Delaunay Triangulation for Curved Surfaces

Hao Chen and Jonathan Bishop

MARC Analysis Research Corporation
260 Sheridan Avenue, Suite 309, Palo Alto, CA 94306
emails: haochen@marc.com, jon@marc.com

Abstract

Surface triangular meshing plays an important role in the areas of computer graphics and engineering analysis. Traditionally, surface meshing is achieved by mapping meshes created in 2D parametric space onto surfaces. Care is taken in generating meshes in the parametric space and mapping them to surfaces because the transformation of geometry from the parameter space to the real space could be extended and twisted along some directions. Therefore, a good looking mesh in 2D parametric space could be very poor on a surface. In this paper, we present a meshing scheme which generate Delaunay type triangular surface mesh. For each triangle on a surface, its circumcircle is mapped into the parametric space, and the geometry of the mapped circle in the parametric space is approximated by an ellipse function. The triangulation in the parametric space is created and maintained using the property of empty circumellipse instead of empty circumcircle. Also surface curvature is used to control surface mesh density distribution. Therefore, the triangular mesh has a good approximation to the surface shape. The implementation of the approach results in very good-quality surface mesh.

Introduction

Surface triangular meshing is important for computer animation and engineering analysis. The 3D tetrahedral meshes are often created from the surface triangular mesh too. Traditionally, surface meshes are created by mapping meshes produced in the parametric space onto surfaces.

In [1], Zheng etc. use the mapping technique to generate surface meshes. With their meshing method, meshes are completely created in surface parametric space, and then mapped onto surfaces. For the triangulation update in 2D parametric space, the quality and size of mesh elements are evaluated using the 3D geometry to overcome the extending and twisting transform problems from 2D parametric space to surface. Swapping and smoothing techniques are also used to improve the mesh structures.

In addition to the mapping mesh method, different approaches on surface triangular meshing were presented in the last several years.

In [2], Lau etc. present a surface meshing scheme using the advancing front technique. The element generation is started from the edge segments on the boundary of a surface. The generation of new nodes from the front is controlled by the local curvature and new elements shape. Mesh elements are created on surfaces directly.

In [3], Chew extends the Delaunay empty circumcircle property from planar 2D to surface with a triangle. The triangulation update is implemented on surfaces directly. The difficulty of the method is how to map a triangle's circumcenter onto surface and how to check if a point is interior to a circumcircle on surface. Chew presents an iterative method. But it may not converge for very curved surfaces.

In this paper, we present a different approach to generate triangular surface mesh based on the mapping technique and Delaunay triangulation method. The triangular mesh generation is implemented in 2D parametric space. The triangulation update in 2D plane is controlled by the Delaunay empty circumcircle property. Due to the twisting transform problem between the parametric space and the real space, a triangle's circumcircle on a surface may not be a circle in the parametric space. In our approach, an ellipse function is used to approximate the mapped circle in 2D parametric space. Therefore, the Delaunay triangulation in 2D parametric space is created and maintained using the circumellipses instead of the circumcircles. When all mesh elements satisfy updating criteria, the triangulation is considered to be finished. Finally, triangular meshes are mapped onto surfaces.

The following sections are organized as:

Curve Division In the real applications, all surfaces must have the bounding curves. Our meshing scheme starts with discretizing the bounding curves. The curve discretization is controlled by the curve curvature.

Meshing in 2D The triangulation meshing is implemented in surface parametric space (2D planar space). In this section, we present the 2D planar Delaunay type triangulation meshing scheme. The Delaunay triangulation is a procedure including two respects: how to triangulate the existing points, when and where to insert the new points into a triangulation. Our 2D meshing scheme is similar to the one in [4] except that no new points are inserted on boundary edges. The Delaunay type triangulation is created and maintained using the property of empty circumellipses instead of empty circumcircles.

Projection of A Circumcircle from Surface to Its Parametric Plane In this section, we present how to compute an ellipse function in 2D parametric space to approximate an mapped circle from a surface.

Finally, we present several examples to illustrate the results of the meshing approach.

Curve Division

Our meshing for a bounded surface starts with discretizing its boundary curves, and then meshing the interior domain bounded by the discretized boundary segments. During the meshing process, the boundary curve discretization is not changed. This makes the merging of meshes over two neighboring surfaces very easy.

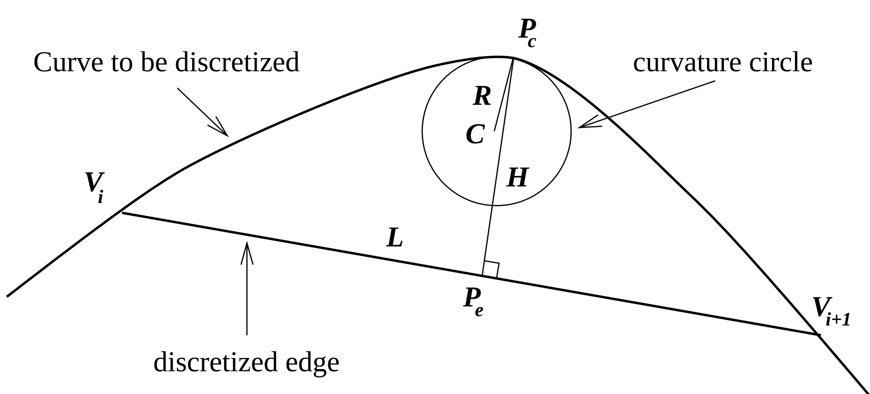

Figure 1: Curve discretization

The curve discretization is controlled by checking the curve deviation from the discretized segments. If all the deviation values are smaller than a specified value, the curve discretization is considered to be finished.

As shown in Figure 1, Point P_c is the middle point between the two end vertices V_i and V_{i+1} of a discretized edge segment, and point P_e is the projection of P_c onto edge $V_i V_{i+1}$. R is the curvature radius at point P_c, H is the arc height of edge $V_i V_{i+1}$ to point P_c, L is the length of edge $V_i V_{i+1}$.

The formula $\gamma = H/R$ can be used to evaluate how close the discretized edge is to the curve. Because computing curvature is time-consuming, a simplified form $\gamma = H/L$ can be used.

In the real application, three parameters are required for the curve division, maximum edge length, minimum edge length, and γ value. Initially, a curve is divided into N segments, where

$$N = \frac{curve\ length}{maximum\ length\ of\ edge\ segment} \tag{1}$$

Then check γ for each edge. If γ is greater than the specified value, further middle division of the curve portion is applied until all γs are less than the specified value or half of the edge length is smaller than the minimum edge length.

After all the boundary curves are discretized, the discretized points are not moved and no new points are added on the bounding curves. The curve discretizing method makes the later surface meshing to create the consistent mesh densities in the areas close to the bounding curves.

Meshing in 2D

Our surface triangular meshing algorithm is implemented in 2D parametric space. It is known that Delaunay triangulation returns the most quality promising mesh with a given set of points in 2D. Delaunay triangulation has several properties. One of them is the empty circumcircle property [6] which is defined as no triangle's circumcircles contain points interiorly. This property has been used to create the Delaunay triangulation.

Procedure

In the real applications, two steps are involved to create a good quality mesh for a bounded domain. The first step is to triangulate the discretized points on the boundary curves, and the second step is to insert new points interior to the boundary, and update the triangulation. The purpose to add the new points into a triangulation is to remove those triangles in poor shape. Here a pseudo code is presented to explain our meshing procedure.

```
Initialize:
    discretize the surface bounding curves to create the seed points
    triangulate the seed points
repeat:
     while deviation of a triangle to surface > γ
              or minimum angle of a triangle < α
              or size of a triangle > R
       add a new point at the triangle's circumcenter
       update the triangulation
until no triangles violate the three conditions
```

The parameters γ and α can be the user specified values. R is computed based on the point spacing values.

Update of Delaunay triangulation

In this section, we briefly explain how to update Delaunay triangulation with a new point.

Figure 2 shows that there are five existing triangles $T_{V_1 V_2 V_7}$, $T_{V_2 V_3 V_4}$, $T_{V_2 V_4 V_7}$, $T_{V_4 V_6 V_7}$, and $T_{V_4 V_5 V_6}$. Point V_n is a new point to be added to the triangulation. Firstly, the triangulation is searched for the triangles whose circumcircles contain the new point V_n. Here all the five triangles are found that their circumcircles contain V_n. Then all the edges separating the point V_n from their triangles are removed. So edges $E_{V_2 V_4}$, $E_{V_2 V_7}$, $E_{V_4 V_7}$ and $E_{V_4 V_6}$ are removed. A star shape polygon $P_{V_1 V_2 V_3 V_4 V_5 V_6 V_7}$ with regard to the point V_n is created. Finally, the point

V_n is connected to all the vertices V_i of the polygon. The triangulation update with the insertion of the point V_n is finished.

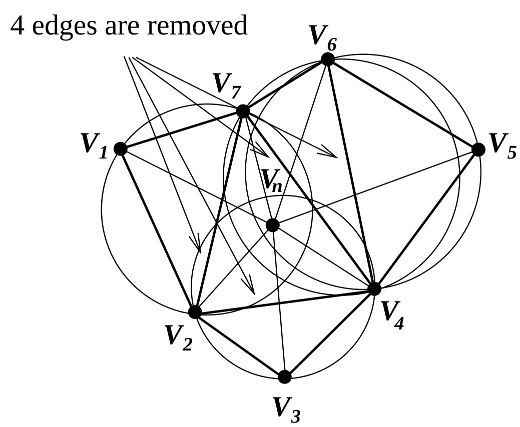

Figure 2: Triangulation update

Spacing function

Delaunay triangulation promises to return good quality mesh over a given set of points. But other criteria are required to determine where to add new points into the triangulation to create good mesh transition from the low mesh density area to the high mesh density area. Generally, spacing value is used to control the mesh transition.

After the discretization of boundary curves is finished, all points on the boundary are assigned a spacing value which is the average of the two distances from it to its two neighboring points on the boundary. As a new point is inserted into the triangulation, a star shape polygon is produced for updating the triangulation. The spacing value of the new point is the average of spacing values of its vertices on the star shape polygon. The formula is

$$S = \frac{\sum_{i=1}^{n} \frac{S_i}{D_i}}{\sum_{i=1}^{n} \frac{1}{D_i}} \tag{2}$$

where S_i is the spacing value of the vertex V_i of the star shape polygon, and D_i is the distance from the new point to the vertex V_i. The spacing function is used as one factor to determine if a new point can be add into an existing triangulation.

Criterion for Triangulation Update

Two criteria are used to check whether a triangle should be removed from the current triangulation by inserting a new point at the triangle's circumcenter. One criterion is checking the minimum angle of a triangle. If the minimum angle is smaller than a specified angle α, a new point at the triangle's circumcenter is added and triangulation is updated. In [4], Ruppert indicates that the minimum angle α can be up to 30 degrees. The other criterion is checking the size of a triangle. If its size is too big, a new point is added at its circumcenter. The size checking uses the spacing values of the three vertices of triangle. Its formula is

$$R \; = \; C * \sum_{i=1}^{3} S_i \tag{3}$$

where S_i is the spacing value at vertex V_i of the testing triangle. if circum radius of a triangle is bigger than R, a point at its circumcenter is added.

One more criterion is set up for the triangulation update with the consideration of good mesh approximation to the surface shape. The mesh density should be controlled by surface curvature. At the high curved area, the smaller size triangular elements may be necessary to well approximate the surface shape. Our surface curvature control method is consistent with the curve division algorithm presented above. γ is computed with the three edges of a triangle. If the maximum of the three γ values is bigger than a specified value, a new point is added at the triangle's circumcenter.

The triangulation update is implemented iteratively until all triangles satisfy the three criteria.

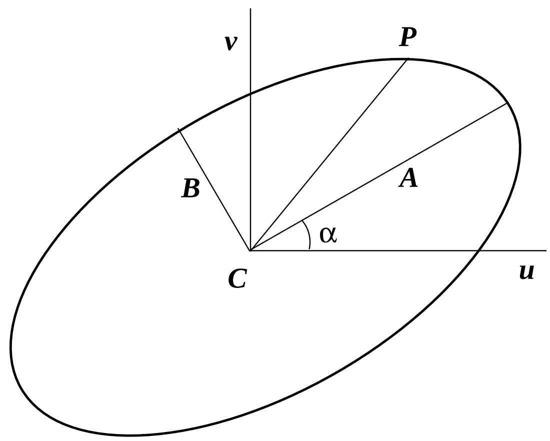

Figure 3: Ellipse definition

Projection of A Circumcircle from Surface to Its Parametric Plane

Our surface meshing is implemented in 2D parametric space. In the above section, we present how the Delaunay type triangulation is created and how new points should be added to update triangular mesh structure. One important property of Delaunay triangulation is empty circumcircle. Due to the extending and twisting transform problem between the real space and the parametric space, a triangle's circumcircle is not in circular shape after it is transformed into 2D parametric space. We use an ellipse function to approximate the mapped circle in 2D parametric space. By doing this, a triangulation satisfying the empty circumellipse property in 2D parametric space becomes the one satisfying empty circumcircle property on surfaces. Therefore, a good looking surface triangular mesh can be obtained. The method is more robust because all the geoemtric computations are implemented in planar 2D space.

As a circle on a surface is mapped back into 2D parametric space, its geometry could be very complicated. An expensive method could map discretized points on the circle back into 2D parametric space, and fit a curve to the mapped points. We use an ellipse to approximate the mapped circle. The following parametric form is used to define an ellipse function:

$$
\begin{aligned}
u &= u_c + A\,cos(t)\,cos(\alpha) - B\,sin(t)\,sin(\alpha) \\
v &= v_c + A\,cos(t)\,sin(\alpha) - B\,sin(t)\,cos(\alpha)
\end{aligned} \tag{4}
$$

where u_c and v_c are the coordinates of the ellipse center C, A and B are the lengths of half of the major and minor axes respectively. The ellipse major axis is inclined with an angle α relative to the u direction. t is the parametric variable. To get an implicit form of the ellipse function, t is canceled out by combining the two equations into:

$$
\begin{aligned}
G \;:=\; & k^2[(u-u_c)^2 sin^2(\alpha) - (u-u_c)(v-v_c)sin(2\alpha) + (v-v_c)^2 cos^2(\alpha)] \\
& + (u-u_c)^2 cos^2(\alpha) + (v-v_c)^2 sin^2(\alpha) + (u-u_c)(v-v_c)sin(2\alpha) \\
& - D = 0
\end{aligned} \tag{5}
$$

where $k = A/B$ and $D = A^2B^2$.

To compute a triangle's circumellipse function G, 5 parameters should be determined. With a triangle on a surface, its local principle curvature direction α and the ratio k of the two principle curvatures can be obtained. A triangle's circumellipse must pass its three vertices. Three equations can be established to compute the three parameters u_c, v_c, and D. Then the circumellipse G is completely determined.

Using the computed circumellipse to maintain the Delaunay triangulation in 2D parametric space simulates the direct Delaunay triangulation on surfaces. Delaunay triangulation on surfaces may not be robust in implementations as indicated in [3]. This approach avoids those problems because all the geometric computations are implemented in 2D space.

Examples

In the paper, we present several examples to show the meshing results. In Figure 4, the curvature is high in the bending area and low in the flat area. The curvature variation is very steep from the bending area to the flat area. The curve division produces a good curve discretization. The mesh density is consistent with the curvature distribution. The mesh transition from the high curvature area to the low curvature area is smooth. In Figure 5, curvature is high only in the area close to the central peak. The mesh shows a good shape approximation to the surface. In Figure 6, the curve division has high density on the top boundary curve, and lower density on the bottom boundary curve. The mesh transition is distributed smoothly from the high divided boundary curve to the low divided boundary curve. In Figure 7, the surface has relatively uniform curvature distribution. The resulting mesh density is uniform too. All these examples show that using the meshing algorithm produces a good quality triangular surface mesh.

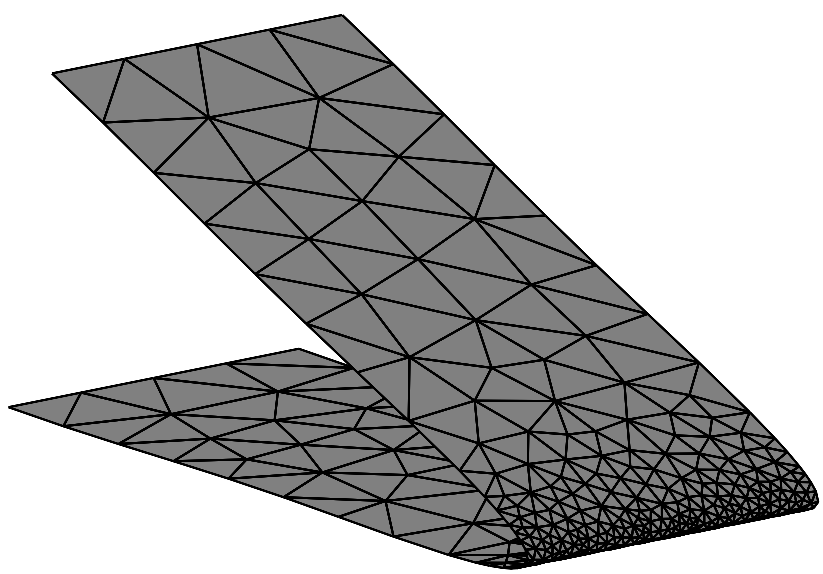

Figure 4: Bending surface: high density mesh around the bending area

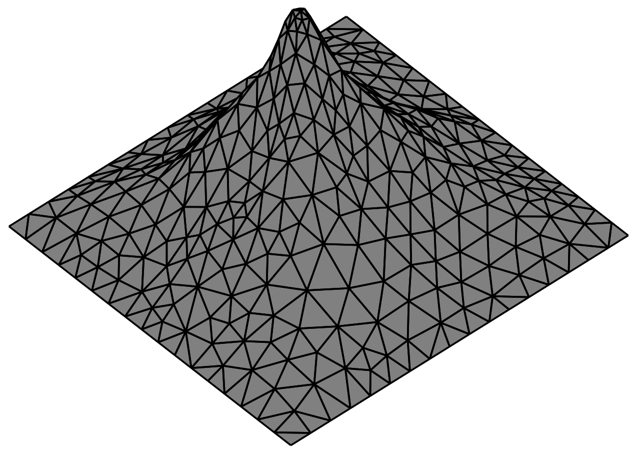

Figure 5: Surface with a peak: small size elements hold the shape of the peak

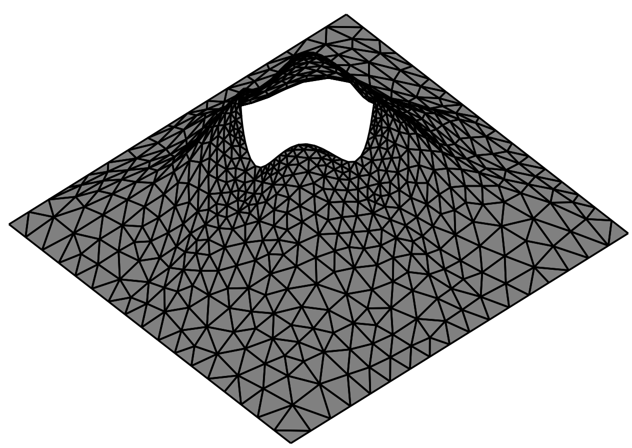

Figure 6: Surface with a hole: smooth mesh transition from boundary curves with high density division to boundary curves with low density division

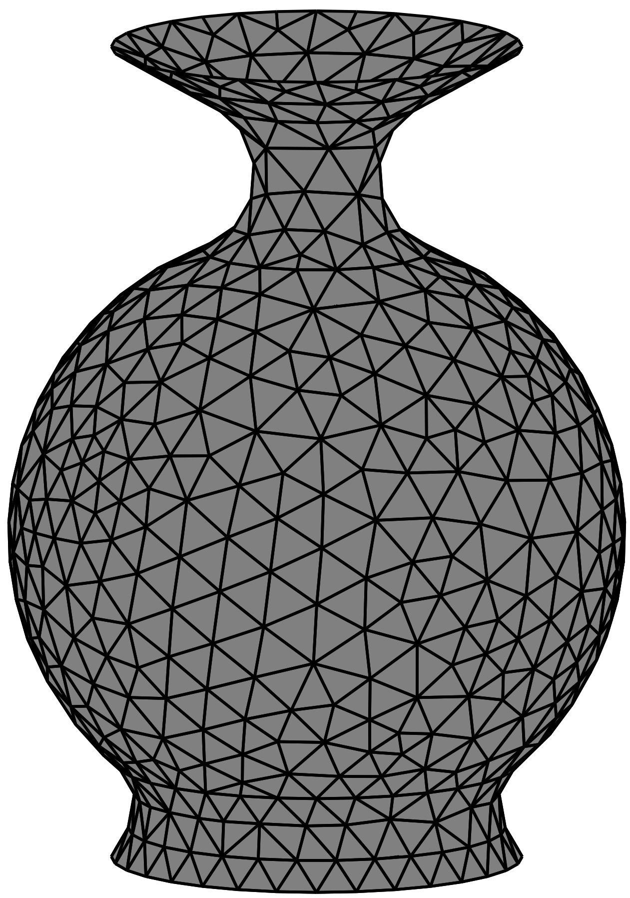

Figure 7: Meshes over surface with uniform curvature distribution

Conclusion

In this paper, we present an algorithm to mesh the parametric surfaces. The meshing is implemented in 2D parametric space only which insures the robustness of the geometric computations. The mesh quality is maintained by the Delaunay triangulation property. Because the circumcircles are replaced with the circumellipses in 2D parametric space during the triangulation update, the meshes satisfy the Delaunay triangulation property over the surfaces. The surface curvature is also used to make the mesh density to be consistent with the surface shape.

References

[1] Y. Zheng, R. W. Lewis, D. T. Gethin, "Three-Dimensional Unstructured Mesh Generation: Part 2: Surface meshes", *Compu. Methods Appl. Mech. Engrg.*, 134, pp. 269-284, 1996

[2] T. S. Lau and S. H. Lo, "Finite Element Mesh Generation over Analytical Curved Surfaces", *Computers & Structures*, Vol. 59, No. 2, pp. 301-309, 1996

[3] L. Paul Chew, "Guaranteed-Quality Mesh Generation for Curved Surfaces", *9th Annual Computational Geometry*, CA, USA, May 1993

[4] Jim Ruppert, "A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation", *Journal of Algorithms*, Vol. 18, pp. 548-585, 1995

[5] K. Sugihara and H. Inagaki, "Why Is the 3D Delaunay Triangulation Difficult to Construct", *Information Processing Letters*, 54, pp. 275-280, 1995

[6] D. F. Watson, "Computing the n-dimensional Delaunay Tessellation with Application to Voronoi Polytopes", *Computer Journal*, Vol. 24, No. 2, pp.167-172, 1981