Title
Decomposing Trimmed Surfaces Using the Voronoi Diagram and a Scan Line Algorithm

Authors
Hamann, Bernd
Tsai, P.-Y.

Publication Date
1998

Peer reviewed

NORTH-HOLLAND

Decomposing Trimmed Surfaces Using the Voronoï Diagram and a Scan Line Algorithm

Bernd Hamann[*,†] and Po-Yu Tsai[‡]

[*] *Co-Director*
Center for Image Processing and
Integrated Computing (CIPIC) and
Visualization Thrust Leader

[†] *Department of Computer Science*
University of California at Davis
Davis, California 95616-8562

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

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 Voronoï diagram implied by the trimming curves in parameter space. We use a triangular background mesh for the computation of the bisecting curves of the generalized Voronoï diagram.

APPLIED MATHEMATICS AND COMPUTATION 89:327–344 (1998)

655 Avenue of the Americas, New York, NY 10010 — 0096-3003/98/$19.00
PII S0096-3003(96)00259-7

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.

It must be emphasized again that the algorithm we present does *not* introduce any approximation errors: The algorithm decomposes the valid part of a trimmed surface in parameter space, and all trimming curves are preserved exactly. Thus, there will be *no* discontinuities along shared edges in physical space.

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

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

where $\mathbf{d}_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j})$ 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 [1] and [2]). It is assumed that $\mathbf{s}$ is C^0 continuous.

The closed trimming curves in parameter space are denoted by

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

where $\mathbf{d}_i^k = (u_i^k, v_i^k)$ and $\mathbf{c}_k(0) = \mathbf{c}_k(1)$. 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 $\mathbf{c}_0$. If this enclosing trimming curve is not explicitly defined, the boundary of the parameter space is chosen to be $\mathbf{c}_0$ (i.e., $\mathbf{c}_0$ is the piecewise linear curve given by the four line segments $v = 0$, $u = 1$, $v = 1$, and $u = 0$). Figure 1 shows the trimming curves of a trimmed surface in physical and parameter space.

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 [3, 4], and [5]. This paper presents a new

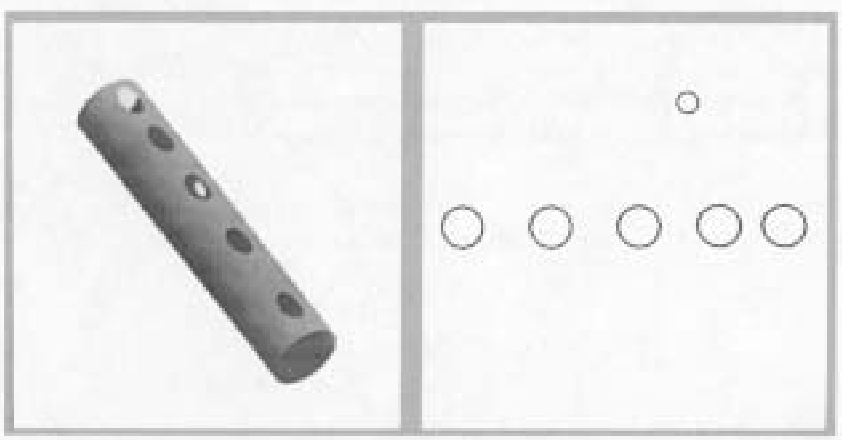

FIG. 1. Trimming curves in physical (left) and parameter space (right).

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 [6–12].

An algorithm for rendering trimmed surfaces by using quadrilateral and triangular elements is described in [13]. In [14], 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 [15]. A method for representing a trimmed NURBS (non-uniform rational B-spline) surface by a set of Bézier patches is discussed in [16]. In [16], the problem of data exchange is addressed; trimmed rational surfaces are approximated by non-rational surfaces.

Curve and surface design techniques used in this paper are covered in [17, 1, 2]. Various solutions to the SSI problem are described in [18]. A survey of SSI algorithms is provided in [18].

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

$$\mathbf{u}_l(\xi,\eta) = (u_l(\xi,\eta), v_l(\xi,\eta))$$

$$= \sum_{i=0}^{m_l} \sum_{j=0}^{n_l} \mathbf{d}^l_{i,j} \bar{\phi}_i(\xi) \bar{\psi}_j(\eta), \qquad \xi,\eta \in [0,1], \tag{3}$$

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

$$\mathbf{s}(\mathbf{u}_l) = \mathbf{s}(u_l(\xi,\eta), v_l(\xi,\eta))$$

$$= \sum_{i=0}^{m} \sum_{j=0}^{n} \mathbf{d}_{i,j} \phi_i(u_l(\xi,\eta)) \psi_j(v_l(\xi,\eta)), \qquad \xi,\eta \in [0,1]. \tag{4}$$

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 [4, 5]). 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.

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 Figure 2.

The following definitions are needed when using the Voronoï diagram for the representation of trimmed parametric surfaces.

Definition 2.1. Given a set of planar, closed, pairwise non-intersecting curves with rotation number +1 such that no curve lies in the interior of any other curve, the locus of points that have a smaller shortest (Euclidean) distance to curve $\mathbf{c}_k$ than to any other curve is called the *tile* T of $\mathbf{c}_k$, denoted by $T(\mathbf{c}_k)$.

If all curves $\mathbf{c}_k$ have a continuous tangent, the smallest shortest distance is equal to the *perpendicular distance*. Assuming that there are K such curves, $T(\mathbf{c}_k)$ is obtained as the intersection of $K-1$ half-spaces, i.e.,

$$T(\mathbf{c}_k) = \bigcap_{\substack{l=1 \\ l \neq k}}^{K} H(\mathbf{c}_k, \mathbf{c}_l), \tag{5}$$

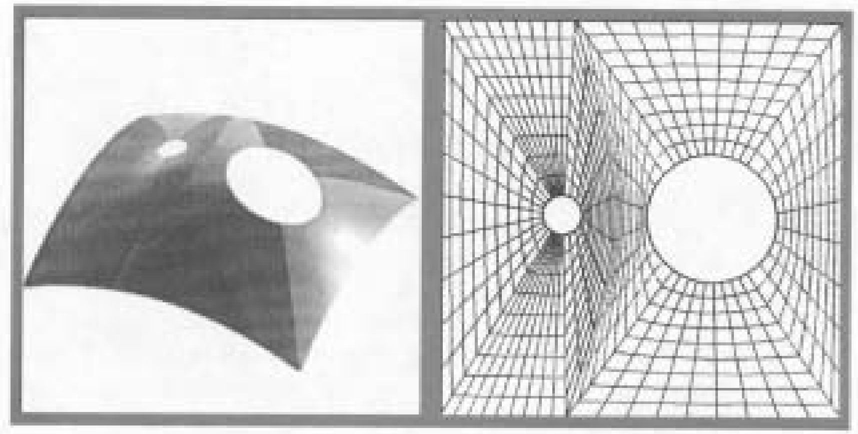

FIG. 2. Four-sided parameter surfaces in tiles around trimming curves.

where $H(\mathbf{c}_k, \mathbf{c}_l)$ is the half-space containing all points that have a smaller shortest distance to $\mathbf{c}_k$ than to $\mathbf{c}_l$ (see [5]).

DEFINITION 2.2. Given K curves as in Definition 2.1, the K tiles $T(\mathbf{c}_k)$ define the *generalized Voronoï diagram*. The curve separating the two half-spaces $H(\mathbf{c}_k, \mathbf{c}_l)$ and $H(\mathbf{c}_l, \mathbf{c}_k)$ is called the *bisector* of $\mathbf{c}_k$ and $\mathbf{c}_l$. The intersection points of bisectors are called *Voronoï vertices*.

Figure 3 shows the generalized Voronoï diagram for five arbitrary curves in the plane. Once the generalized Voronoï diagram (simply referred to as Voronoï diagram in the following) is known for a set of trimming curves, all tiles are represented by sets of four-sided parameter surfaces. The boundary curves of the parameter surfaces are defined by line segments, segments of the bisectors, and segments of the 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 $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \ldots,$ and $\mathbf{c}_K$. The final tiles are obtained by intersecting the tile boundary curves in the Voronoï diagram for $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \ldots,$ and $\mathbf{c}_K$ with the enclosing trimming curve $\mathbf{c}_0$.

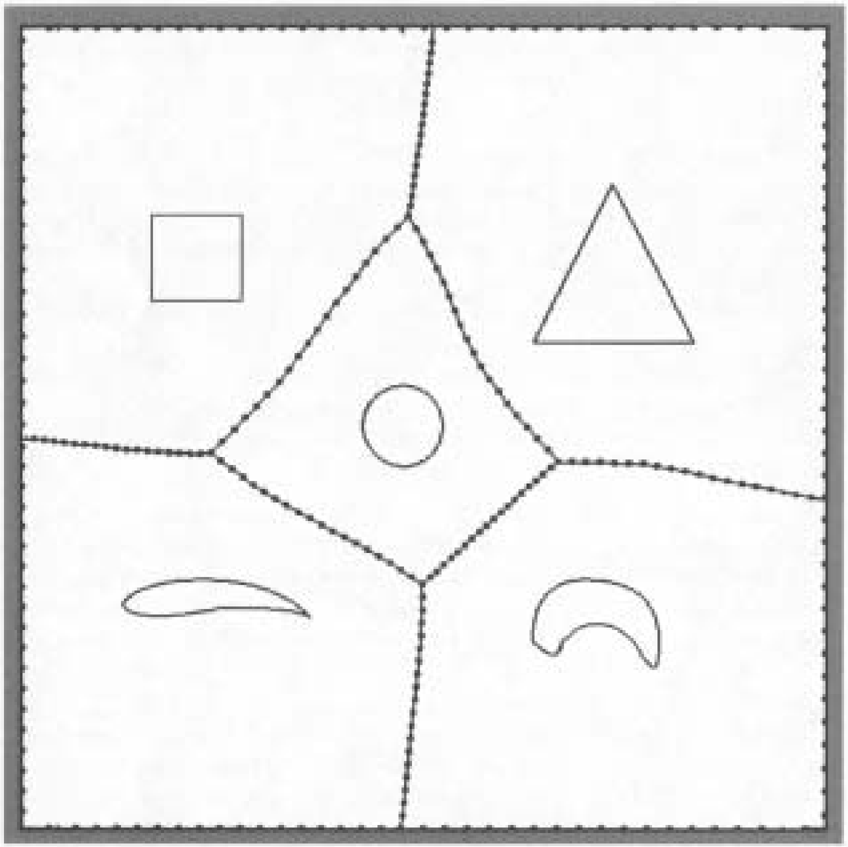

FIG. 3. Generalized Voronoï diagram for five curves.

Initially, the trimming curve $\mathbf{c}_0$ 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 $[0, 1] \times [0, 1]$
(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 $\mathbf{c}_0$
(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 $\mathbf{c}_k(t)$ 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 $\mathbf{c}_k$ 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 on the Voronoï diagram are computed based on Algorithm 3.1.

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

Input:
- trimming curves $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \ldots, \mathbf{c}_K$,
- triangulation of parameter space $[0, 1] \times [0, 1]$,
- label $I \in \{1, 2, 3, \ldots, K\}$ at each vertex in triangulation referring to closest curve $\mathbf{c}_I$,
- 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 $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$
  { *compute the midpoints $\mathbf{m}_{i,j}$ of the edges $e_{i,j} = \overline{\mathbf{v}_i\mathbf{v}_j}$, $(i, j) \in \{(1, 2), (2, 3), (1, 3)\}$;
   for all edges $e_{i,j}$ whose end points have different labels I_i and I_j **do**
   { *find the point $\mathbf{p}_{i,j}$ on $e_{i,j}$ that has the same distance to $\mathbf{c}_{I_i}$ and $\mathbf{c}_{I_j}$;
     if there is no trimming curve that is close to $\mathbf{p}_{i,j}$ than both $\mathbf{c}_{I_i}$ and $\mathbf{c}_{I_j}$
      *consider the point $\mathbf{p}_{i,j}$ as a point on a bisector;
     else
      *replace the value of $\mathbf{m}_{i,j}$ by the value of $\mathbf{p}_{i,j}$;
   }
   if one has found at least one point $\mathbf{p}_{i,j}$ for which neither $\mathbf{c}_{I_i}$ nor $\mathbf{c}_{I_j}$ is the closest curve
   { *split the triangle into four subtriangles given by the triples
     $(\mathbf{v}_1, \mathbf{m}_{1,2}, \mathbf{m}_{1,3})$, $(\mathbf{v}_2, \mathbf{m}_{2,3}, \mathbf{m}_{1,2})$, $(\mathbf{v}_3, \mathbf{m}_{1,3}, \mathbf{m}_{2,3})$, and $(\mathbf{m}_{1,2}, \mathbf{m}_{2,3}, \mathbf{m}_{1,3})$;
     *find the intersections of bisectors with the edges of these subtriangles;
     /*stop subdividing a triangle when each of its edges is intersected by at */
     /*most one bisector or the longest edge of a triangle is shorter than ε */
   }
}

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 $\mathbf{p}_1$ and $\mathbf{p}_2$, then $\mathbf{p}_1$ and $\mathbf{p}_2$ are connected. If all three edges of a triangle each contain one point on a bisector, denoted by $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$, then $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ are assumed to be lying on three different bisectors. In the second case, each of the three points is connected with the point $\mathbf{q}$ that is the intersection of three bisectors.

An iterative method is used to approximate the coordinates of $\mathbf{q}$. The centroid of $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ is used as the initial approximation $\mathbf{q}^0$ of $\mathbf{q}$, and subsequent approximations $\mathbf{q}^i$ are obtained by repeatedly computing the three closest points on the trimming curves $\mathbf{c}_{I_1}$, $\mathbf{c}_{I_2}$, and $\mathbf{c}_{I_3}$ (i.e., the trimming curves closest to $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$) 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 $\mathbf{q}^i$ and $\mathbf{q}^{i+1}$ is smaller than ε. Whenever Algorithm 3.1 leads to a triangle whose longest edge is shorter than ε (one of the

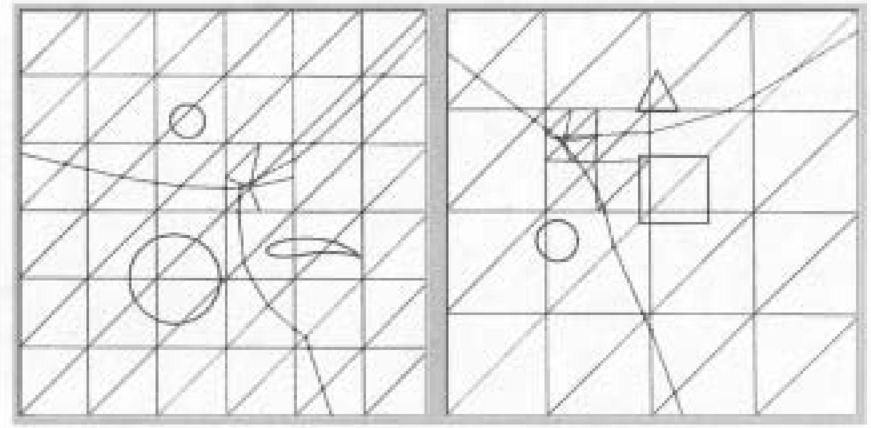

FIG. 4. Recursive subdivision of parameter space triangulations for generation of bisectors.

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 $\mathbf{c}_0$, a cubic B-spline approximation is constructed for all tile boundary curves. The tile boundary curve associated with the trimming curve $\mathbf{c}_i$ is denoted by $\bar{\mathbf{c}}_i$. The cubic B-spline representation of $\bar{\mathbf{c}}_i$ is based on a chord length parametrization defined by the lengths of the line segments in the piecewise linear approximation (see [1]). Fig. 5 shows the cubic B-spline curves approximating the tile boundaries for a configuration with an enclosing trimming curve $\mathbf{c}_0$. The piecewise linear approximation of the Voronoï diagram is intersected with the enclosing trimming curve $\mathbf{c}_0$, and the resulting curve segments on $\mathbf{c}_0$ are used for the definition of the tile boundary curves around $\mathbf{c}_1$, $\mathbf{c}_2$, $\mathbf{c}_3, \ldots,$ and $\mathbf{c}_K$.

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 $\mathbf{c}_i$ and $\bar{\mathbf{c}}_i$ must be considered for the construction of the parameter surfaces inside a particular tile.

4. COMPUTING THE BOUNDARY CURVES FOR PARAMETER SURFACES

The next step is the decomposition of the tiles in the Voronoï diagram (bounded by the outer trimming curve $\mathbf{c}_0$) into parameter surfaces. The

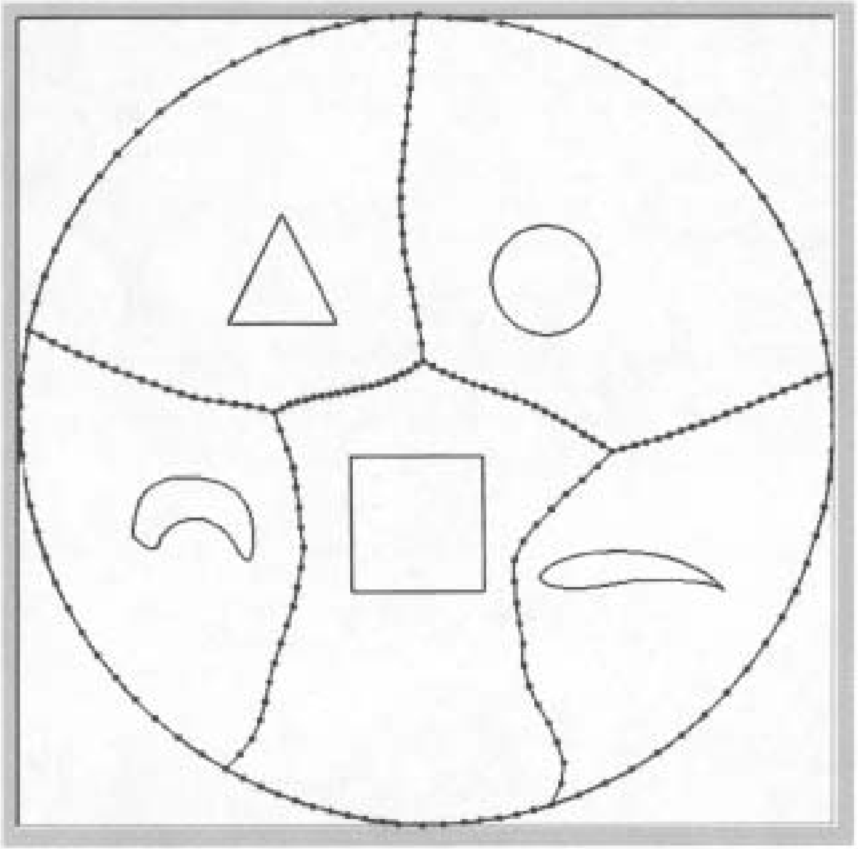

FIG. 5. Cubic B-spline approximation of Voronoï diagram for various trimming curves.

union of all parameter surfaces defines the valid part of the trimmed surface. The algorithm used for the construction of the parameter surfaces is based on decomposing a tile into a set of surfaces that have two horizontal boundary curves. The algorithm is similar to the *scan line* algorithm used for filling the interior of 2D polygons (see [19]).

The basic idea for the construction of the parameter surfaces inside the tile associated with a particular trimming curve $\mathbf{c}$ is as follows. The valid part of the parameter space, i.e., the region between the trimming curve $\mathbf{c}$ and the tile boundary curve $\bar{\mathbf{c}}$, is represented by a set of *ruled parameter surfaces*. They are obtained by identifying local extrema in v-direction (and horizontal line segments) on $\mathbf{c}$ and $\bar{\mathbf{c}}$, computing the intersections between horizontal lines passing through these local extrema (and the horizontal line segments) and $\mathbf{c}$ and $\bar{\mathbf{c}}$, and constructing ruled parameter surfaces from the resulting horizontal line segments inside the tile and non-horizontal curve segments on $\mathbf{c}$ and $\bar{\mathbf{c}}$.

Assuming that the trimming curves are C^0 continuous, the generation of the boundary curves of the parameter surfaces inside the tile associated with trimming curve $\mathbf{c}$ (having tile boundary curve $\bar{\mathbf{c}}$) requires these steps:

(i) Determining extrema in v-direction and end points of horizontal line segments on $\mathbf{c}$ and $\bar{\mathbf{c}}$

(ii) Computing intersection points between horizontal semi-infinite lines (passing through the local extrema on $\mathbf{c}$ and $\bar{\mathbf{c}}$) and $\mathbf{c}$ and $\bar{\mathbf{c}}$; computing intersection points between horizontal semi-infinite lines (defined by the horizontal line segments on $\mathbf{c}$ and $\bar{\mathbf{c}}$) and $\mathbf{c}$ and $\bar{\mathbf{c}}$

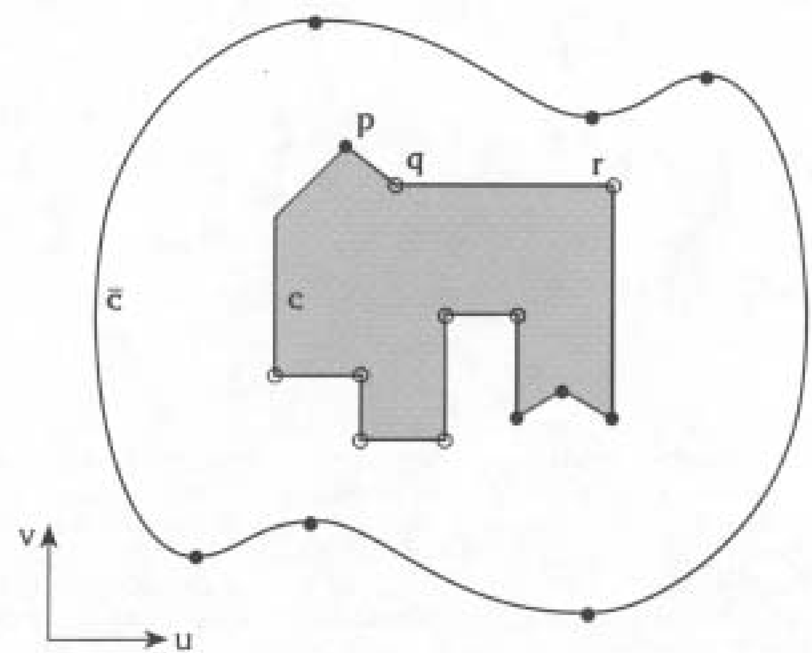

FIG. 6. Local extreme in v-direction and end points of horizontal line segments on $\mathbf{c}$ and $\bar{\mathbf{c}}$.

(iii) Defining boundary curves of parameter surfaces by connecting local extrema, end points of horizontal line segments on $\mathbf{c}$ and $\bar{\mathbf{c}}$, and certain intersection points resulting from (ii)

(iv) Generating ruled parameter surfaces by applying linear interpolation to pairs of non-horizontal curve segments on $\mathbf{c}$ and $\bar{\mathbf{c}}$

The local extrema in v-direction on $\mathbf{c}$ and $\bar{\mathbf{c}}$, computed in step (i), are not always characterized by a horizontal curve tangent. They can coincide with points where tangent discontinuities occur. One must consider all "nearly horizontal" line segments on $\mathbf{c}$ and $\bar{\mathbf{c}}$, i.e., line segments whose absolute slope is smaller than some tolerance. Fig. 6 shows the local extrema $\mathbf{p}$ in v-direction (solid points) and the end points $\mathbf{q}$ and $\mathbf{r}$ of horizontal line segments (circles) on $\mathbf{c}$ and $\bar{\mathbf{c}}$.

When computing the intersections between semi-infinite horizontal lines and $\mathbf{c}$ and $\bar{\mathbf{c}}$ in step (ii), one must consider the existence of "nearly horizontal" curve segments on $\mathbf{c}$ and $\bar{\mathbf{c}}$ that lie inside a "small" strip to both sides of one of the semi-infinite horizontal lines. If such "nearly horizontal"

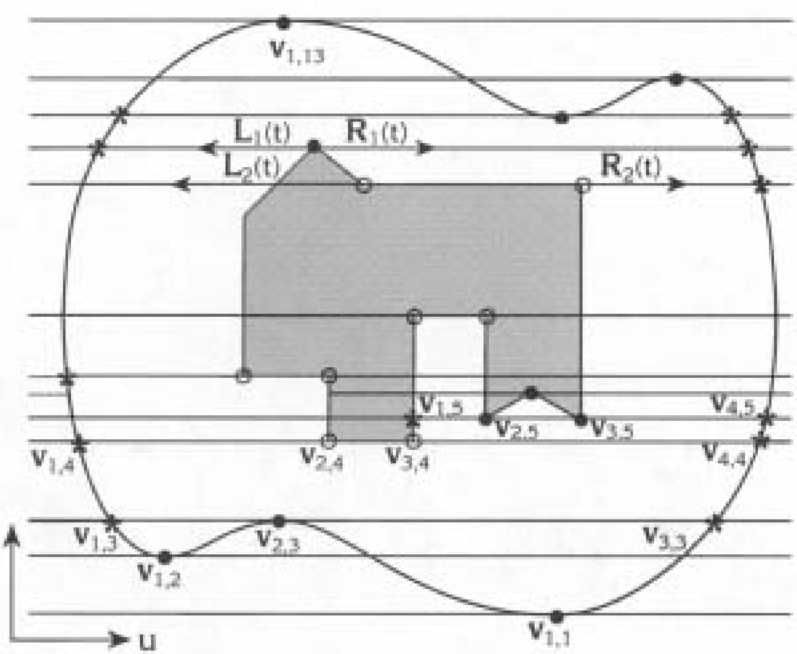

FIG. 7. Required intersection points between semi-infinite horizontal lines and $\mathbf{c}$ and $\bar{\mathbf{c}}$.

curve segments are found, they are not considered for the computation of intersections. Fig. 7 shows the intersection points (stars) that are required for the definition of the parameter surfaces. The semi-infinite lines are denoted by $\mathbf{L}_i(t)$ and $\mathbf{R}_i(t)$, and the corner vertices of the parameter surfaces are denoted by $\mathbf{v}_{j,k}$.

Step (iii) defines a set of curves (line segments and segments of $\mathbf{c}$ and $\bar{\mathbf{c}}$) that define the boundaries of the parameter surfaces. It is necessary to order these curves in both u- and v-direction such that the proper sets of curves are used for the generation of the parameter surfaces. Fig. 8 shows the horizontal line segments $\mathbf{e}_{i,j,k}$ (solid lines) and the non-horizontal curve segments $\mathbf{e}_{q,r,s,t}$ (dashed curves) used as boundary curves.

Since the two non-horizontal boundary curves of each ruled parameter surface are curve segments on $\mathbf{c}$ and $\bar{\mathbf{c}}$, one must represent these curve segments as independent curves, e.g., by a set of B-spline or NURBS curves. If one chooses to use a piecewise polynomial (or piecewise rational) representation, it is necessary to represent the two non-horizontal boundary curves using the same degree and the same knot sequence. This is done by elevating

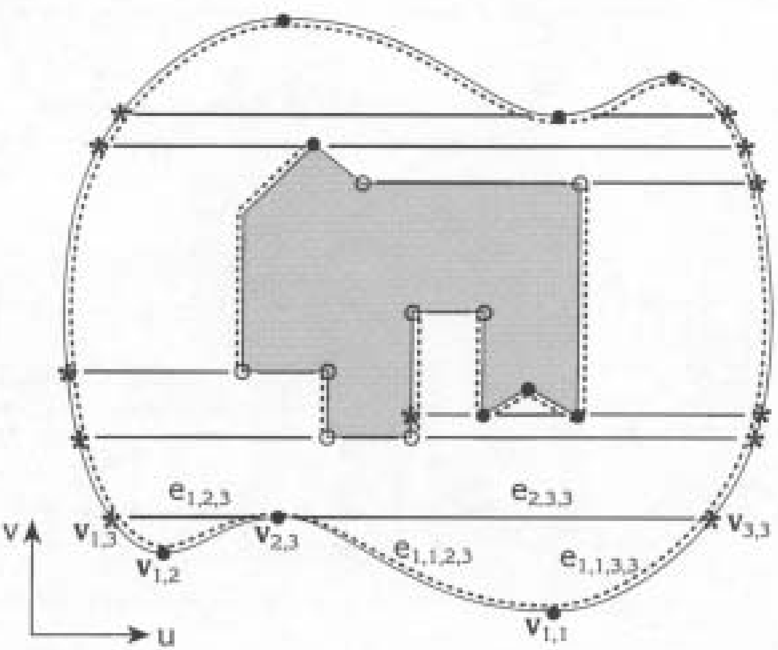

FIG. 8. Horizontal line segments and non-horizontal curve segments defining boundary curves.

the degree of the boundary curve of lower degree to the degree of the boundary curve of higher degree and by "merging" the knot sequences of the two curves (see [19, 20]). Each parameter surface is eventually obtained by performing linear interpolation in u-direction of non-horizontal curve pairs. Fig. 9 shows some of the curvilinear grids obtained when evaluating parameter surfaces $\mathbf{u}_i(\xi, \eta)$ uniformly.

Figs. 10 and 11 show two real-world examples of trimmed surfaces (left: shaded version of trimmed surface; right: parameter space configuration). The trimming curves, the tile boundary curves of the generalized Voronoï diagram, and the resulting curvilinear grids in parameter space are shown.

REMARK 4.1. In the context of grid generation, one must ensure that the same (u, v)-values are used along shared parameter surface boundary curves that are created by the decomposition algorithm in parameter space. Storing the connectivity among the parameter surfaces explicitly allows the generation of grids of arbitrary smoothness across parameter surface boundaries.

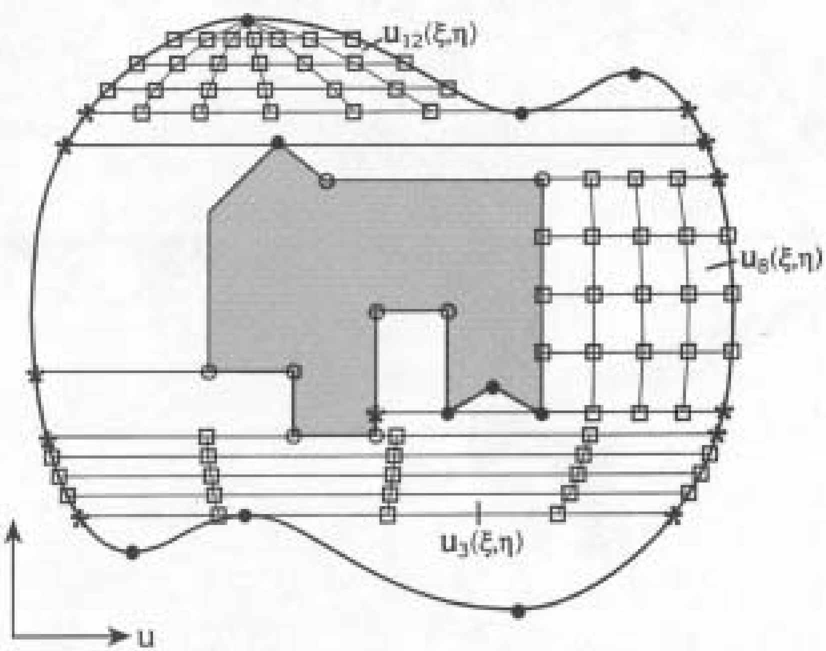

FIG. 9. Curvilinear grids of parameter surfaces resulting from uniform evaluation.

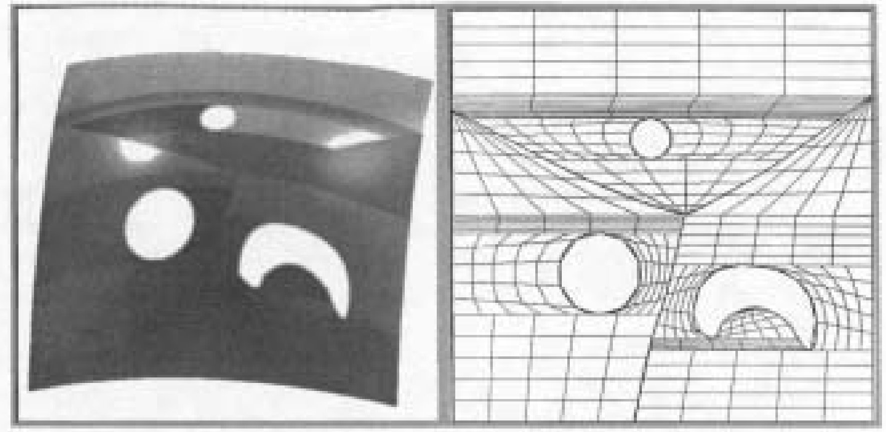

FIG. 10. Trimmed surface with multiple trimming curves.

REMARK 4.2. It takes less than a minute to generate the surface decompositions shown in Figs. 10 and 11 (SGI Indigo[2] workstation).

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,

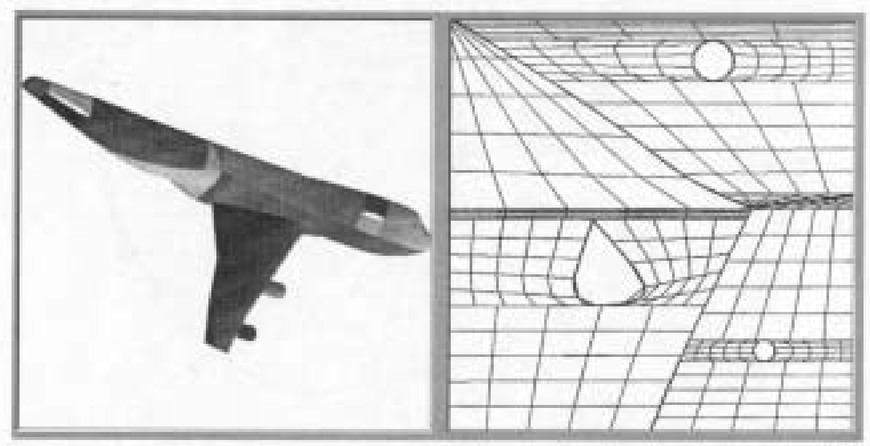

FIG. 11. Fuselage with multiple trimming curves.

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. Two issues must be addressed by further research: (1) Is it possible to modify the algorithm such that no degenerate, i.e., three-sided, patches are created? (2) Can one change the algorithm such that a much smaller number of patches is created?

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 was performed at the NSF Engineering Research Center for Computational Field Simulation, Mississippi State University.

REFERENCES

1 G. Farin, *Curves and Surfaces for Computer Aided Geometric Design*, fourth edition, Academic Press, San Diego, CA, 1997.

2 J. Hoschek and D. Lasser, *Fundamentals of Computer Aided Geometric Design*, A K Peters, Ltd., Wellesley, MA, 1993.

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

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

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

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

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

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

9 D. Leven and M. Sharir, 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, (1986).

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

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

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

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

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

15 A. E. Vries-Baayens and C. H. Seebregts, 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, (1992).

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

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

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

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

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

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

20 L. A. Piegl, 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, (1991b).