A NEW APPROACH FOR TESSELLATING TRIMMED PARAMETRIC SURFACES

Liu Dezhi  Dong Jinxiang  and  Tong Ruofeng
State Key Laboratory of CAD&CG, Zhejiang University
PO Box 549,Zhejiang University, Hangzhou Post Code 310027 P.R.C

ABSTRACT
Trimmed parametric surfaces play important roles in CAD/CAM systems. In this paper, a new approach for tessellating trimmed parametric surfaces under Brep topologic structures is presented. The algorithm does this by three stages, and the key stage is splitting irregular trimmed parametric surfaces into monotone regions which will not include inner loops. The algorithm will work well compared with [1-4] because all operations are done under consistence of surfaces’ topologic structures.

KEYWORDS
Trimmed Parametric Surface, Tessellation Brep Solid Model, Algorithm

1. BACKGROUND
With the appearance of CAD/CG in the 1960s, there have been two distinct threads of 3D modeling technology after wireframe model. The first thread is surface modeling and the second thread is solid modeling, which is based on polygon model. During the 1980s, research addressed the problem of introducing sculptured surface facilities in boundary representation models, with the result that surface and solid modeling systems have by now been quite successfully merged. One of the bridges of surface models and solid models is trimmed surface, and trimmed surface plays a more and more important role in CAD/CAM systems.

A trimmed surface is a bounded region defined either a larger surface or an unbounded surface. In a solid modeling context, the faces of an object are trimmed surfaces, at least from the geometric point of view. Such regions may be quite complex with multiple non-intersecting boundaries. Many algorithms for non-trimmed surfaces, such as surface rendering, FEM, cutter trajectory calculation, etc, can not be used for trimmed surfaces directly. A method for solving the problem is to tessellate trimmed surfaces, and use triangular facets approximating trimmed surfaces.

In this paper, our work is concerned with the conversion of trimmed parametric surfaces into small triangular facets. Differently with [1, 4], our approach for tessellating trimmed surfaces is under Brep topologic structures, and all operations automatically keep topologic consistence.

2. BASE OF THE ALGORITHM

In our CAD system ZD-MCAD II, the product model is feature model which geometric kernel is Brep solid model based on sculptured surfaces[5-7]. In the structure, geometric information attaches to topologic structure by surfaces, curves and coordinates. Moreover, faces are discrete facets of a surface with mapping relationship many-to-one, and edges are discrete segments of a curve with mapping relationship many-to-one too. Advantages of the structure are well merging solid models with sculptured surfaces. Here, surfaces are trimmed NURBS surfaces except for primary surfaces, such as planes, conic surfaces, etc.

In order to tessellate irregular trimmed surfaces, we first divide trimmed surfaces into monotonic patches, then triangulate these patches. Now we introduce some definitions used in the article.

Definition 1 The point on boundary curves \( u(t) = 0, v(t) = 0 \) of a trimmed surface is called extreme point in \( U \) direction, if it satisfies \( \frac{du(t)}{dt} = 0 \) in domain. Definition of extreme point in \( V \) direction is similar.
**Definition 2** A trimmed surface is called monotonic surface in U direction, if it satisfies that a scan line intersects in U direction intersect with boundary curves of the surface at most two times in domain. Definition of monotonic surface in V direction is similar.

**Definition 3** A trimmed surface is called monotonic surface if it is monotonic both in U direction and in V direction.

### 3. DETAILS OF THE ALGORITHM

The main algorithm is shown in the following:

**STEP 1** do initial work, divide inner loop curves and outer loop curves of trimmed surface into segments in domain and space respectively according to accuracy.

**STEP 2** mark all discrete edges of trimmed surface according to inner loop edges and outer loop edges.

**STEP 3** build two bidirection links FaceList and ResultFaceList, put irregular trimmed surfaces at tail of FaceList.

**STEP 4** WHILE not empty of FaceList DO

- **4.1** get a surface TempFace from head of FaceList;

- **4.2** build a bidirection link LoopList for store loops of TempFace. Link node includes two fields: SegList and UVBox. The SegList is segment list for loops of TempFace, while UVBox is bounding box for loops of TempFace in parametric domain. Each loops of TempFace put into LoopList as a node, and outer loop is put at LoopList tail;

- **4.3** find extreme points of all loops in U direction and V direction;

- **4.4** IF there is not extreme point THEN TempFace is a monotonic surface, and put it into ResultFaceList;

  ELSE

  TempFace is not monotone, split it at extreme points, and put split faces into FaceList;

  ENDIF

ENDWHILE

**STEP 5** triangulate monotonic surfaces in ResultFaceList

In the main algorithm, dividing inner loop curves and outer loop curves is under topologic structures, i.e., we apply Euler operators, such as SplitEdge, MEV, and so forth, to divide curves in parametric domain and space respectively.

**Sub-algorithm 1** for splitting non-monotonic surfaces at extreme points

**STEP 1** find intersection between the scan line at the extreme point and segments in Seglist of LoopList by scan line algorithm, and put intersected segments into InterSegList

**STEP 2** sort InterSegList by direction of scan lines

**STEP 3** find a ‘applicable’ segment in sorted InterSegList, which can be linked with the segment at the extreme point, split face, eliminate the extreme point by Euler operation SplitFace

**Sub-algorithm 2** for finding intersection between the horizontal scan line (U direction) or the vertical scan line (V direction) at the extreme point and all segments of the face, then putting intersected segments into InterSegList

**WHILE** each segments in SegList of each loops in LoopList DO table 1

In table 1, dotted lines represent scan lines, and solid lines represent segments of trimmed boundary curves divided in parametric domain. Case 1 belongs to the case which scan lines do not intersect with segments. In case 5, 6 and 7, there is at least one segment between segment A and B. Case 8 belongs to ‘real’ intersection case.

### 4. EXAMPLES

We apply the algorithm in ZD-MCADII as a method to render solid models of parts. Two examples are shown in Fig.1 and Fig. 2.

### 5. CONCLUSION

The accuracy demand of tessellating rimmed parameter surfaces in the Brep solid model from engineering point of view is higher than from rendering point of view. According to consistent constrain of topologic structure in Brep solid model, we apply Euler operations, such as MEV, SplitEdge and SplitFace, to tessellate irregular trimmed surfaces. Therefore, our
method guarantees effectiveness of discrete. The time cost and space cost of our algorithm both are \( \Theta(n+m) \), where \( n \) is complexity of trimmed surfaces, i.e. the number of extreme points of trimmed surfaces, and \( m \) is accuracy given by users.

### 6 ACKNOWLEDGEMENT

This work was supported by the National Nature Science Foundation under contract 69684006 and CAO Guangbiao Foundation under contract 2232-9714-1016 to Zhejiang University. Special thanks go to the members of CIMS Laboratory, Zhejiang University.

### REFERENCES


<table>
<thead>
<tr>
<th>Intersection of segments and scan line</th>
<th>Do</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Skip if trend of segments is from A to B, otherwise put B twice into link</td>
</tr>
<tr>
<td>A          B</td>
<td>Put A into link if trend is from A to B, otherwise put B into link</td>
</tr>
<tr>
<td>A          B</td>
<td>Put A and B into link if trend is from A to B, otherwise skip</td>
</tr>
<tr>
<td>A          B</td>
<td>Skip if trend is from A to B, otherwise put B into link</td>
</tr>
<tr>
<td>A          B</td>
<td>Put A into link</td>
</tr>
<tr>
<td>A          B</td>
<td>Skip if trend is from A to B, otherwise put B into link</td>
</tr>
</tbody>
</table>

**Table 1 Intersection case between scan lines and segments**

**Fig. 1 Part 1**

**Fig. 2 Part 2**