Decomposing trimmed surfaces using the Voronoï tesselation

Po-Yu Tsai [†]
Bernd Hamann [‡]

[†] NSF Engineering Research Center for
Computational Field Simulation,
Mississippi State University,
P.O. Box 6176,
Mississippi State, MS 39762, USA,

[‡] Department of Computer Science,
University of California,
Davis, CA 95616-8562, USA.

Abstract

Many applications deal with the rendering of trimmed surfaces and the generation of grids for trimmed surfaces. Usually, a structured or unstructured grid must be constructed in the parameter space of the trimmed surface. Trimmed surfaces not only cause problems in the context of grid generation but also when exchanging data between different CAD systems. This paper describes a new approach for decomposing the valid part of the parameter space of a trimmed surface into a set of four-sided surfaces. The boundaries of these four-sided surfaces are line segments, segments of the trimming curves themselves, and segments of bisecting curves that are defined by a generalized Voronoi diagram implied by the trimming curves in parameter space. We use a triangular background mesh for the approximation of the bisecting curves of the generalized Voronoi diagram.

1. Introduction

This paper is concerned with the approximation of parametric surfaces containing trimming curves by a set of surfaces that do not contain trimming curves. Trimmed surfaces arise frequently in real-world applications. Typically, they are the result of surface-surface intersection (SSI). Complex geometries are defined in terms of thousands of surfaces which might intersect each other. The intersection curves are usually defined in the parameter space of the surfaces, *e.g.*, by a set of planar Bézier, B-spline, or NURBS (non-uniform rational B-spline) curves.

The algorithm presented in this paper has various potential applications. Many CAD systems cannot represent trimmed parametric surfaces *implicitly*, *i.e.*, as parametric surfaces with the trimming curves defined in parameter space. This causes a problem when exchanging trimmed surface data between CAD systems. Grid generation algorithms also have to handle trimmed surfaces, and it is important to generate grids in the valid part only. Furthermore, certain grid lines should conform to the given trimming curves. Rendering algorithms define yet another type of algorithms that have to deal with trimmed surfaces: it is essential to use only the valid part for surface polygonization and rendering. We present a new method that decomposes the valid part of a trimmed surface by a set of untrimmed, four-sided surfaces. The decomposition into a set of *basic* surfaces is done in parameter space, and the union of all these basic surfaces defines the valid part of a trimmed surface exactly. We use the term ``valid part'' to refer to the part that remains when disregarding all holes implied by the trimming curves.

This paper discusses an approach for the representation of trimmed surfaces. The complement of the part that is ``cut out'' by the trimming curves is defined by means of decomposing the valid part of the parameter space into a set of four-sided regions. In the following, only tensor product surfaces are considered. They are denoted by

$$\mathbf{s}(u,v) = \big(x(u,v), y(u,v), z(u,v)\big) = \sum_{i=0}^{m} \sum_{j=0}^{n} \mathbf{d}_{i,j}\, \phi_i(u)\, \psi_j(v), \quad u,v \in [0,1], \ (1)$$

where $d_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j})$ and $\phi_i(u)$ and $\psi_j(v)$ could be Bernstein-Bézier polynomials, B-spline basis functions, or even rational B-spline basis functions (see [Farin '93]). It is assumed that s is C^0 continuous.

The closed trimming curves in parameter space are denoted by

$$\mathbf{c}_k(t) = \big(u(t), v(t)\big) = \sum_{i=0}^{n_k} \mathbf{d}_i^k\, \psi_i(t), \quad t \in [0,1], \ \ k = 0, ..., K, \qquad (2)$$

where $d_i^k = (u_i^k, v_i^k)$ and $c_k(0) = c_k(1)$. It is assumed that the rotation number of all trimming curves is the same, *i.e.*, they have the same orientation. Each trimming curve must be at least C^0 continuous but can have tangent discontinuities. A trimming curve must not intersect another trimming curve and must not self-intersect. In most practical applications, there is one trimming curve enclosing the region that contains all the other trimming curves, which is assumed to be c_0. If this enclosing trimming curve is not explicitly defined, the boundary of the parameter space is chosen to be c_0 (*i.e.*, c_0 is the piecewise linear curve given by the four line segments **v=0, u=1, v=1,** and **u=0**). Fig. 1 shows the trimming curves of a trimmed surface in physical and parameter space.

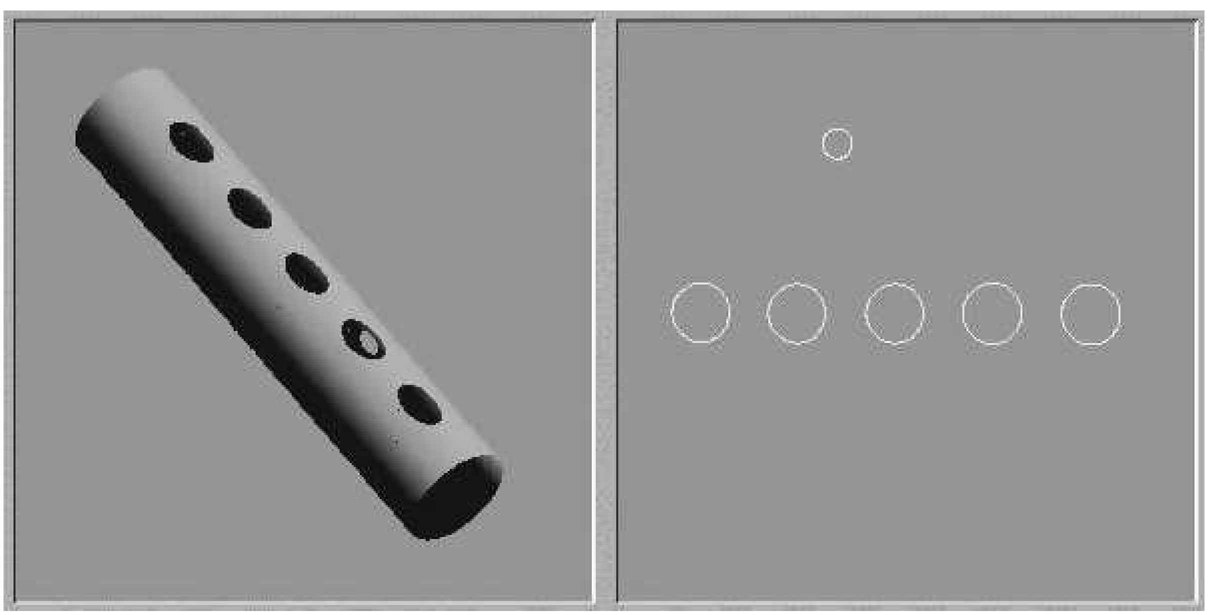

Fig. 1. Trimming curves in physical (left) and parameter space (right).

The approach described in this paper is similar to the construction of planar Voronoï and power diagrams in the sense that a tesselation of the valid part of the parameter space of a trimmed surface is computed. These diagrams are described in detail in [Aurenhammer '87], [Edelsbrunner '87], and [Preparata & Shamos '90]. This paper presents a new method for the construction of the curved boundaries (bisecting curves) defining the tiles associated with the trimming curves. A computationally efficient technique is used for generating a finite set of points on each bisecting curve. The technique is based on shortest distance computations for a finite set of points on a rectilinear grid in parameter space.

Typical generalizations of Voronoï diagrams in the plane deal with the construction of Voronoï diagrams for sets of points, line segments, polygons, circles, and more general planar curves. The construction of Voronoï diagrams for such sets is discussed in [Farouki & Johnstone '94], [Klein *et al.* '93], [Lee & Drysdale '81], [Leven & Sharir '86], [Sharir '85], [Srinivasan & Nackman '87], and [Yap '87].

An algorithm for rendering trimmed surfaces by using quadrilateral and triangular elements is described in [Rockwood *et al.* '89]. In [Baehmann *et al.* '87], a two-dimensional (2D) mesh generation algorithm is described that automatically discretizes 2D regions containing trimming curves based on the identification of certain geometrical features of the trimming curves, *e.g.*, slope discontinuities. A technique utilizing a combined ``triangulation-quadrangulation" strategy of the valid part of the parameter space of a trimmed surface is presented in [Vries-Baayens & Seebregts '92]. A method for representing a trimmed NURBS (non-uniform rational B-spline) surface by a set of Bézier patches is discussed in [Hoschek & Schneider '90] in the context of data exchange: trimmed rational surfaces are approximated by non-rational surfaces.

Curve and surface design techniques used in this paper are covered in [Boehm & Prautzsch '94] and [Farin '93]. Various solutions to the SSI problem are described in [Barnhill '92]. A survey of SSI algorithms is provided in [Patrikalakis '93].

2. Problem statement and definitions

The trimming curves define a simply connected region in parameter space. The goal is to represent this region by a set of planar, four-sided surfaces whose union is the valid part of the parameter space. Such a surface is referred to as a *parameter surface* and is denoted by

$$\mathbf{u}_l(\xi,\eta) = \left(u_l(\xi,\eta), v_l(\xi,\eta)\right) = \sum_{i=0}^{m_l}\sum_{j=0}^{n_l} \mathbf{d}^l_{i,j}\,\overline{\phi}_i(\xi)\,\overline{\psi}_j(\eta), \qquad \xi,\eta\in[0,1], \quad (3)$$

where $\mathbf{d}^l_{i,j} = (u^l_{i,j}, v^l_{i,j})$ Thus, the part of a surface $\mathbf{s}$ that is implied by the parameter surface $\mathbf{u}_l$ is given by

$$\mathbf{s}(\mathbf{u}_l) = \mathbf{s}\left(u_l(\xi,\eta), v_l(\xi,\eta)\right) = \sum_{i=0}^{m}\sum_{j=0}^{n} \mathbf{d}_{i,j}\,\phi_i\left(u_l(\xi,\eta)\right)\psi_j\left(v_l(\xi,\eta)\right), \quad \xi,\eta\in[0,1]. \quad (4)$$

The main problem to be solved is the generation of the boundary curves of the parameter surfaces. This problem can be solved using a generalization of the *Voronoï diagram* of a point set. When dealing with trimmed surfaces, the trimming curves define the set for which a tesselation, a generalized Voronoï diagram, must be computed. The tile boundaries in a planar *Voronoï diagram* implied by a point set are obtained from the perpendicular bisectors of all possible point pairs (see [Edelsbrunner '87] and [Preparata & Shamos '90]).

Generalizations of this ``standard" Voronoï diagram are obtained when the elements for which a tesselation is to be constructed are points, line segments, circles, polygons, and more general curves. Fig. 3 shows Voronoï diagram for a set of points and a set of circles (see [Aurenhammer '87] and [Facello '95]).

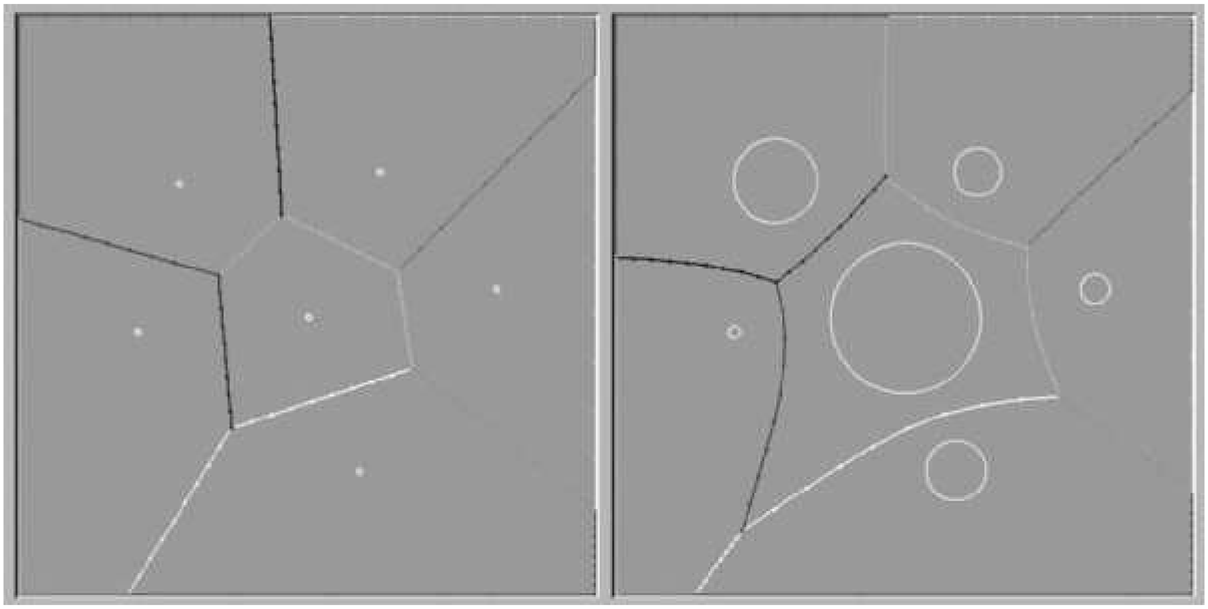

Fig. 2. Voronoï diagram for set of points and set of circles. trimming curves.

Voronoï diagrams introduce *tiles* around each element (points, line segments, circles, etc.) according to some distance measure. A tile is defined as the region that contains all the points being closer to a particular element than any other element. The tile boundary is used to subdivide a tile into a set of four-sided planar surfaces whose union represents the area between the element and the element's tile boundary. Each four-sided surface can be constructed by subdividing the tile boundary curve into segments and generating additional curves connecting end points of the tile boundary segments and points on the element. This principle is shown in Fig. 3.

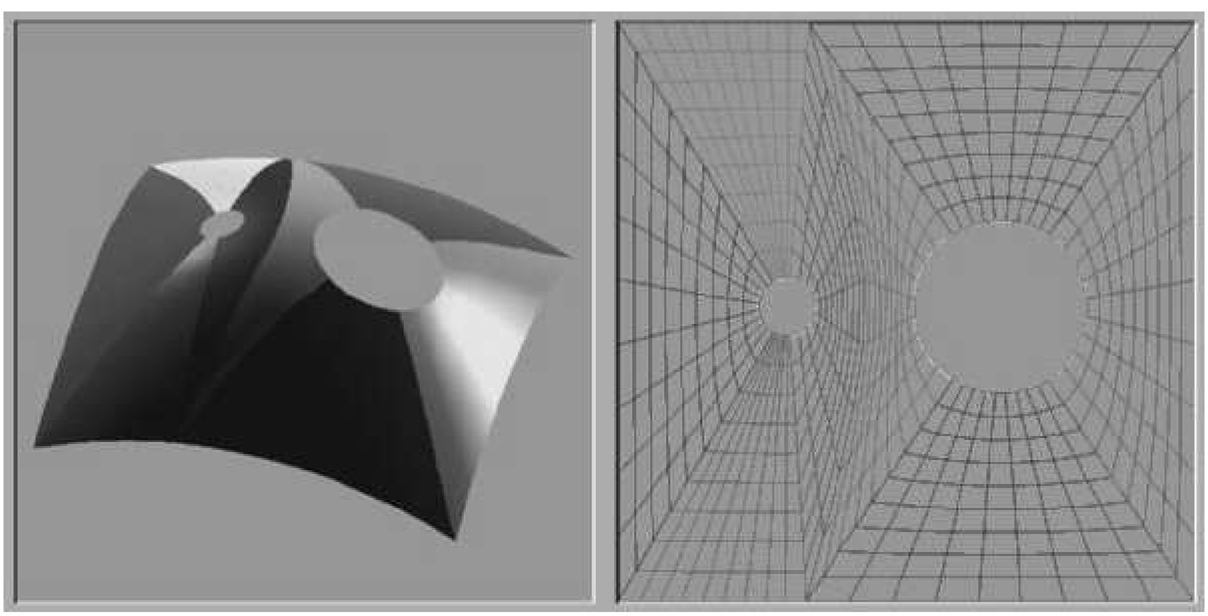

Fig. 3. Four-sided parameter surfaces in tiles around trimming curves.

3. Computing the Voronoï diagram for a set of trimming curves

An efficient algorithm is needed for the generation of the tile boundaries around each trimming curve in parameter space. First, tiles are constructed around the trimming curves $c_1, c_2, c_3,$ ..., and $c_K.$ The final tiles are obtained by intersecting the tile boundary curves in the Voronoï diagram for $c_1, c_2, c_3,$ ..., and c_K with the enclosing trimming curve $c_0.$

Initially, the trimming curve c_0 is not considered in the construction of the Voronoï diagram. The generation of the Voronoï diagram is based on the computation of the intersections of bisectors with the edges in a triangulation of the parameter space $[0, 1] \times [0, 1].$ The vertices in this triangulation are labeled according to the index of the closest trimming curve. The labels are used to determine whether there is an intersection between bisectors and edges in the triangulation. The intersections between the edges and the bisectors are computed and properly connected, thus defining the topology and an initial approximation of the Voronoï diagram.

Multiple bisectors can intersect the same edge in the triangulation, and multiple bisectors can intersect in the interior of a triangle. These cases are covered by Algorithm 3.1. described below. Algorithm 3.1. does not consider the case of one bisector intersecting the same edge multiple times. It turns out that this is not necessary for obtaining the approximation of the Voronoï diagram, which requires these steps:

- (i) Construction of a triangulation of the parameter space
- (ii) Extraction of all triangles whose three vertices all lie in the valid part of the parameter space
- (iii) Labeling each vertex in the triangulation with the index of the closest trimming curve
- (iv) Computation of intersections between bisectors and edges in the triangulation using a recursive subdivision strategy
- (v) Computation of intersections of bisectors
- (vi) Computation of intersections between bisectors and curve
- (vii) Generation of piecewise linear and cubic B-spline approximations of all tile boundaries in the Voronoï diagram

Denoting the minimal distance of all possible pairs of trimming curves by $d_{min},$ the initial triangulation of the parameter space only contains edges that are shorter than $d_{min}/2.$ This is accomplished by subdividing the parameter square $[0, 1] \times [0, 1]$ into squares whose diagonal is shorter than $d_{min}/2$ and splitting each square into two triangles. Only triangles whose three vertices all lie in the valid part of the parameter space are considered for the following computations.

Each vertex in the triangulation is labeled according to the closest trimming curve. The square of the distance **d** between a vertex with coordinate vector (x, y) and a trimming curve $c_k(t)$ is given by $d^2(t) = (x - u_k(t))^2 + (y - v_k(t))^2, t \in [0, 1].$ The critical points of $d^2(t)$ are identified, and the associated distances are computed. In addition, one computes the distances to those points on c_k where slope discontinuities occur. The index of the trimming curve that has minimal distance to (x, y) is used as the label for this vertex. It turns out that the case of multiple trimming curves all having minimal distance to

(x, y) does not require a special case treatment.

The labels at each vertex in the triangulation are used to identify edges which are intersected by at least one bisector. It is possible that the three labels at a triangle's vertices are the same, that there are two different labels, or that they are all different. In the first case, it is assumed that no bisector intersects the triangle. In the second case, it is assumed that there are bisectors intersecting the two edges whose end points have different labels. In the third case, it is assumed that there are bisectors intersecting all three edges. Points lying on bisectors in the Voronoï diagram are computed based on Algorithm 3.1.

Algorithm 3.1. (*Computation of points on bisectors in Voronoï diagram*).

Input:

- trimming curves c_1, c_2, c_3, ..., c_K,
- triangulation of parameter space $[0, 1] \times [0, 1]$,
- label $I \in \{1, 2, 3, ..., K\}$ at each vertex in triangulation referring to closest curve c_I,
- tolerance ϵ;

Output:

- set of points on bisectors in Voronoï diagram;

for all triangles in the parameter space triangulation **do**
  if there are at least two different labels among I_1, I_2, and
  I_3 associated with the triangle's vertices v_1, v_2, and v_3
 { ○ compute the midpoints $m_{i,j}$ of the edges $e_{i,j} = \overline{v_i v_j}$,
   $(i, j) \in \{(1, 2), (2, 3), (1, 3)\}$;
   for all edges $e_{i,j}$ whose end points have different labels I_i and I_j **do**
   { ○ find the point $p_{i,j}$ on $e_{i,j}$ that has the same distance to c_{I_i} and c_{I_j};
     if there is no trimming curve that is closer to $p_{i,j}$ than both c_{I_i} and c_{I_j}
      ○ consider the point $p_{i,j}$ as a point on a bisector;
     else
      ○ replace the value of $m_{i,j}$ by the value of $p_{i,j}$;
   }
   if one has found at least one point $p_{i,j}$ for which neither c_{I_i} nor c_{I_j} is the closest curve
   { ○ split the triangle into the four subtriangles given by the triples
     $(v_1, m_{1,2}, m_{1,3}), (v_2, m_{2,3}, m_{1,2})$, $(\mathbf{v}_3, \mathbf{m}_{1,3}, \mathbf{m}_{2,3})$ and $(\mathbf{m}_{1,2}, \mathbf{m}_{2,3}, \mathbf{m}_{1,3})$;
    ○ find the intersections of bisectors with the edges of these subtriangles;
   }
 }

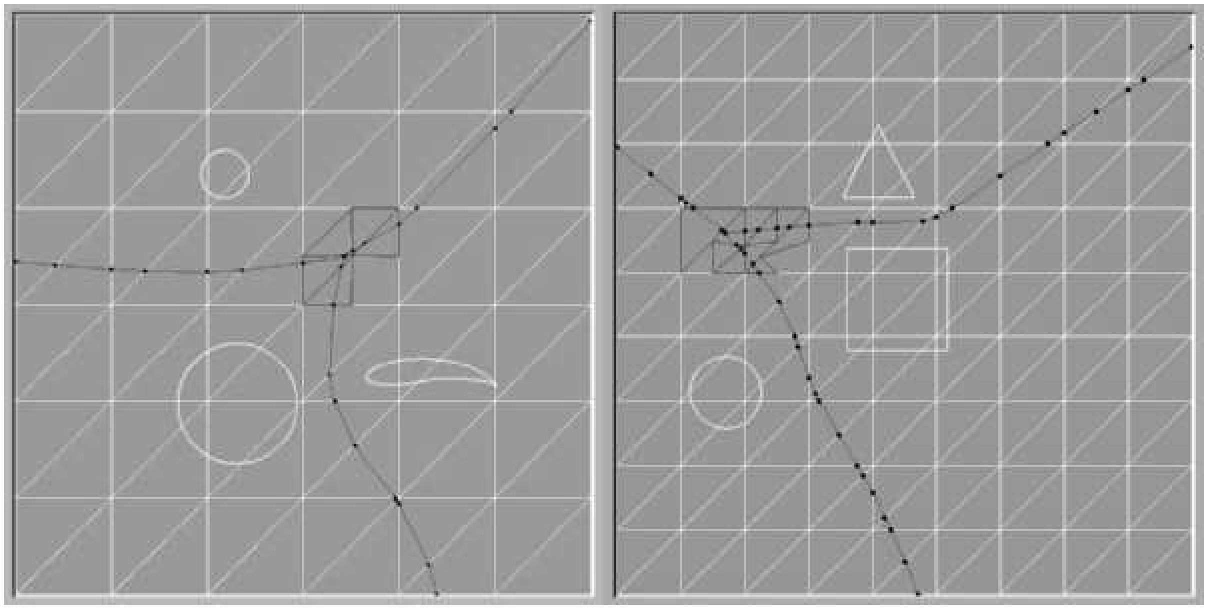

Fig. 4. Recursive subdivision of triangulations for generation of bisectors.

Fig. 4 shows the results of Algorithm 3.1. for two different configurations. A piecewise linear approximation of the Voronoï diagram is obtained by connecting the points resulting from Algorithm 3.1. If exactly two edges of a triangle contain each one point on a bisector, denoted by p_1 and p_2, then p_1 and p_2 are connected. If all three edges of a triangle each contain one point on a bisector, denoted by p_1, p_2, and p_3, then p_1, p_2, and p_3 are assumed to be lying on three different bisectors. In the second case, each of the three points is connected with the point q that is the intersection of three bisectors.

An iterative method is used to approximate the coordinates of q. The centroid of p_1, p_2, and p_3 is used as the initial approximation q^0 of q, and subsequent approximations q^i are obtained by repeatedly computing the three closest points on the trimming curves c_{I_1}, c_{I_2}, and c_{I_3} (*i.e.*, the trimming curves closest to p_1, p_2, and p_3) and replacing a previous approximation by the center of the circle passing through these three closest points. The method terminates when the Euclidean distance between two successive approximations q^i and q^{i+1} is smaller than ϵ. Whenever Algorithm 3.1. leads to a triangle whose longest edge is shorter than ϵ (one of the termination criteria of the algorithm) the centroid of such a triangle is considered to be the intersection of bisectors. Eventually, one obtains a piecewise linear approximation of all bisectors in the Voronoï diagram.

Based on the piecewise linear approximation of the Voronoï diagram and the curve segments of c_0, a cubic B-spline approximation is constructed for all tile boundary curves. The tile boundary curve associated with the trimming curve c_i is denoted by $\bar{c}_i$. The cubic B-spline representation of $\bar{c}_i$ is based on a chord length parametrization defined by the lengths of the line segments in the piecewise linear approximation (see [Farin '93]). Fig.5 shows the cubic B-spline curves approximating the tile boundaries for a configuration with an enclosing trimming curve c_0. The piecewise linear approximation of the Voronoï diagram is intersected with the enclosing trimming curve c_0, and the resulting curve segments on c_0 are used for the definition of the tile boundary curves around c_1, c_2, c_3, ..., and c_K.

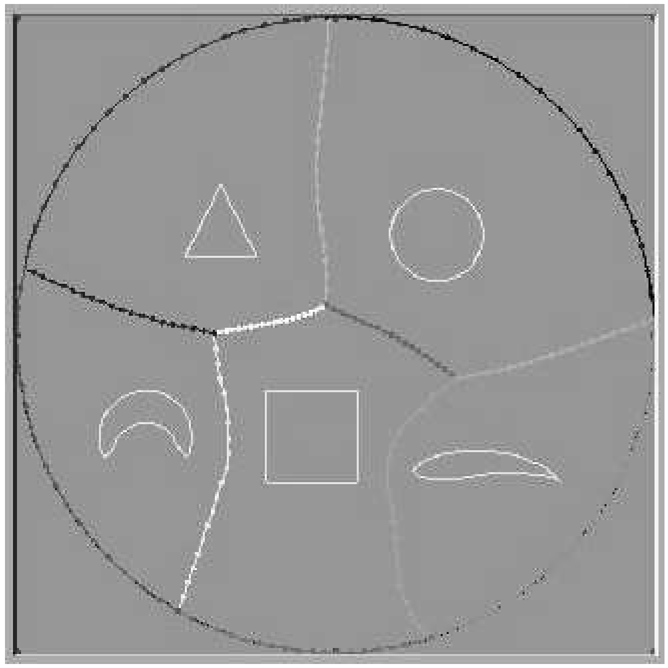

Fig. 5. Cubic B-spline approximation of Voronoï diagram for trimming curves.

In practical applications, one is often concerned with interior trimming curves that are not closed and intersect the outer trimming curve c_0. If this is the case, the intersections between those trimming curves and c_0 are computed, and the resulting curve segments on c_0 - in combination with the trimming curves that are not closed - define a new enclosing trimming curve c_0. Once this new curve c_0 has been computed, the construction of the Voronoï diagram follows the same principle as discussed above. Fig. 6 shows a configuration the trimming curve c_0 is intersected by four trimming curves.

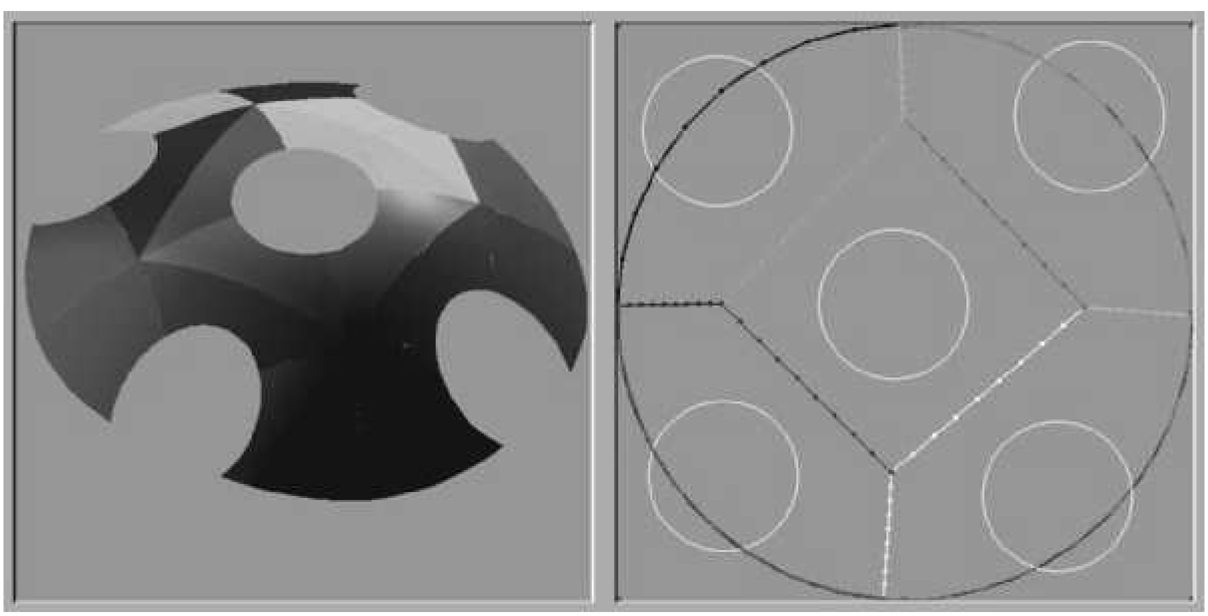

Fig. 6. Trimming curves intersecting enclosing trimming curve.

Once the Voronoï diagram is known, each tile can independently be decomposed into parameter surfaces. Thus, all tiles can be processed in parallel, and only the two curves c_i and $\bar{c}_i$ must be considered for the construction of the parameter surfaces inside a particular tile.

4. Computing the boundary curves for parameter surfaces

A *scan line algorithm* (see [Foley *et al.* '90]) is the basis for the construction of the parameter surfaces inside the tile associated with trimming curve c. The region between the trimming curve c and the tile boundary curve $\bar{c}$ is represented by a set of *ruled parameter surfaces*. They are obtained by (i) identifying local extrema in **v**-direction (and horizontal line segments) on c and $\bar{c}$; (ii) computing the intersections between horizontal lines passing through the local extrema (and the horizontal line segments) and c and $\bar{c}$; and (iii) constructing ruled parameter surfaces using the horizontal line segments inside the tile and curve segments on c and $\bar{c}$.

Fig. 7 shows a real-world example. The trimming curves, the tile boundary curves of the generalized Voronoï diagram, and the resulting curvilinear grids in parameter space are shown.

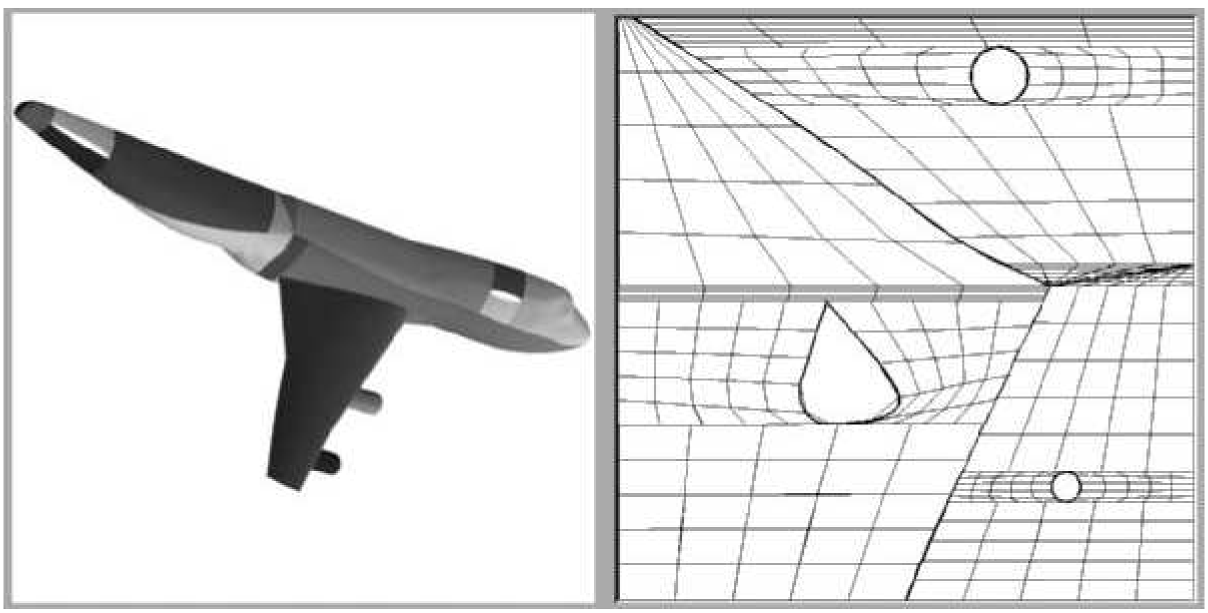

Fig. 7. Fuselage with multiple trimming curves.

5. Conclusions

A method for representing the region ``between" the trimming curves in the parameter space of a trimmed parametric surface has been presented. A generalized Voronoï diagram is computed for the set of trimming curves, and the resulting tiles associated with each trimming curve are subdivided into ruled parameter surfaces. The method has potential applications in the areas of surface rendering, grid generation, and data exchange.

6. Acknowledgements

This work was supported by the National Grid Project consortium and the National Science Foundation under contract EEC-8907070 to Mississippi State University. Special thanks go to the members of the research and development team of the National Grid Project, which is performed at the NSF Engineering Research Center for Computational Field Simulation, Mississippi State University.

References

Aurenhammer, F. (1987), Power diagrams: properties, algorithms, and applications, SIAM Journal on Computing 16, pp. 78--96.

Baehmann, P. L, Wittchen, S. L., Shephard, M. S., Grice, K. R. and Yerry, M. A. (1987), Robust, geometrically based, automatic two-dimensional mesh generation, International Journal for Numerical Methods in Engineering 24, pp. 1043--1078.

Barnhill, R. E. (1992), *Geometry Processing for Design and Manufacturing*, SIAM, Philadelphia, PA.

Boehm, W. and Prautzsch, H. (1994), *Numerical Methods*, A K Peters, Ltd., Wellesley, MA.

Edelsbrunner, H. (1987), *Algorithms in Combinatorial Geometry*, Springer-Verlag, New York, NY.

Farin, G. (1993), *Curves and Surfaces for Computer Aided Geometric Design*, third edition, Academic Press, San Diego, CA.

Farouki, R. T. and Johnstone, J. K. (1994), The bisector of a point and a plane parametric curve, Computer Aided Geometric Design 11(2), pp. 117--151.

Foley, J. D., van Dam, A., Feiner, S. K., and Hughes, J. F. (1990), *Computer Graphics*, second edition, Addison Wesley, Reading, MA.

Hoschek, J. and Schneider, F. J. (1990), Spline conversion for trimmed rational Bézier and B-spline surfaces, Computer Aided Design 22, pp. 580--590.

Klein, R., Mehlhorn, K. and Meiser, S. (1993), Randomized incremental construction of abstract Voronoï diagrams, Computational Geometry: Theory and Applications 3, pp. 157--184.

Lee, D. T. and Drysdale, R. L. (1981), Generalization of Voronoï diagrams in the plane, SIAM Journal on Computing 10(1), pp. 73--87.

Leven, D. and Sharir, M. (1986), Intersection and proximity problems and Voronoï diagrams, in: Schwartz, J. and Yap, C. K., eds., *Advances in Robotics*, Vol. I, Lawrence Erlbaum, London, U.K., pp. 187--228.

Patrikalakis, N. M. (1993), Surface-to-surface intersections, IEEE Computer Graphics and Applications 13 (1), pp. 89--95.

Piegl, L. A. (1991a), On NURBS: A survey, IEEE Computer Graphics and Applications 11(1), pp. 55--71.

Piegl, L. A. (1991b), Rational B-spline curves and surfaces for CAD and graphics, in: Rogers, D. F. and Earnshaw, R. A., eds., *State of the Art in Computer Graphics*, Springer-Verlag, New York, NY, pp. 225--269.

Preparata, F. P. and Shamos, M. I. (1990), *Computational Geometry*, third printing, Springer-Verlag, New York, NY.

Rockwood, A. P., Heaton, K. and Davis, T. (1989), Real time rendering of trimmed surfaces, ACM Transactions on Graphics 23, pp. 107--116.

Sharir, M. (1985), Intersection and closest-pair problems for a set of discs, SIAM Journal on Computing 14 (2), pp. 448--468.

Srinivasan, V. and Nackman, L. R. (1987), Voronoï diagram for multiply-connected polygonal domains I: Algorithm, IBM J. Res. Develop. 31(3), pp. 361--372.

Vries-Baayens, A. E. and Seebregts, C. H. (1992), Exact conversion of a trimmed nonrational Bézier surface into composite or basic nonrational Bézier surfaces, in: Hagen, H., ed., *Topics in Surface Modeling*, SIAM, Philadelphia, PA, pp. 115--143.

Yap, C. K. (1987), An $O(n \log n)$ algorithm for the Voronoï diagram of a set of simple curve segments, Discrete Computational Geometry 2, pp. 365--393.

Po-Yu Tsai
Wed Feb 14 00:31:25 CST 1996