; Sasaki, D.: Improvement of flattenability using particle swarm optimizer for surface unfolding in bolus shaping, SN Applied Sciences, 2(9), 2020, 1-19. https://doi.org/10.1007/s42452-020-03330-9

[15] Li, R.; Peng, Q.; Ingleby, H.; Sasaki, D.: Improved bolus shaping accuracy using the surface segmentation and spectral clustering, International Journal of Modelling and Simulation, 2020, 1-12. https://doi.org/10.1080/02286203.2020.1756036

[16] Liu, X.; Li, S.; Zheng, X.; Lin, M.: Development of a flattening system for sheet metal with free-form surface, Advances in Mechanical Engineering, 8(2), 2016, 1687814016630517. https://doi.org/10.1177/1687814016630517

[17] Mejia, D.; Ruiz-Salguero, O.; Sánchez, J.-R.; Posada, J.; Moreno, A.; Cadavid, C.-A.: Hybrid geometry/topology-based mesh segmentation for reverse engineering, Computers & Graphics, 73, 2018, 47-58. https://doi.org/10.1016/j.cag.2018.03.004

[18] Mizoguchi, T.; Date, H.; Kanai, S.; Kishinami, T.: Segmentation of scanned mesh into analytic surfaces based on robust curvature estimation and region growing, In International conference on geometric modeling and processing Springer, Berlin, Heidelberg, 2006, July, 644-654. https://doi.org/10.1007/11802914_52

[19] Pal, P.: Fast freeform hybrid reconstruction with manual mesh segmentation, The International Journal of Advanced Manufacturing Technology, 63(9-12), 2012, 1205-1215. https://doi.org/10.1007/s00170-012-3986-6

[20] Taubin, G.: Estimating the tensor of curvature of a surface from a polyhedral approximation, In Proceedings of IEEE International Conference on Computer Vision, IEEE, 1995, June, 902-907. https://doi.org/10.1109/ICCV.1995.466840

[21] Theologou, P.; Pratikakis, I.; Theoharis, T.: A comprehensive overview of methodologies and performance evaluation frameworks in 3D mesh segmentation, Computer Vision and Image Understanding, 135, 2015, 49-82. https://doi.org/10.1016/j.cviu.2014.12.008

[22] Wang, C.-C.: Towards flattenable mesh surfaces, Computer-Aided Design, 40(1), 2008, 109-122. https://doi.org/10.1016/j.cad.2007.06.001

[23] Wang, H.; Lu, T.; Au, O.-K.-C.; Tai, C.-L.: Spectral 3D mesh segmentation with a novel single segmentation field. Graphical models, 76(5), 2014, 440-456. https://doi.org/10.1016/j.gmod.2014.04.009

[24] Wu, H. Y.; Pan, C.; Pan, J.; Yang, Q.; Ma, S.: A sketch-based interactive framework for real-time mesh segmentation, In Computer graphics international, 2007.

[25] Yasseen, Z.; Nasri, A.; Boukaram, W.; Volino, P.; Magnenat-Thalmann, N.: Sketch-based garment design with quad meshes, Computer-Aided Design, 45(2), 2013, 562–567. https://doi.org/10.1016/j.cad.2012.10.041

[26] Zhang, E.; Mischaikow, K.; Turk, G.: Feature-based surface parameterization and texture mapping, ACM Transactions on Graphics (TOG), 24(1), 2005, 1-27. https://doi.org/10.1145/1037957.1037958

[27] Zhang, Y.; Luo, X.; Jia, J.: A Compact Face-Based Topological Data Structure for Triangle Mesh Representation, Computer-Aided Design & Applications, 16(3), 2019, 539-557. https://doi.org/10.14733/cadaps.2019.539-557

[28] Zheng, Y.; Tai, C.-L.: Mesh decomposition with cross-boundary brushes, In Computer Graphics Forum Oxford, UK: Blackwell Publishing Ltd, 29(2), 2010, May, 527-535. https://doi.org/10.1111/j.1467-8659.2009.01622.x

[29] Zheng, Y.; Tai, C.-L.; Au, O.-K.-C.: Dot scissor: a single-click interface for mesh segmentation, IEEE transactions on visualization and computer graphics, 18(8), 2011, 1304-1312. https://doi.org/10.1109/TVCG.2011.140