Swept Volumes

M. Peternell[1], H. Pottmann[1], T. Steiner[1], H. Zhao[2]

[1] *Vienna Univ. of Technology, {peternell, pottmann, tibor}@geometrie.tuwien.ac.at*
[2] *University of California, Irvine, zhao@math.uci.edu.*

Abstract

Given a solid $S \subset \mathbb{R}^3$ with a piecewise smooth boundary, we compute an approximation of the boundary surface of the volume which is swept by S under a smooth one-parameter motion. Using knowledge from kinematical and elementary differential geometry, the algorithm computes a set of points plus surface normals from the envelope surface. A study of the evolution speed of the so called characteristic set along the envelope is used to achieve a prescribed sampling density. With a marching algorithm in a grid, the part of the envelope which lies on the boundary of the swept volume is extracted. The final boundary representation of the swept volume is either a triangle mesh, a B-spline surface or a point-set surface.

Keywords: motion, swept volume, envelope, marching algorithm, point-set surface, NC verification.

1 Introduction

The swept volume of a solid S is formed by all points in space which are covered by positions of S during a motion. The most frequently considered motions are those which depend smoothly on a parameter t to be thought of as time. The arising swept volumes have a variety of applications including NC machining verification, geometric modeling, robot workspace analysis, collision detection, motion planning, etc., see e.g. [1, 11].

1.1 Previous work

The theoretical foundation of swept volumes lies in the computation of envelope surfaces in kinematical geometry. This is classical knowledge and can be found in a number of textbooks [5, 16].

Probably the first contribution on envelopes dealing with computational aspects is the PhD thesis of J. Hoschek [9]. There is a number of papers which provide explicit and even rational parameterizations of envelope surfaces (see the survey [13]). In general, however, one can only achieve a numerical approximation of the boundary surface of a swept volume.

Although there is already a pretty large number of contributions on this topic (for a survey, see [1, 4]), the problem cannot be considered as sufficiently well solved and thus swept volumes are hardly included in today's CAD systems.

From the most recent work we point to a paper by Kim et al. [11] which addresses swept volumes of moving polyhedra and exploits the fact that in this case we have only ruled or developable surfaces on the boundary of the swept volume.

1.2 Contribution of the article

Our paper does not make any special assumption on the moving solid S apart from the natural requirement that its boundary shall be a piecewise smooth surface Φ. As many other papers in the past, we use envelope theory to compute the so-called characteristic sets at a discrete number of time instances. This immediately gives a cloud of points plus surface normals in a superset Γ of the boundary surface Ψ of the swept volume. As a new component in our algorithm we use a formula for the evolution speed of the characteristic curve to get a good time step selection which yields a nice data point distribution on Ψ and Γ.

A kind of marching algorithm on an underlying grid is used to trim away the part of the envelope surface Γ which lies in the interior of the swept volume. Special care is taken to correctly resolve edges at the boundary of the swept volume. At this stage we have a sufficiently dense set of points plus surface normals — a set of *surfels* in Graphics terminology — at the boundary surface Ψ of the swept volume. Now, any method that can interpolate or approximate such data is appropriate to get the final boundary representation. We present solutions with a triangle mesh, a point-set surface and B-spline surfaces.

The novelty of our method lies in the combination of classical results of kinematical and differential geometry with recent developments in geometry processing such as fast marching, point-set surfaces [2, 3, 12] and B-spline approximation using squared distance minimization [14].

2 Geometric background

2.1 The velocity vector field

Kinematical geometry provides the necessary geometric background for our problem. Thus we briefly review a few basics and refer to the literature for more information [5, 16].

Consider a rigid body moving in Euclidean three-space $\mathbb{R}^3$. We think of two copies of $\mathbb{R}^3$: One copy associated with the moving body and called *moving space* or *moving system* Σ, and one copy called the *fixed space* or *fixed system* Σ^0. We use Cartesian coordinates and denote points of the moving system Σ by $\mathbf{x}, \mathbf{y}, \dots$, and points of the fixed system by $\mathbf{x}^0$, $\mathbf{y}^0$, and so on.

A *one-parameter motion* Σ/Σ^0 is a smooth family of Euclidean congruence transformations depending on a parameter t which can be thought of as time. A point $\mathbf{x}$ of Σ is, at time t, mapped to the point

$$\mathbf{x}^0(t) = A(t) \cdot \mathbf{x} + \mathbf{a}^0(t) \qquad (1)$$

of Σ^0. Each point $\mathbf{x}$ of Σ has a *path curve* or *trajectory* $\mathbf{x}^0(t)$ in Σ^0. The trajectory of the origin is $\mathbf{a}^0(t)$. $A(t)$ describes the rotational part of the motion; we have $A^T = A^{-1}$ and $\det(A) = 1$.

The first derivative $\dot{\mathbf{x}}^0(t) = \dot{A}(t) \cdot \mathbf{x} + \dot{\mathbf{a}}^0(t)$ of the path of $\mathbf{x}$ is its *velocity vector* at time t. We write $\mathbf{v}(\mathbf{x}^0)$ for the vector field of vectors $\dot{\mathbf{x}}^0(t)$ attached to the points $\mathbf{x}^0(t)$. The same velocity field, represented in the moving system, is $\mathbf{v}(\mathbf{x}) = A(t)^T \cdot \dot{\mathbf{x}}^0(t) = A(t)^T \cdot [\dot{A}(t) \cdot \mathbf{x} + \dot{\mathbf{a}}^0(t)]$. Since A is orthogonal, $A^T \cdot \dot{A}$ is skew symmetric. This gives immediately the well-known fact that the vector field $\mathbf{v}(\mathbf{x})$ is linear and has the special form

$$\mathbf{v}(\mathbf{x}) = \overline{\mathbf{c}} + \mathbf{c} \times \mathbf{x}, \quad \text{where} \qquad (2)$$

$$\dot{A}(t) \cdot A^T(t) = \begin{pmatrix} 0 & -c_3 & c_2 \\ c_3 & 0 & -c_1 \\ -c_2 & c_1 & 0 \end{pmatrix}, \quad \mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, \quad \overline{\mathbf{c}} = A^T \cdot \dot{\mathbf{a}}^0(t).$$

The vector $\mathbf{c}$ is called *vector of angular velocity.*

Up to the first differentiation order, any one-parameter motion agrees locally with a *uniform* motion, whose velocity vector field is constant over time. Apart from the trivial uniform motion, where nothing moves at all and all velocities are zero, there are the following three cases: Uniform *translations* have $\mathbf{c} = \mathbf{o}$, but $\overline{\mathbf{c}} \neq \mathbf{o}$, i.e., all velocity vectors equal $\overline{\mathbf{c}}$. Uniform *rotations* with nonzero angular velocity about a fixed axis are given by $\mathbf{c} \cdot \overline{\mathbf{c}} = 0$, but $\mathbf{c} \neq \mathbf{o}$. Uniform *helical motions* are the superposition of a uniform rotation and a uniform translation parallel to the rotation's axis. They are characterized by $\mathbf{c} \cdot \overline{\mathbf{c}} \neq 0$.

2.2 Envelope condition and characteristic curve

For our problem, we have to consider the solid S in the moving system and its boundary surface Φ. Let us first assume that Φ is smooth. Then, at any time instant t, we consider those points of Φ, whose velocity vector is tangent to Φ. In general, the set of these so-called *characteristic points* is a curve, called *characteristic curve* $C(t)$. Note that $C(t)$ may be composed of several connected components. If $\mathbf{n}(\mathbf{x})$ is a normal vector field of Φ, the curve $C(t)$ is the zero level set of the function

$$G(\mathbf{x}) := \mathbf{n}(\mathbf{x}) \cdot \mathbf{v}(\mathbf{x}) = \mathbf{n}(\mathbf{x}) \cdot (\overline{\mathbf{c}} + \mathbf{c} \times \mathbf{x}), \tag{3}$$

which is defined on Φ. For a line geometric interpretation of the envelope condition $G(\mathbf{x}) = 0$ see [16]. The set of all positions $C^0(t)$ of characteristic curves in the fixed system is the complete envelope surface Γ of Φ under the given motion. However, only part of the surface Γ lies on the boundary of the swept volume. Our algorithm will at first compute Γ via the characteristic curves and later remove that part of Γ which is inside the swept volume.

Remark 1. At a time instant with $\mathbf{c} = 0$ (instantaneous translation), the characteristic curve is the silhouette of Φ for parallel projection in direction $\overline{\mathbf{c}}$. In general, however, this is not true and the rotational part needs to be considered. We warn the reader that there is a number of papers in the literature, which consider any time instant as translational in the context of envelope computations.

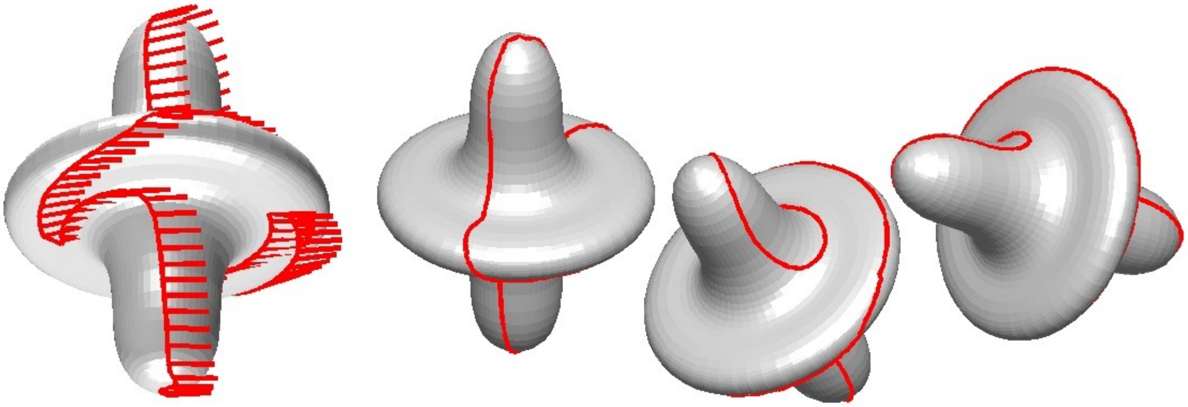

Figure 1: Left: Velocity vectors in points of the characteristic curve of an object. Right: Three positions of the moving object.

Let us briefly discuss the case where Φ is given as a level set $F(\mathbf{x}) = 0$. Since surface normals are given by the gradient field ∇F of F, the characteristic curve also satisfies

$$G(\mathbf{x}) = \nabla F(\mathbf{x}) \cdot (\overline{\mathbf{c}} + \mathbf{c} \times \mathbf{x}) = 0.$$

Note that the surface $G(\mathbf{x}) = 0$ simultaneously contains the characteristic curves of all level sets $F(\mathbf{x}) = k = const$ of F.

The positions of the characteristic curves in the fixed system can also be found as follows: Φ's position in Σ^0 at time t is $F^0(\mathbf{x}^0) := F(A^T \cdot (\mathbf{x}^0 - \mathbf{a}^0)) = 0$. From general envelope theory, the derivative of this equation with respect to t is known to contain the characteristic curve. This derivative surface is precisely the position of the surface $G(\mathbf{x}) = 0$ in Σ^0.

We also get some insight into the case of a moving algebraic surface $F(\mathbf{x}) = 0$ of degree n. Its positions $F^0(\mathbf{x}^0) = 0$ in Σ^0 are represented by a polynomial $F^0(\mathbf{x}^0)$ with coefficients depending on t.

The derivative with respect to t is a polynomial of degree $\leq n$ and thus $C^0(t)$ is by Bezout's theorem an algebraic curve of degree $\leq n^2$. This explains the well-known simplicity of the cases of a moving plane (C is a straight line), a sphere (C is a circle), a quadratic cylinder or cone (C is a rational curve of degree ≤ 4), or a torus (C is algebraic of degree ≤ 8; as for the sphere, the degree reduction is caused by the absolute conic). But even for simple freeform surfaces, the degree will become very high and thus algebraic methods seem to be infeasible for our purpose.

2.3 Non-smooth boundary surface or non-smooth motions

If the boundary surface Φ of S contains sharp edges and corners, the considerations above remain correct if we generalize the meaning of the normal vector $\mathbf{n}$. In order to get a good feeling what we have to do at these places, one may consider them smoothed by a rolling ball blend of a very small radius.

At an edge point $\mathbf{p}$, there are two different outside unit normals $\mathbf{n}_l, \mathbf{n}_r$ at either side of it. The set of normals at $\mathbf{p}$ is then of the form $(1-h)\lambda_l \mathbf{n}_l + h\lambda_r \mathbf{n}_r$ with $h \in [0,1]$. The corresponding unit normals form an arc of a great circle on the unit sphere.

At a corner point $\mathbf{p}$, the set of normals is even two-dimensional; it is defined by the convex cone to the unit normals of the surface patches which meet at $\mathbf{p}$.

We see that the envelope condition becomes weaker at edges or corners so that those appear more likely as part of the characteristic curve.

If the motion is not smooth at some time instant t_0, we may also consider a locally smoothed version. The left-sided derivative at $t = t_0$ yields a velocity field characterized by $(\mathbf{c}_l, \overline{\mathbf{c}}_l)$, and on the right side we have $(\mathbf{c}_r, \overline{\mathbf{c}}_r)$. This yields a one-parameter family of vector fields $(1-h)(\mathbf{c}_l, \overline{\mathbf{c}}_l) + h(\mathbf{c}_r, \overline{\mathbf{c}}_r),\ h \in [0,1]$. It has to be considered for the computation of the characteristic set which is in general in a two-dimensional subset of Φ. For an initial position of a non-closed motion, we define all points with $G \leq 0$ as characteristic, and for an end position all points with $G \geq 0$. The entire surface Φ arises as characteristic set for an instant with $(\mathbf{c}, \overline{\mathbf{c}}) = (0,0)$.

2.4 Evolution of characteristic curves

In our algorithm we will sample the motion at discrete time instances $t_0, \ldots, t_n$ and compute the corresponding characteristic sets $C^0(t_i)$ in Σ^0. In order to get a good representation of the envelope surface, the distances between adjacent curves $C^0(t_i)$ and $C^0(t_{i+1})$ should not become too large. In other words, our time sampling has to respect the evolution speed of the curves $C^0(t)$. We will derive here an equation for the speed of this evolution. This is similar to results on the evolution of apparent contours on a surface, which have been studied extensively in Computer Vision [6].

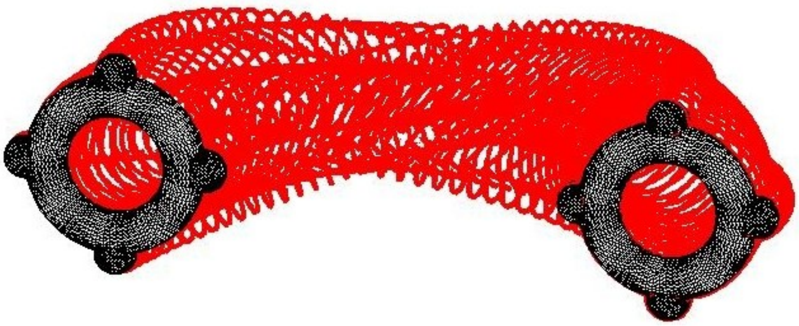

Figure 2: Characteristic curves together with start and end position of the moving object.

Let us first look into the moving system and consider – just for the sake of simplicity of the derivation – a local parametrization $\mathbf{s}(u,t)$ of Φ in which the iso-parameter lines $t = const$ are the characteristic curves $C(t)$ and $u = const$ are orthogonal trajectories of those. We are interested in the evolution speed, which is measured orthogonal to $C(t)$. Thus, it is given by the vector $\partial\mathbf{s}/\partial t =: \mathbf{s}_t = \lambda\mathbf{r}$ (with $\mathbf{r} := \mathbf{s}_t/\|\mathbf{s}_t\|$). With $\mathbf{n}(u,t)$ as unit normal vector field of Φ, the parametrization $\mathbf{s}$ must satisfy condition (3), $\mathbf{n}(u,t)\cdot[\overline{\mathbf{c}}(t) + \mathbf{c}(t)\times\mathbf{s}(u,t)] = 0$. Differentiation of this identity with respect to t yields,

$$\mathbf{n}_t\cdot(\overline{\mathbf{c}} + \mathbf{c}\times\mathbf{s}) + \mathbf{n}\cdot(\dot{\overline{\mathbf{c}}} + \dot{\mathbf{c}}\times\mathbf{s} + \mathbf{c}\times\mathbf{s}_t) = 0. \tag{4}$$

Let v denote an arc length parameter along the parameter line $u = const$ passing through the considered point $\mathbf{s}(u,t)$. Then,

$$\mathbf{n}_t = \mathbf{n}_v\frac{dv}{dt} = \mathbf{n}_v\|\mathbf{s}_t\| = \lambda\mathbf{n}_v.$$

The variation $\mathbf{n}_v$ of the normal in direction $\mathbf{r}$ is given by

$$\mathbf{n}_v = -\omega(\mathbf{r}). \tag{5}$$

Here, ω is the Weingarten map (shape operator) [7]. We use a right handed Cartesian system $\mathbf{r},\mathbf{h},\mathbf{n}$ at $\mathbf{s}(u,t)$ and solve (4) for the *evolution speed in the moving system* λ,

$$\lambda = \frac{\mathbf{n}\cdot(\dot{\overline{\mathbf{c}}} + \dot{\mathbf{c}}\times\mathbf{s})}{\omega(\mathbf{r})\cdot\mathbf{v}(\mathbf{s}) + \mathbf{c}\cdot\mathbf{h}}. \tag{6}$$

In the numerator we have the component of the acceleration of $\mathbf{s}$ in direction of the surface normal. The denominator contains only first order information of the motion, namely components of the velocity $\mathbf{v}(\mathbf{s})$ and of the angular velocity vector $\mathbf{c}$; the directions $\mathbf{h},\mathbf{r}$ and $\omega(\mathbf{r})$ however depend on second order surface information. Considering a point $\mathbf{s}$ which moves on Φ with the evolution velocity vector $\lambda\mathbf{r}$, the corresponding velocity vector in the fixed system (but represented in the moving system) is the sum $\lambda\mathbf{r} + \mathbf{v}(\mathbf{s})$. Since we are again only interested in the component orthogonal to $C(t)$, we find for the magnitude of the *evolution speed in the fixed system*

$$\lambda^0 = |\lambda + \mathbf{r}\cdot\mathbf{v}(\mathbf{s})|. \tag{7}$$

3 Outline of the algorithm

In order to simplify the presentation, we first consider the case where we are not interested in eventually appearing holes inside the swept volume. Even if there are holes, a number of applications is not interested in them. How to handle holes will be outlined in section 7.

Our algorithm contains the following three basic steps, which will be explained in more detail in the following sections.

1. At a number of time instances $t_0,\dots,t_n$, chosen with an appropriate time step control, we compute the respective positions $C^0(t_i)$ of the characteristic set in the fixed system. The characteristic set is represented as a set of points plus surface normals. This results in a set Q.

2. Based on a marching algorithm in an underlying grid of the fixed system, we remove those points from Q which lie in the interior of the swept volume. This yields a new point set B on the boundary of the swept volume.

3. We compute an approximating surface to the point set B, which respects the provided surface normals well.

For the detailed explanation we assume that the boundary surface Φ of the moving surface is given as a triangle mesh (with reliably estimated surface normals) or a parametric surface. We also assume that the motion can be evaluated at any time instant. If the motion is only given at a discrete number of positions, a motion interpolation scheme (see e.g. [10, 17]) might be required if the time steps are not satisfying our criteria.

4 Computation of characteristic points

In order to compute the characteristic set $C(t_i)$ at a given time instant t_i, we set up the velocity vector field $(\mathbf{c}, \overline{\mathbf{c}})$ in the moving system, and search for the zero level set of the function F (inner product between velocity vector and surface normal) in equation (3). It is sufficient to compute a sequence of points plus surface normals along $C(t_i)$. The maximum distance of adjacent points along the same component of $C(t_i)$ shall be bounded by a prescribed value δ. Finally, the positions $\mathbf{q}_i$ of characteristic points and associated surface normals in the fixed system Σ^0 are computed.

Of course, at discontinuities of motion or surface, we have to compute characteristic points according to subsection 2.3.

Along a given characteristic $C(t_i)$ we estimate the maximum evolution velocity v_{max} in the fixed system using equation (7). Then, a time step $\Delta t_i := t_{i+1} - t_i$ is chosen such that

$$v_{max}\Delta t_i \leq \delta. \tag{8}$$

In this way the spacing across characteristics is in general not much larger than the spacing of points along the characteristic. If one wants to avoid second order estimates arising in (7), one can use the pairs of corresponding points $\mathbf{q}_j^- \in C^0(t_{i-1})$ and $\mathbf{q}_j \in C^0(t_i)$ (such that $\mathbf{q}_j - \mathbf{q}_j^-$ is orthogonal to $C^0(t_i)$) to get an estimate of v_{max} via $v_{max} = (\max\|\mathbf{q}_j - \mathbf{q}_j^-\|)/\Delta t_{i-1}$. If $\max\|\mathbf{q}_j - \mathbf{q}_j^-\|$ exceeds δ, we can reduce t_{i-1} accordingly and recompute $C^0(t_i)$.

5 Selection of points on the boundary of the swept volume

The result of the previous step is a point set Q (plus surface normals) on the envelope surface Γ. Note that we do not care about holes inside a swept volume and thus we proceed as follows to compute the outer boundary surface of the swept volume.

The data point set Q is placed in a rectangular grid whose cell size is larger than δ. We compute the unsigned distance function for the grid points with respect to Q. In order to trim parts of Q to obtain the trimmed envelope Γ we perform a fast marching method similar to that described in [11] on the grid points.

Roughly this works as follows: At the beginning all grid points are labeled as *unknown* and *inside.* We begin the loop by taking a point surely outside of Γ and initialize an *active front* by storing this point in it and label it as *trial.* We extract a point $\mathbf{p}$ from the active front and this point $\mathbf{p}$ is labeled as *visited* and *outside.* We collect all its *unknown* neighbors $\mathbf{q}$ on the grid. If the distance value of $\mathbf{q}$ is larger than some threshold τ, $\mathbf{q}$ is labeled as *outside* and *trial* and $\mathbf{q}$ is added to the active front. In this manner the front propagates on the grid and this procedure is performed until the active front is empty.

The active front stops at the outer boundary of the envelope Γ. We perform a nearest neighbor search for all those grid points close to Γ with respect to the point set Q which is actually formed by points of the characteristic curves, see sections 2.2 and 2.4. This results in a relatively uniform sampling of Γ by a subset B of points of the set Q. We note that also the normal vectors of Γ at points of B are available.

Self intersection of the envelope Γ are smoothed out and are treated in a separate procedure (section 5.1).

5.1 Handling edges and features

There are methods for the computation of intersections of point sets in the literature, see e.g. [2, 12]. We follow a different approach and additionally use a triangulation of the point set B of the trimmed envelope Γ.

The method for detection of sharp features on Γ is motivated by the following observations:

- The envelope Γ possesses a relatively uniform sampling by the point set B.
- The distances of points of B do not vary much and thus the size of triangles and the length of their segments is nearly constant.

According to these observations we detect sharp features by investigating distances of adjacent points taking into account the variation of their normal vectors. Let $\mathbf{p} \in B$ and $\mathbf{n}$ an arbitrary point and its (unit) normal vector. We form the surface $\hat{B} \subset \mathbb{R}^6$ consisting of points $\hat{\mathbf{p}} = (\mathbf{p}, w\mathbf{n})$. The canonical metric in $\mathbb{R}^6$ induces a metric on $\hat{B}$ and this defines a *feature sensitive* metric on B. The value w controls the influence of the normal vectors.

See [15] for more information concerning the feature sensitive metric.

Once having defined the feature locations on B we form connected components (on the triangulation) and investigate the neighborhood of each connected component (with respect to the triangulation) of the sharp features. Each connected component is replaced by a spine curve which represents the shape of the sharp feature.

Then profile sections of the investigated neighborhoods, orthogonal to the spine curve are computed. For typical self intersections we can recognize different sides of the neighborhood of a sharp feature. The intersection points of the approximating lines of the profile sections are good estimates for the true self intersection. The intersection points are sorted and smoothed by a spline curve which serves as approximation of the sharp feature. After having inserted a regularly sampled sequence of points of each of these spline curves into the data set B, a new triangulation is performed, see Fig. 3.

Note that there is an easy kind of sharp edges: they are (parts of) trajectories of corner points of the moving solid and do not require the present approach.

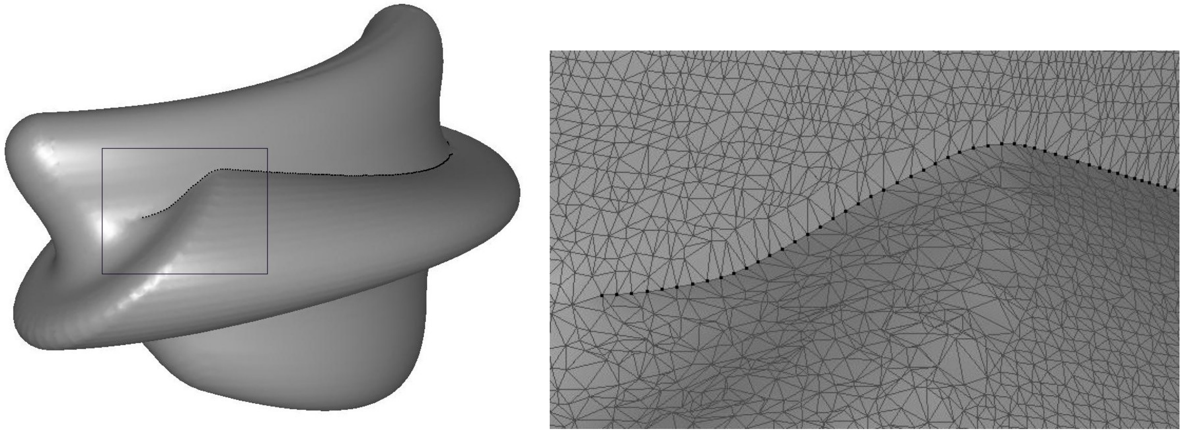

Figure 3: Right: Global self intersection of the swept volume. Left: Triangulation and the inserted points along the sharp edge of the detail.

6 Approximating the boundary surface of the swept volume

6.1 Approximating triangle mesh

Since the careful handling of features labels edges, we can perform a good triangulation. The edge points have already been assembled into polygons. So we just have to perform a triangulation of the point set which contains these edge polygons.

6.2 B-spline approximation

Especially if the boundary surface of the swept volume has a simple topology, a recently developed B-spline approximation scheme called *squared distance minimization* [14] can be used. We do not recommend to use the full squared distance formula which also requires curvature estimates, but just the corresponding Gauss-Newton iteration that is based on surfels. Figure 5 displays an example, where the self intersections are smoothed.

6.3 Approximating point-set surface

The definition of the boundary of the swept volume with a set of points (surfels) allows us to apply recent work on point-set surfaces. For rendering, we need a disk radius of the surfels; it is derived from the sampling density δ as $\lambda\delta$ with $\lambda > 1/\sqrt{3}$. If we did not recover edges on the boundary of the swept volume, there are many good methods in the literature, e.g. [2, 3, 12], to represent these data as a point-set surface (implicit surface). Since surfels are available, the use of corresponding point-set surfaces according to Amenta and Kim [3] is recommended. In the presence of edges special care has to be taken, as outlined e.g. in [12].

7 Extensions

Our method can be extended in various ways. We list here a only few items and point to a forthcoming journal publication in which we will address these topics in detail.

1. In order to resolve holes inside the swept volume, one can write information into the grid in Σ^0 already during the sweep through the motion. Basically, we have to mark grid cells which are at least partially covered by a position of the moving body. One way to do this is to sample Φ (can be coarser than the sampling used for the computation of characteristic points); this gives a set I. At any time instant t_i, we then compute all points $\mathbf{x} \in I$ with $G(\mathbf{x}) > 0$ and label those cells into which their positions fall.

2. The main ideas of the method also work for an implicit representation of the boundary surface Φ of S. We are currently exploring an approach which derives a signed distance representation of the swept volume boundary from a signed distance representation of Φ.

3. The present approach can be extended to swept volumes generated by motions which depend on more than one parameter. For example, we may consider two-parameter motions. An important special case of those, namely purely translational motions, are generating *Minkowski sums*, which form the basic building block of mathematical morphology [8].

Acknowledgements This research has been carried out as part of the project P16002-N05 of the Austrian Science Fund (FWF), and is partially supported by the innovative project '3D Technology' of Vienna University of Technology. Additionally this work has been carried out partially within the Kplus competence center *Advanced Computer Vision*.

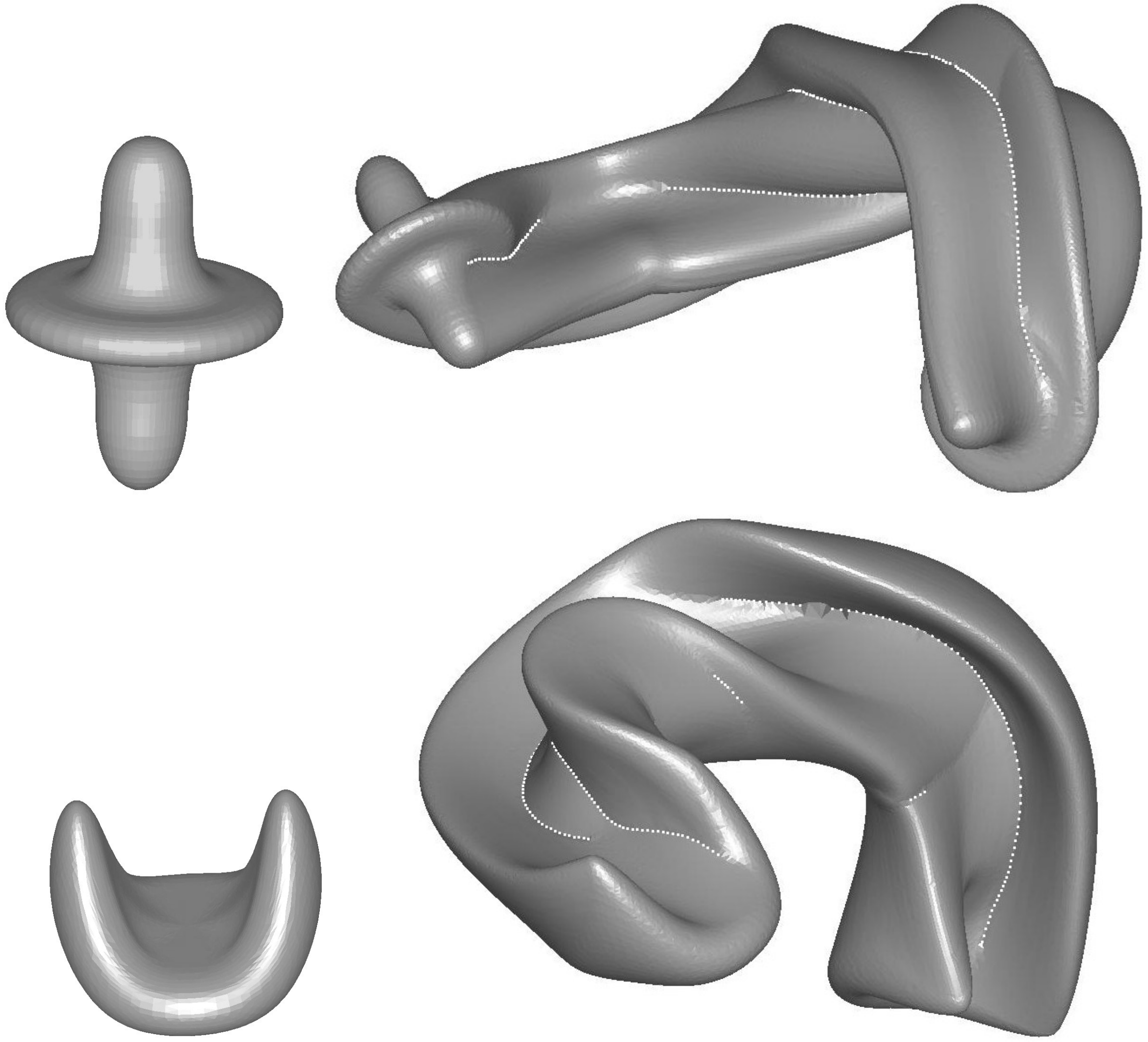

Figure 4: Objects and their swept volume with respect to some motions

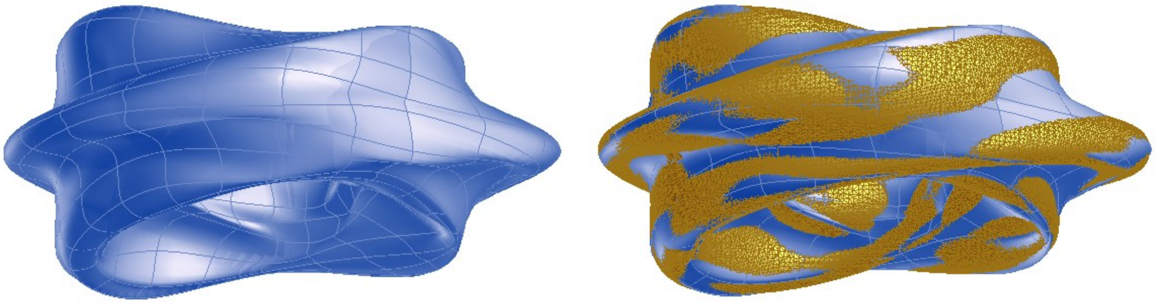

Figure 5: B-spline approximation of the swept volume. Left: B-spline approximation. Right: B-spline approximation superimposed with the triangulated data points.

References

[1] K. Abdel-Malek, D. Blackmore, and K. Joy. Swept volumes: foundations, perspectives and applications. *Intl. Journal of Shape Modeling*, 2004.

[2] Anders Adamson and Marc Alexa. Approximating and intersecting surfaces from points. In *Geometry Processing 2003*, pages 245–254. Eurographics, 2003.

[3] Nina Amenta and Yong Joo Kil. Defining point-set surfaces. In *Proceedings of SIGGRAPH 2004*, Computer Graphics Proceedings, Annual Conference Series, pages 264–270. ACM, ACM Press / ACM SIGGRAPH, 2004.

[4] D. Blackmore, M. C. Leu, L. P. Wang, and H. Jiang. Swept volume: A retrospective and prospective view. *Neural, Parallel and Scientific Computations*, 5(1):81–102, 1997.

[5] O. Bottema and B. Roth. *Theoretical Kinematics.* Dover Publ., New York, 1990.

[6] Roberto Cipolla and Peter Giblin. *Visual Motion of Curves and Surfaces.* Cambridge Univ. Press, Cambridge, UK, 2000.

[7] Manfredo P. do Carmo. *Differential Geometry of Curves and Surfaces.* Prentice-Hall, 1976.

[8] H. J. A. M. Heijmans. *Morphological Image Operators.* Academic Press, Boston, 1994.

[9] Josef Hoschek. *Zur Ermittlung von Hüllgebilden in der Kinematik.* PhD thesis, TH Darmstadt, 1964.

[10] Bert Jüttler and Michael Wagner. Kinematics and animation. In G. Farin, M. S. Kim, and Josef Hoschek, editors, *Handbook of Computer Aided Geometric Design*, pages 723–748. Elsevier, 2002.

[11] Young J. Kim, Gokul Varadhan, Ming C. Lin, and Dinesh Manocha. Fast swept volume approximation of complex polyhedral models. *Computer Aided Design*, 36:1013–1027, 2004.

[12] M. Pauly, R. Keiser, L. Kobbelt, and M. Gross. Shape modeling with point-sampled geometry. In *Proceedings of SIGGRAPH 2003*, Computer Graphics Proceedings, Annual Conference Series, pages 641–650. ACM, ACM Press / ACM SIGGRAPH, 2003.

[13] H. Pottmann and S. Leopoldseder. Geometries for CAGD. In G. Farin, J. Hoschek, and M.S. Kim, editors, *Handbook of Computer Aided Geometric Design*, pages 43–73. Elsevier, 2002.

[14] H. Pottmann and S. Leopoldseder. A concept for parametric surface fitting which avoids the parametrization problem. *Computer Aided Geometric Design*, 20:343–362, 2003.

[15] H. Pottmann, T. Steiner, M. Hofer, C. Haider, and A. Hanbury. The isophotic metric and its application to feature sensitive morphology on surfaces. In T. Pajdla and J. Matas, editors, *Computer Vision — ECCV 2004, Part IV*, volume 3024 of *Lecture Notes in Computer Science*, pages 560–572. Springer, 2004.

[16] H. Pottmann and J. Wallner. *Computational Line Geometry.* Springer, Berlin-Heidelberg-New York, 2001.

[17] J. Wallner and H. Pottmann. Intrinsic subdivision with smooth limits for graphics and animation. *Geometry Preprint Nr.* **120***, TU Wien*, 2004.