UC Davis
IDAV Publications

Title
Using Isosurface Methods for Visualizing the Envelope of a Swept Trivariate Solid

Permalink
https://escholarship.org/uc/item/6kd702tq

Authors
Conkey, Jason
Joy, Ken

Publication Date
2000

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

Using Isosurface Methods for Visualizing the Envelope of a Swept Trivariate Solid

Jason Conkey[*]
Kenneth I. Joy[†]

Center for Image Processing and Integrated Computing
Department of Computer Science
University of California, Davis

Abstract

We present a method for calculating the envelope surface of a parametric solid object swept along a path in three-dimensional space. The boundary surface of the solid is the combination of parametric surfaces and an implicit surface where the Jacobian of the defining function has a rank-deficiency condition. Using this condition, we determine a set of square sub-Jacobian determinants that must all vanish simultaneously on the implicit surface. When the generator of the swept surface is a trivariate tensor-product B-spline solid and the path is a B-spline curve, we can give a robust algorithm to determine the implicit surface. This algorithm is based upon the "marching tetrahedra" method, which is adapted to work on 4-simplices. The envelope of the swept solid is given by the union of the parametric and implicit surfaces.

Keywords: swept surface; envelopes; boundary surface determination; trivariate B-spline solids; rank-deficient Jacobians; marching tetrahedra.

1 Introduction

The definition of a swept object depends on three factors: the specification of the generator – the object to be swept; the specification of the trajectory – the sweeping path; and the specification of the orientation of the generator as it progresses along the trajectory. Thus, given a generator $\mathbf{g}(u,v,w)$, a trajectory curve $\mathbf{c}(t)$, and a *coordinate frame transformation* $R(t)$, defined over the same domain as $\mathbf{c}$, the swept object $\mathbf{s}$ is defined to be the set of points where

$$\mathbf{s}(u,v,w,t) = \mathbf{c}(t) + R(t)\mathbf{g}(u,v,w) \qquad (1)$$

for some t in the domain of the curve $\mathbf{c}$ and some (u,v,w) in the domain of $\mathbf{g}$. The swept objects defined by this definition are actually quite general. They allow an arbitrary parametric solid to be swept along a parametric curve, utilizing an arbitrary coordinate frame transformation to define the orientation/distortion of the solid at a point of the curve. This allows, for example, the generator to be scaled, rotated or distorted (the coordinate frame elements need not be mutually perpendicular) as it proceeds along the curve.

In this paper, we are interested in the boundary surfaces (the envelope) of $\mathbf{s}$. These surfaces consist of two types: the envelopes of swept surfaces that correspond to boundaries of the domain space, and an isosurface determined by the Jacobian of $\mathbf{s}$. We utilize the techniques of Joy [5], Joy and Duchaineau [6] and Abdel-Malek and Yeh [2] to calculate these surfaces. The Jacobian of $\mathbf{s}$ is not square (it is 4×3) and the methodology of Abdel-Malek and Yeh provides means to break up the Jacobian, giving sub-Jacobians that can be used in our calculations. The techniques of Joy and Duchaineau provide the surface generation techniques for the swept surfaces that define the parametric boundaries. If the generators are defined as trivariate tensor-product B-spline solids, the trajectories are defined to be B-spline curves, and the coordinate frames are specified by B-splines, a robust test can be developed that indicates the presence of an implicit boundary surface in a region of the domain space. An adaptive subdivision procedure is generated which splits regions of the domain into four-dimensional cells that (1) do not contain the solid, or (2) that may contain the solid.

After determining a set of cells in the domain space that may contain the implicit surface, we use an isosurface generation scheme to generate the implicit surface. Here we split the four-dimensional cells into 4-simplices and adapt a "marching tetrahedra" algorithm to generate the isosurface. The union of the envelopes of the surfaces swept from the boundary of the domain, together with the isosurface gener-

[*] conkey@cs.ucdavis.edu
[†] joy@cs.ucdavis.edu

ated from the sub-Jacobians then approximate the boundary surface of the solid.

In Section 2, we review research related to the sweeping of parametric solid models, and the calculation of the boundary surfaces of trivariate solids. Section 3 reviews the basic mathematical principles necessary for the generation of the swept solid. Section 4 develops the defining function for the swept solid, and Section 5 develops the algorithm for finding the implicit surface. The isosurface generation algorithm is presented in Section 6. Results of the use of the algorithm are shown in Section 7.

2 Related Work

The research on swept surfaces and solids is quite vast and we refer the interested reader to the review by Abdel-Malek *et al.* [1] for a good review of the results in this area over the past few years.

Boltianskii [4] has an excellent introduction to envelope theory. Wang and Wang [13] utilized Boltyanskii's work to find a geometric definition of the envelope. They find that the envelope of a generator $\mathbf{g}(u, v, w)$ along a curve $\mathbf{c}(t)$ can be characterized at a point t_0 by boundary points (u_0, v_0, w_0) of the generator where the normal vector to the generator at (u_0, v_0, w_0) is perpendicular to the tangent vector of $\mathbf{c}$ at t_0.

Joy [5] discusses the sweeping of a trivariate solid, and uses a subdivision strategy to generate the image of the solid. This algorithm subdivides the trajectory curve until a sweep of the domain boundaries accurately represents the solid. Unfortunately, this algorithm must subdivide the object into very small pieces and renders many surfaces within the solid.

Madrigal and Joy [8] have proposed an algorithm that generates characteristic curves on the envelope for each step along the trajectory curve. Points on these characteristic curves have the property that the normal vectors to the points are perpendicular to the tangent vectors at the respective points on the trajectory curve. A "zippering" algorithm generates triangle strips from characteristic curves of two consecutive steps along the trajectory. For each step, the generator is placed at its position, and the curves calculated. Unfortunately, the zippering algorithm fails when complex objects are rotated while being swept. These rotations can create complex characteristic curves on the envelopes.

Abdel-Malek and Yeh [2] and Abdel-Malek *et al.* [3], have developed a rank-deficiency condition that characterizes the Jacobians of swept solids in any dimension. If a swept solid is characterized by n-dimensional equations, then the Jacobian is a $n \times 3$ matrix. They have shown that the implicit surface is defined when all the 3×3 sub-Jacobians are simultaneously zero. The paper gives examples on how to find these surfaces by solving the sub-Jacobians explicitly.

Joy and Duchaineau [6] have developed algorithms for rendering of the trivariate B-spline solid. This solid can be visualized as the sweep of a continuously changing B-spline patch along a B-spline curve. They show that the boundary surface of such a solid consists of two components: surfaces that are images of the domain boundary, and an implicit surface defined as the isosurface of the Jacobian determinant with isovalue zero. They utilize cone approximations to give a robust test to see if the implicit surface exists in a domain cell, and employ a subdivision algorithm to focus the attention of the algorithm on smaller and smaller cells. An adaptive isosurface routine is used to generate the implicit surface.

In this paper, we combine the results of Joy and Duchaineau [6] with Abdel-Malek and Yeh [2] to develop an isosurface-based method for sweeping a trivariate B-spline solid along a B-spline curve. We provide a robust method that guarantees the presence of the implicit boundary and renders it to a desired accuracy. To do this, we approximate the Jacobian sub-determinants using interval techniques, and subdivide the domain space to isolate rectilinear cells in the domain that contain the implicit surface. To determine the isosurface within these cells, we present a new isosurface algorithm for four-simplices. We split cells into 4-simplices and determine the isosurface within each simplex, combining the results to produce the implicit representation of the surface.

3 Calculating the Envelope

The mathematics of swept objects is rooted in the Implicit Function Theorem [11]. This theorem implies that surfaces contributing to the envelope that contains the swept solid include the boundary surfaces of solids defined by the boundaries of the domain space, and those points where the Jacobian has a rank-deficiency condition.

If we assume that the domain space for our solids is a 4-dimensional rectangle defined by $0 \leq u, v, w, t \leq 1$, then by the Implicit Function Theorem, the superset of the surfaces that make up the boundary of $\mathbf{s}$ includes the following:

- The boundary surfaces of the solid $\mathbf{s}(u, v, w, 0)$ corresponding to the domain boundary $t = 0$. This is a copy of the generator, translated to $\mathbf{c}(0)$ and transformed by $R(0)$.

- The boundary surfaces of the solid $\mathbf{s}(u, v, w, 1)$ corresponding to the domain boundary $t = 1$. This is a copy of the generator, translated to $\mathbf{c}(1)$ and transformed by $R(1)$.

- The boundary surfaces of the solid $\mathbf{s}(0, v, w, t)$ corresponding to the domain boundary $u = 0$. This solid is a boundary patch $\mathbf{g}(0, v, w)$ of the generator, swept along the trajectory curve $\mathbf{c}(t)$.

- The boundary surfaces of the solid $\mathbf{s}(1, v, w, t)$ corresponding to the domain boundary $u = 1$. This solid is a boundary patch $\mathbf{g}(1, v, w)$ of the generator, swept along the trajectory curve $\mathbf{c}(t)$.
- The boundary surfaces of the solid $\mathbf{s}(u, 0, w, t)$ corresponding to the domain boundary $v = 0$. This solid is a boundary patch $\mathbf{g}(u, 0, w)$ of the generator, swept along the trajectory curve $\mathbf{c}(t)$.
- The boundary surfaces of the solid $\mathbf{s}(u, 1, w, t)$ corresponding to the domain boundary $v = 1$. This solid is a boundary patch $\mathbf{g}(u, 1, w)$ of the generator, swept along the trajectory curve $\mathbf{c}(t)$.
- The boundary surfaces of the solid $\mathbf{s}(u, v, 0, t)$ corresponding to the domain boundary $w = 0$. This solid is a boundary patch $\mathbf{g}(u, v, 0)$ of the generator, swept along the trajectory curve $\mathbf{c}(t)$.
- The boundary surfaces of the solid $\mathbf{s}(u, v, 1, t)$ corresponding to the domain boundary $w = 1$. This solid is a boundary patch $\mathbf{g}(u, v, 1)$ of the generator, swept along the trajectory curve $\mathbf{c}(t)$.
- The surface defined where the rank of the "Jacobian" of $\mathbf{s}$ is less than or equal to two.

In our case, the Jacobian is a 4×3 matrix defined by

$$\begin{pmatrix} \frac{\partial x}{\partial u} & \frac{\partial y}{\partial u} & \frac{\partial z}{\partial u} \\ \frac{\partial x}{\partial v} & \frac{\partial y}{\partial v} & \frac{\partial z}{\partial v} \\ \frac{\partial x}{\partial w} & \frac{\partial y}{\partial w} & \frac{\partial z}{\partial w} \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} & \frac{\partial z}{\partial t} \end{pmatrix}$$

Abdel-Malek and Yeh [2] have shown that the rank-Jacobian condition is equivalent to the vanishing of the determinants of the four possible 3×3 sub-Jacobians of J. These sub-Jacobians, J_{123}, J_{124}, J_{134}, and J_{234} are defined as follows:

$$J_{123} = \begin{vmatrix} \frac{\partial x}{\partial u} & \frac{\partial y}{\partial u} & \frac{\partial z}{\partial u} \\ \frac{\partial x}{\partial v} & \frac{\partial y}{\partial v} & \frac{\partial z}{\partial v} \\ \frac{\partial x}{\partial w} & \frac{\partial y}{\partial w} & \frac{\partial z}{\partial w} \end{vmatrix}$$

$$J_{124} = \begin{vmatrix} \frac{\partial x}{\partial u} & \frac{\partial y}{\partial u} & \frac{\partial z}{\partial u} \\ \frac{\partial x}{\partial v} & \frac{\partial y}{\partial v} & \frac{\partial z}{\partial v} \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} & \frac{\partial z}{\partial t} \end{vmatrix}$$

$$J_{134} = \begin{vmatrix} \frac{\partial x}{\partial u} & \frac{\partial y}{\partial u} & \frac{\partial z}{\partial u} \\ \frac{\partial x}{\partial w} & \frac{\partial y}{\partial w} & \frac{\partial z}{\partial w} \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} & \frac{\partial z}{\partial t} \end{vmatrix}$$

$$J_{234} = \begin{vmatrix} \frac{\partial x}{\partial v} & \frac{\partial y}{\partial v} & \frac{\partial z}{\partial v} \\ \frac{\partial x}{\partial w} & \frac{\partial y}{\partial w} & \frac{\partial z}{\partial w} \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} & \frac{\partial z}{\partial t} \end{vmatrix}$$

A point $\mathbf{s}(u, v, w, t)$ is on the surface of the envelope, if

$$J_{123} = J_{124} = J_{134} = J_{234} = 0$$

at (u, v, w, t).

Using equation (1), we can expand the partial derivatives as follows:

$$\begin{aligned} \frac{\partial \mathbf{s}}{\partial u} &= R(t) \frac{\partial \mathbf{g}}{\partial u}(u, v, w) \\ \frac{\partial \mathbf{s}}{\partial v} &= R(t) \frac{\partial \mathbf{g}}{\partial v}(u, v, w) \\ \frac{\partial \mathbf{s}}{\partial w} &= R(t) \frac{\partial \mathbf{g}}{\partial w}(u, v, w) \\ \frac{\partial \mathbf{s}}{\partial t} &= \frac{d\mathbf{c}}{dt}(t) + \frac{dR}{dt}(t)\mathbf{g}(u, v, w) \end{aligned}$$

and we have that

$$\begin{aligned} J_{123}(u, v, w, t) &= \begin{vmatrix} \frac{\partial x}{\partial u} & \frac{\partial y}{\partial u} & \frac{\partial z}{\partial u} \\ \frac{\partial x}{\partial v} & \frac{\partial y}{\partial v} & \frac{\partial z}{\partial v} \\ \frac{\partial x}{\partial w} & \frac{\partial y}{\partial w} & \frac{\partial z}{\partial w} \end{vmatrix} \\ &= |R(t)| \begin{vmatrix} \frac{\partial \mathbf{g}_x}{\partial u} & \frac{\partial \mathbf{g}_y}{\partial u} & \frac{\partial \mathbf{g}_z}{\partial u} \\ \frac{\partial \mathbf{g}_x}{\partial v} & \frac{\partial \mathbf{g}_y}{\partial v} & \frac{\partial \mathbf{g}_z}{\partial v} \\ \frac{\partial \mathbf{g}_x}{\partial w} & \frac{\partial \mathbf{g}_y}{\partial w} & \frac{\partial \mathbf{g}_z}{\partial w} \end{vmatrix} \end{aligned}$$

This is the product of the determinant of $R(t)$ and the Jacobian determinant of $\mathbf{g}$. If we assume that the determinant of $R(t)$ does not vanish (*i.e.*, no degeneracies exist in the coordinate frame transformation), then J_{123} is non-zero if and only if the Jacobian determinant of $\mathbf{g}$ is non-zero. We can use this to show the following theorem.

Theorem 3.1
If there are no points on the generator $\mathbf{g}$ where $J_{123} = 0$, then the boundary of the swept solid contains points only from the parametric boundaries of the domain space.

For $\mathbf{s}(u, v, w, t)$, we can characterize the points on the boundary of the swept solid by defining the Jacobian determinants in terms of triple scalar products. We have

$$\begin{aligned} J_{123} &= |R(t)| \left(\frac{\partial \mathbf{g}}{\partial u} \right) \cdot \left[\left(\frac{\partial \mathbf{g}}{\partial v} \times \frac{\partial \mathbf{g}}{\partial w} \right) \right] \\ J_{124} &= -\left(\frac{d\mathbf{c}}{dt}(t) + \frac{dR}{dt}(t)\mathbf{g}(u, v, w) \right) \cdot \left[R(t) \left(\frac{\partial \mathbf{g}}{\partial u} \times \frac{\partial \mathbf{g}}{\partial v} \right) \right] \\ J_{134} &= \left(\frac{d\mathbf{c}}{dt}(t) + \frac{dR}{dt}(t)\mathbf{g}(u, v, w) \right) \cdot \left[R(t) \left(\frac{\partial \mathbf{g}}{\partial u} \times \frac{\partial \mathbf{g}}{\partial w} \right) \right] \\ J_{234} &= -\left(\frac{d\mathbf{c}}{dt}(t) + \frac{dR}{dt}(t)\mathbf{g}(u, v, w) \right) \cdot \left[R(t) \left(\frac{\partial \mathbf{g}}{\partial v} \times \frac{\partial \mathbf{g}}{\partial w} \right) \right] \end{aligned}$$

These four equations imply that points on the implicit surface must satisfy the following:

- the vectors $\frac{\partial \mathbf{g}}{\partial u}$, $\frac{\partial \mathbf{g}}{\partial v}$, and $\frac{\partial \mathbf{g}}{\partial w}$ must be coplanar, and

- the vectors

$$\left[R(t)\left(\frac{\partial \mathbf{g}}{\partial u}\times\frac{\partial \mathbf{g}}{\partial v}\right)\right],$$
$$\left[R(t)\left(\frac{\partial \mathbf{g}}{\partial u}\times\frac{\partial \mathbf{g}}{\partial w}\right)\right], \text{and}$$
$$\left[R(t)\left(\frac{\partial \mathbf{g}}{\partial v}\times\frac{\partial \mathbf{g}}{\partial w}\right)\right]$$

  must be perpendicular to the vector

$$\left(\frac{d\mathbf{c}}{dt}(t)+\frac{dR}{dt}(t)\mathbf{g}(u,v,w)\right)$$

4 Applying the Method to Trivariate B-Spline Solids

If all functional components of $\mathbf{s}$ are given in B-spline form, then the resulting sweep can also be put in B-spline form, as follows: Let $\mathbf{g}$ be the trivariate B-spline generator

$$\mathbf{g}(u,v,w)=\sum_{i=0}^{n_1}\sum_{j=0}^{n_2}\sum_{k=0}^{n_3}\mathbf{g}_{i,j,k}N_{i,m}(u)N_{j,m}(v)N_{k,m}(w)$$

and let $\mathbf{c}$ be the B-spline curve trajectory

$$\mathbf{c}(t)=\sum_{l=0}^{n_4}\mathbf{c}_l N_{l,m}(t)$$

with a coordinate frame transformation $R(t) = (\mathbf{b}(t),\mathbf{n}(t),\mathbf{t}(t))$ given by

$$\mathbf{b}(t)=\sum_{l=0}^{n_4}\mathbf{b}_l N_{l,m}(t)$$
$$\mathbf{n}(t)=\sum_{l=0}^{n_4}\mathbf{n}_l N_{l,m}(t)$$
$$\mathbf{t}(t)=\sum_{l=0}^{n_4}\mathbf{t}_l N_{l,m}(t)$$

where the frame functions are defined over the same knot sequence as $\mathbf{c}$. The swept solid $\mathbf{s}(u,v,w,t)$ can be written as

$$\begin{aligned}
\mathbf{s} &= \mathbf{c}(t)+x_{\mathbf{g}(u,v,w)}\mathbf{b}(t)+y_{\mathbf{g}(u,v,w)}\mathbf{n}(t)+z_{\mathbf{g}(u,v,w)}\mathbf{t}(t)\\
&=\sum_{l=0}^{n_4}\left[\mathbf{c}_l+x_{\mathbf{g}(u,v,w)}\mathbf{b}_l+y_{\mathbf{g}(u,v,w)}\mathbf{n}_l+z_{\mathbf{g}(u,v,w)}\mathbf{t}_l\right]N_{l,m}(t)\\
&=\sum_{l=0}^{n_4}\left[\mathbf{c}_l+\left(\sum_{i=0}^{n_1}\sum_{j=0}^{n_2}\sum_{k=0}^{n_3}x_{\mathbf{g}_{i,j,k}}N_{i,m}(u)N_{j,m}(v)N_{k,m}(w)\right)\mathbf{b}_l\right.\\
&\quad+\left(\sum_{i=0}^{n_1}\sum_{j=0}^{n_2}\sum_{k=0}^{n_3}y_{\mathbf{g}_{i,j,k}}N_{i,m}(u)N_{j,m}(v)N_{k,m}(w)\right)\mathbf{n}_l\\
&\quad\left.+\left(\sum_{i=0}^{n_1}\sum_{j=0}^{n_2}\sum_{k=0}^{n_3}z_{\mathbf{g}_{i,j,k}}N_{i,m}(u)N_{j,m}(v)N_{k,m}(w)\right)\mathbf{t}_l\right]N_{l,m}(t)\\
&=\sum_{l=0}^{n_4}\sum_{i=0}^{n_1}\sum_{j=0}^{n_2}\sum_{k=0}^{n_3}\left[\mathbf{c}_l+x_{\mathbf{g}_{i,j,k}}\mathbf{b}_l+y_{\mathbf{g}_{i,j,k}}\mathbf{n}_l+z_{\mathbf{g}_{i,j,k}}\mathbf{t}_l\right]\\
&\qquad N_{i,m}(u)N_{j,m}(v)N_{k,m}(w)N_{l,m}(t)
\end{aligned}$$

which is a 4-dimensional B-spline with control points

$$\mathbf{c}_l+x_{\mathbf{g}_{i,j,k}}\mathbf{b}_l+y_{\mathbf{g}_{i,j,k}}\mathbf{n}_l+z_{\mathbf{g}_{i,j,k}}\mathbf{t}_l$$

for $0\le l\le n_4$, $0\le k\le n_3$, $0\le j\le n_2$, and $0\le i\le n_1$.

In this case, the eight parametric solids defining the boundaries of the sweep are all trivariate solids and can be rendered using the algorithm of [6]. To calculate the implicit surface we can utilize subdivision to locate those domain rectangles where the implicit surface exists.

5 The Algorithm

Assuming that $\mathbf{s}(u,v,w,t)$ is given in B-spline form, we can use an adaptive subdivision algorithm to generate the implicit boundary surface of $\mathbf{s}$ where

$$J_{123}=J_{124}=J_{134}=J_{234}=0.$$

Assuming that the domain space is defined by a four-dimensional rectangle where $0\le u,v,w,t\le 1$, the algorithm proceeds as follows:

- Using subdivision on u, v and w, find a set of cells where $J_{123}=0$. This can be done by analyzing the trivariate generator $\mathbf{g}$, see [6].

- Using subdivision on t, refine the remaining cells, removing those cells where the sub-Jacobians cannot be simultaneously zero.

- Split each four-dimensional cell into 24 4-simplices using CMK-splits, see [9, 10].

- For each 4-simplex, determine a set of tetrahedra that approximates the points where $J_{123}=0$, see Section 6.

- For each tetrahedron, determine the isosurface corresponding to $J_{124} = J_{134} = J_{234} = 0$.

The triangles generated through this process define the implicit surface bounding the swept solid $\mathbf{s}$.

Following [6], we use cone approximations to the partial derivative vectors to get bounds on the individual Jacobian determinants. The strategy is to use subdivision and continually refine the approximations, throwing out those cells where the Jacobian determinants cannot be simultaneously zero. In this way, we obtain a set of small cells in which the envelope surface lies and can use the isosurface methods of the following section to determine the surface.

6 Isosurface Determination in 4-Simplices

The marching tetrahedra algorithm [14, 12, 10] is a simplification of the marching cubes algorithm [7] for finding isosurfaces in gridded data sets. Assuming a scalar field over a tetrahedral mesh and an isovalue v for which we must determine an isosurface, each tetrahedron can be classified into three cases:

- A tetrahedron whose vertices have associated scalar values that are all greater than, or less than v. In this case, we say that no isosurface is present.
- A tetrahedron having one vertex with scalar value less than v and three vertices with scalar values greater than v. In this case, we approximate the isosurface with a triangle, using linear interpolation to define the points on the edges. (The case where three vertices have scalar values greater than v, and one vertex with value less than v is handled similarly.) This case is shown in Figure 1a.
- A tetrahedron having two vertices with scalar value greater than v and two vertices with values less than v. In this case, the surface is represented by a quadrilateral, using linear interpolation to define the points on the edges. This quadrilateral is usually triangulated by using one of the possible two triangulations. This case is shown in Figure 1b.

Consider a 4-simplex T, and let $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$ and $\mathbf{p}_5$ be the five vertices of T. T has five 3-dimensional tetrahedra as boundaries:

$$\begin{aligned}
\partial T_1 &: \text{ with vertices } \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \text{ and } \mathbf{p}_4 \\
\partial T_2 &: \text{ with vertices } \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \text{ and } \mathbf{p}_5 \\
\partial T_3 &: \text{ with vertices } \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_4, \text{ and } \mathbf{p}_5 \\
\partial T_4 &: \text{ with vertices } \mathbf{p}_1, \mathbf{p}_3, \mathbf{p}_4, \text{ and } \mathbf{p}_5 \\
\partial T_5 &: \text{ with vertices } \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \text{ and } \mathbf{p}_5
\end{aligned}$$

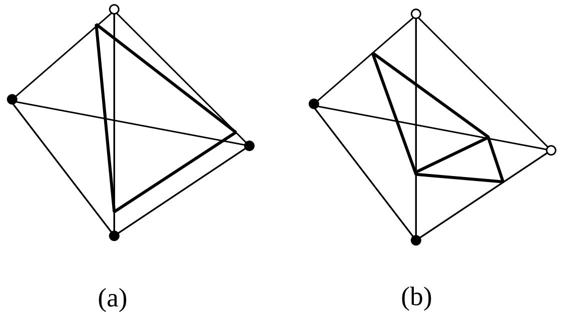


Figure 1. Determining isosurfaces of a scalar field over tetrahedra. This illustration shows the two non-trivial cases. In (a), we approximate the isosurface with a single triangle. In (b), we approximate the isosurface with a quadrilateral.

An isosurface of a scalar field defined over the vertices of T will be three-dimensional, and can be approximated by sets of tetrahedra in four-dimensions. Without loss of generality, we assume that the desired isosurface has isovalue zero. There are two cases to consider:

CASE I: Suppose, without loss of generality, that $\mathbf{p}_5$ has a scalar value greater than zero and $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ and $\mathbf{p}_4$ have isovalues less than zero. Then we have the case shown in Figure 2a. Here, the "isosurface" can be approximated by a single tetrahedron. The vertices of the tetrahedron are calculated by linear interpolation on the edges of the 4-simplex.

CASE II: Suppose that $\mathbf{p}_4$ and $\mathbf{p}_5$ have scalar values greater than zero, and that $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ have scalar values less than zero. The five boundary tetrahedra of the 4-simplex T can be separated into two classes: ∂T_1 and ∂T_2 each have three vertices with negative scalar values and one with a positive value; ∂T_3, ∂T_4, and ∂T_5 each have two vertices with negative scalar values and two vertices with positive values. In the first case, the isosurface on each boundary tetrahedron can be approximated by a single triangle. In the second case, the isosurface on each boundary tetrahedron can be approximated by a quadrilateral. The three quadrilaterals and two triangles form a three-dimensional prism in the 4-simplex T, as is shown in Figure 2b.

The prism can be split into three tetrahedra. There are six ways to do this, each corresponding to the specification of the diagonals of the quadrilateral faces of the prism, see Figure 3. Since we have a free choice of the six possible splits, we choose to split the prism into the three tetrahedra that correspond to the minimum length (in four-dimensions) diagonal of each quadrilateral face. If we choose a regular split of the domain space, where all cells are the same size,

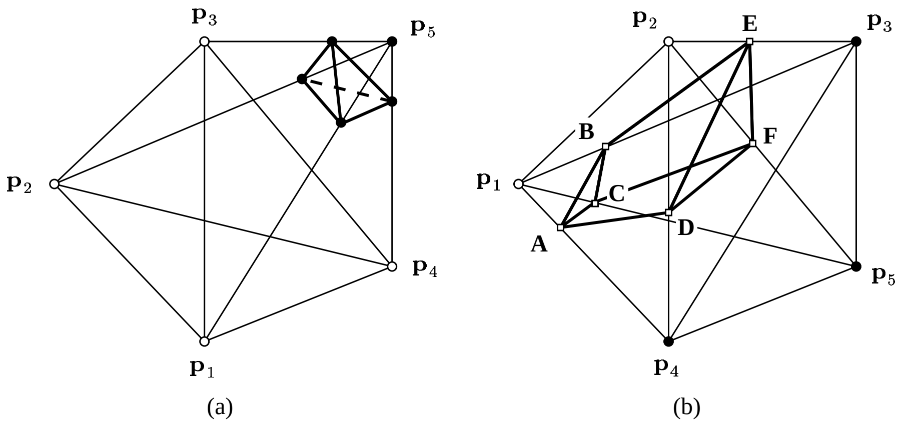


Figure 2. Two cases for approximating the isosurface in 4-simplices. In (a), one tetrahedron is formed to approximate the surface. In (b), a four-dimensional prism is formed.

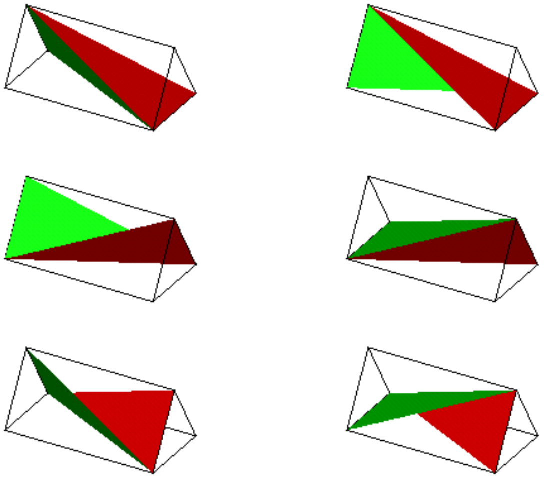


Figure 3. Six ways to split a prism into tetrahedra

this eliminates any "cracking" problems in the isosurface, because adjacent prisms will have common face diagonals chosen in the same manner.

7 Results

Figure 4 illustrates a torus that has been swept along a semi-circular arc. Since the torus contains no surfaces where the Jacobian determinant of the defining function is zero, the boundary surfaces of the swept object are contained completely in the surfaces of the swept trivariate solids that are defined by the boundary of the domain space. The generator is composed of four trivariate solids, each consisting of one-quarter of the torus. The semi-circular arc is approximated by B-splines. The torus is rotated by approximating the Frenet frame of the curve with B-splines. Figure 5 illustrates a cylinder that has been tumbled $180°$ as it is swept along a linear path. The cylinder has been approximated by a trivariate B-spline, and the coordinate frame vectors have been approximated by B-spline curves.

Figure 6 illustrates a trivariate generator swept along a curve. Here, the generator is not rotated. The protrusion from the cube in the generator is created by pulling out the interior control points of the trivariate B-spline. On the surface of this protrusion, $J_{123} = 0$. When swept, this protrusion tests the full capabilities of our algorithm.

In Figure 7, the generator is a trivariate approximation of a cylinder, with some interior control points pulled to the exterior of the object. The generator is rotated about the linear trajectory as it is swept along the curve.

Figure 8 illustrates a swept solid using the generator of Figure 6. The generator is swept along a semi-circular curve and rotated.

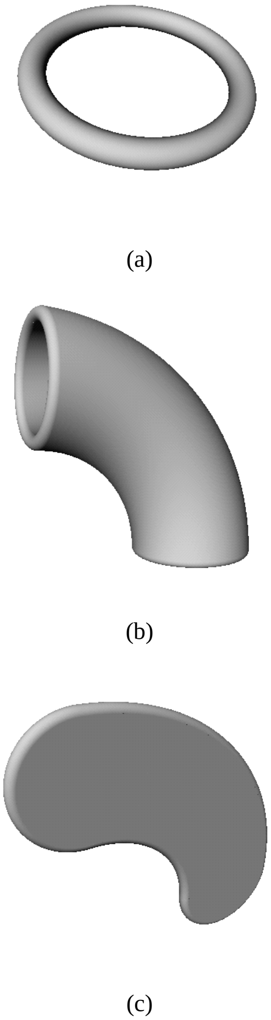


Figure 4. Sweeping a toroid along a semicircular arc. The generator is shown in (a). In (b), the generator has been swept along the arc and the boundary surfaces are shown. In (c), the generator was first rotated into the plane of the arc, and then swept.

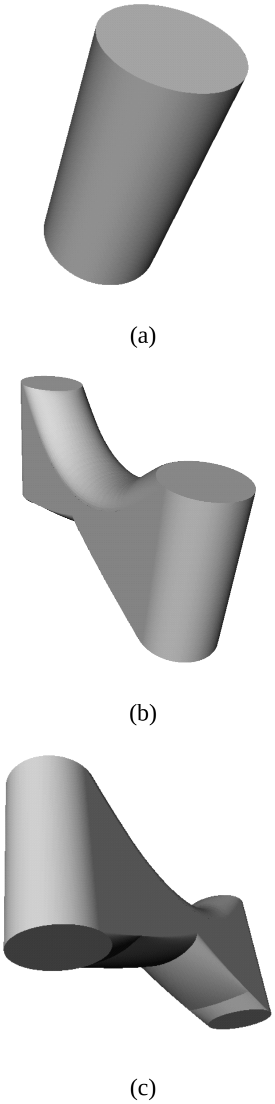
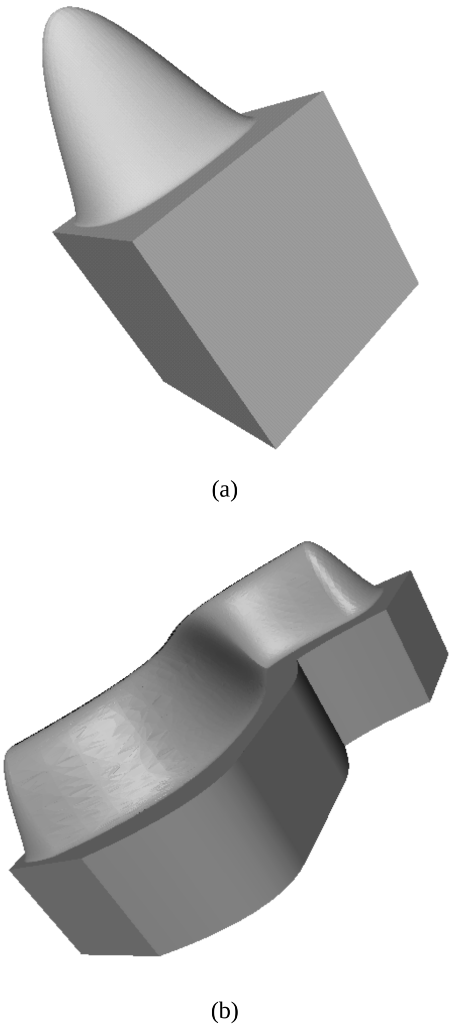


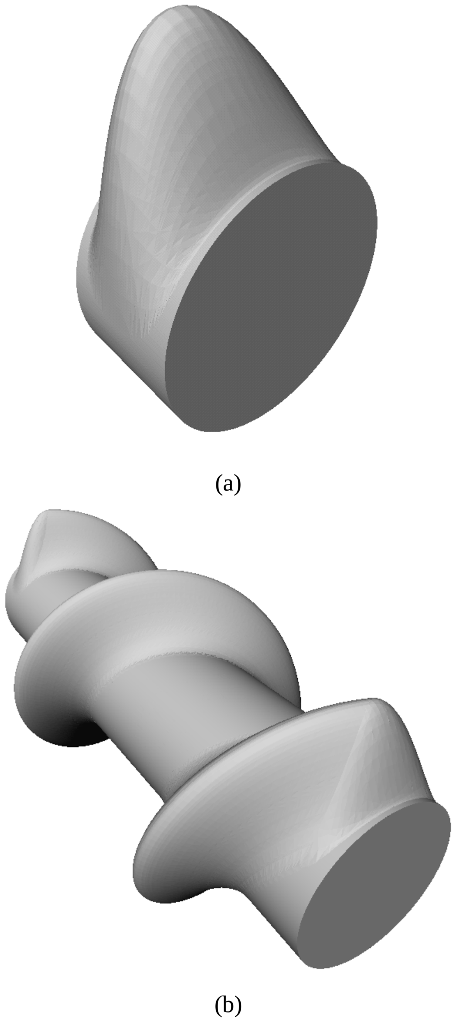
Figure 5. Sweeping a cylinder along a linear path. The cylinder is rotated as it is swept. The generator is shown in (a) and two illustrations of the swept object are shown in (b) and (c).



(a)

(b)

Figure 6. Sweeping along a general B-spline curve. The generator is shown in (a), and the swept surface is shown in (b).



(a)

(b)

Figure 7. Sweeping along a linear curve with rotation. The generator is shown in (a), and the swept surface is shown in (b).

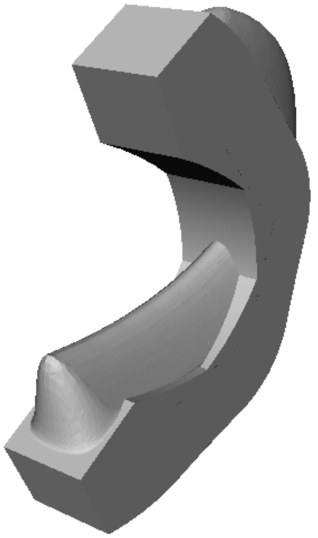


Figure 8. Sweeping the generator of Figure 6 along a semi-circular arc. The generator is tumbled as it is swept.

8 Conclusions

The concept of determining the envelope of a solid object swept along a curve has been a fundamental problem in geometric modeling, solid modeling, robotics, manufacturing automation and computer graphics since the early 1960s. The goal is to determine the outer boundary of an object during its motion along the curve from an initial to a final position. While the mathematics has become more sophisticated, the difficulty remains the identification of this boundary using rigorous mathematical techniques and robust methods that lead to implementable algorithms on computer systems.

We have presented an algorithm that generates the boundary surfaces of a swept trivariate solid. This algorithm generates the boundary through a combination of parametric and implicit surfaces. Generation of the implicit surface is accomplished through an isosurface routine that has been adapted to 4-simplices in four-dimensional space. We have found that this algorithm can be used to generate a variety of swept solids. The algorithm depends on the B-spline representation only to calculate the bounds on each cell and to use the algorithm presented in [6] for the trivariate B-spline solids generated by the method. It can be extended to other surface types if similar approximation techniques can be developed.

This method generates a superset of the actual boundary of the swept solid, generating some surfaces on the interior of the solid. In future work, we will identify these interior points and eliminate them from the description of the boundary.

References

[1] K. Abdel-Malek, D. Blackmore, and K. I. Joy. Swept volumes: Foundations, perspectives and applications, 2000 (submitted for publication).

[2] K. Abdel-Malek and H. Yeh. Geometric representation of the swept volume using jacobian rank-deficiency conditions. *Computer-aided Design*, 29(6):457–468, 1997.

[3] K. Abdel-Malek, H. Yeh, and S. Othman. Swept volumes: Void and boundary identification. *Computer-aided Design*, 30(8):1009–1018, 1999.

[4] V. G. Boltianskii. *Envelopes*. Macmillan, New York, 1964.

[5] K. I. Joy. Visualization of swept hyperpatch solids. In T. L. Kunii, editor, *Visual Computing : Integrating Computer Graphics with Computer Vision (Proceedings of Computer Graphics International 92)*, pages 567–582. Springer-Verlag, Tokyo, June 1992.

[6] K. I. Joy and M. A. Duchaineau. Boundary determination for trivariate solids. In *Proceedings of Pacific Graphics '99*, pages 82–91, Oct. 1999.

[7] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics*, 21(4):163–169, July 1987.

[8] C. Madrigal and K. I. Joy. Generating the envelope of a swept trivariate solid. In *Proceedings of the IASTED International Conference on Computer Graphics and Imaging*, pages 5–9, Oct. 1999.

[9] D. Moore. Subdividing simplices. In D. Kirk, editor, *Graphics Gems III*, pages 244–249. Academic Press, 1987.

[10] G. M. Nielson. Tools for triangulations and tetrahedrizations and constructing functions defined over them. In G. M. Nielson, H. Hagen, and H. Müller, editors, *Scientific Visualization: Overviews, Methodologies, and Techniques*, pages 429–525. IEEE Computer Society Press, 1997.

[11] B. O'Neill. *Elementary Differential Geometry*. Academic Press, New York, NY, 1966.

[12] R. Shu, C. Zhou, and M. S. Kankanhalli. Adaptive marching cubes. *The Visual Computer*, 11(4):202–217, 1995.

[13] W. P. Wang and K. K. Wang. Geometric modeling for swept volume of moving solids. *IEEE Computer Graphics and Applications*, 6(12):8–17, Dec. 1986.

[14] Y. Zhou, B. Chen, and A. Kaufman. Multiresolution tetrahedral framework for visualizing regular volume data. In R. Yagel and H. Hagen, editors, *IEEE Visualization 9́7*, pages 135–142. IEEE, Nov. 1997.