Technical report GIT-GVU-06-19 from the GVU Center at Georgia Tech, September 2006

Boundary of the volume swept by a free-form solid in screw motion

J.R. Rossignac[a], J.J. Kim[b], S.C. Song[c], K.C. Suh[b], C.B. Joung[d]

[a]College of Computing, Georgia Institute of Technology, Atlanta, GA, USA
[b]School of Mechanical Engineering, Hanyang University, Seoul, Korea
[c]CIMVI, Catholic University, Pucheon City, Korea
[d]Design Process Group, NIST, Gaithersburg, Maryland, USA

Abstract

The swept volume of a moving solid is a powerful computational and visualization concept. It provides an excellent aid for path and accessibility planning in robotics and for simulating various manufacturing operations. It has proven difficult to evaluate the boundary of the volume swept by a solid bounded by trimmed parametric surfaces undergoing an arbitrary analytic motion. Hence prior solutions use one or several of the following simplifications: (1) approximate the volume by the union of a finite set of solid instances sampled along the motion; (2) approximate the curved solid by a polyhedron, (3) approximate the motion by a sequence of simpler motions. The approach proposed here is based on the third type of simplification: it uses a polyscrew (continuous, piecewise-screw) approximation of the motion. This approach leads to a simple algorithm that generates candidate faces, computes the two-cells of their arrangement, and uses a new point-in-sweep test to select the correct cells whose union forms the boundary of the swept volume.

Keywords: *Screw Motion, Boundary Evaluation, Swept Volume, Trimmed Surfaces*

1. Introduction

In this paper, we focus on the computation of the close approximation of the boundary of the volume swept by a free-form solid moving along a piecewise screw motion (Fig. 1). We say that a solid is ***free-form*** (as opposite to polygonal) when its boundary comprises smooth curved surfaces (for example trimmed bi-cubic or higher order parametric patches). The swept volume of a moving object provides a powerful computational and visualization tool in various applications. For example, it is used in NC machining simulation [El Mounayri98, Sambandan89, Weinert04, Chiou04] to model the volume removed by a cutter. It provides a valuable aid in the determination of collisions between the moving object and static obstacles [Keiffe91].

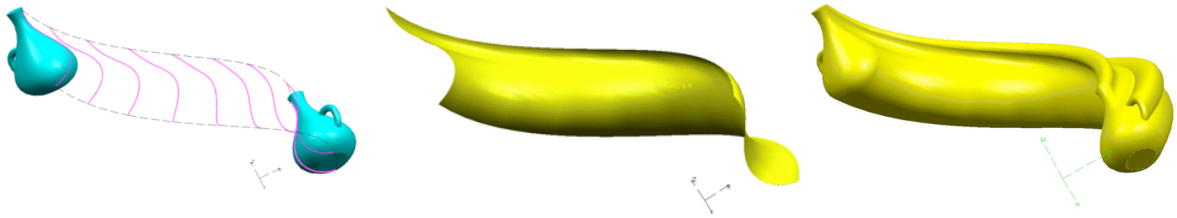

Figure 1: *A solid in screw motion interpolating two key-poses showing several instances of a generator curve G along the motion (left). The face E generated by sweeping a subset of G along the screw motion (center). The boundary of the swept volume made of the trimmed portions of faces of the initial and final instances and of the extrusions of generators (right).*

In this section, below, we explain the limitations and drawbacks of existing algorithms for generating boundaries of swept volumes and stress the research contributions presented in this paper over prior art.

The **volume swept** by a moving solid is completely specified by a representation of the object and of its motion. It is defined as the infinite union of the instances of the solids defined by poses taken during the motion. Unfortunately, such implicit representations are ill-suited for further processing in solid modeling systems, which require the ability of efficiently testing whether a candidate point lies in the swept volume and of producing a representation of the surfaces that bound the volume suitable for computing intersections with rays, curves, or other surfaces, as required for rendering, interference detection, or Boolean operations. Hence, numerous researchers have developed techniques for computing more explicit, although usually approximate, models of the swept volume. We review them below.

Spatial occupancy models divide space into **voxels** (usually aligned along a uniform grid). Binary voxel models may be derived by testing a sample point of each voxels for containment in the swept volume. For general motions, such a test is delicate and computationally expensive, because it involves computing geometric intersections between the boundary of the stationary solid and the path traveled by the candidate point when subject to the inverse motion. Hence, in practice, the motion is sampled and for each pose of the solid along the motion, the solid instance is rasterized (possibly using graphics hardware to perform a 3D rasterization) to classify all the sample point of each voxel. Voxels whose sample point belongs to at lease one

instance are assumed to be in the swept volume. Such volumetric models are valuable for a variety of applications. Unfortunately, unless prohibitively dense grids of voxels and small time steps are used, these models remain crude approximations of the true swept volume. They may miss small topological or morphological features. They also lack the surface continuity that may be necessary for many design, analysis, manufacturing, and simulation applications.

To improve on this time-and-space sampling approach and produce a more accurate model, one may decide to only sample the motion (not the space) and to express the swept volume as the **union** of a finite set of continuous instances of the solid. Computing the union of a large number of instances is not only computationally expensive, but numerically delicate because consecutive instances are nearly aligned, leading to ill-conditioned surface/surface intersection problems where the two surfaces are nearly tangent to each other along the intersection curve. Furthermore, this approach usually produces scalloped approximations of the smooth envelope surfaces that are swept by the sharp convex edges of the moving solid.

To alleviate the aliasing of the voxelized and finite union methods, one must generate a finite set of smooth surfaces, which, when combined with the surfaces bounding the initial and final instances, are guaranteed to contain the boundary of the swept volume. These smooth (**envelope**) surfaces are swept by characteristic points on the boundary of the moving solid. At any given time t during the motion, the characteristic points are fully defined as pointsets that satisfy equality or inequality equations. These equations simply cancel the dot-product between surface normal and the instantaneous velocity vector. The boundary of the swept volume is a subset of the union of these characteristic points gathered for all times t during the motion. Hence, since t is a parameter in these dot-product equations, we have a time-parameterized implicit definition of a superset of the boundary of the swept volume. Unfortunately, such implicit definitions cannot be directly processed by contemporary CAD/CAM systems. A natural solution is to convert these implicit equations into parametric representations of surfaces swept by these characteristic points as the slide on the surface and move with the object over time.

To summarize, previously proposed sampling solutions produce non-smooth approximations or require numerically delicate and computationally prohibitive Boolean on a large number of instances, while previously proposed implicit formulations are impractical for further processing in application systems.

One of the research **contribution** proposed in this paper is precisely focused on this problem of computing smooth parametric patches that accurately approximate these swept surfaces. Note that we must address two problems. The first one is to turn the standard implicit (dot-product) formulation of the characteristic points into a set of **parametric curves**. The second one is to establish a parametric **correspondence** between the time-evolving instances of these curves, which in general slide along the surface and may change topology.

The first problem (structure the characteristic points into parameterized curves) has been addressed previously for a variety of motion types and surface types. At any given time t, the desired curves bound the portion of the boundary where the instantaneous velocity vectors points outwards of the solid instance. We introduce here a practical solution for computing the characteristic curves on bi-parametric surfaces. In particular, we present a method that generates an **approximation of the characteristic curve on each bi-parametric patch**. We provide the system of differential equations, which we solve to **compute sample points along the characteristic curves as geometric intersections of parametric lines and curves in the parameter space of the bi-parametric surface**.

The second problem (correspondence through time) is more delicate, because for general motions, the characteristic curve is not fixed on the moving object, but evolves (i.e. slides along the surface of the solid) with time. Several authors generate the characteristic curves for successive time samples and then attempt to interpolate them by a piecewise linear or higher order surface. This process is unreliable, because the characteristic curves for two consecutive time samples may be drastically different. In fact, they may have different numbers of components. Even if a one-to-one matching between components can be established, the shapes and placement of the components may be drastically different. We eliminate this problem by restricting the nature of the motion to be a polyscrew (i.e. a smooth piecewise screw motion). Indeed, during a screw motion segment, the characteristic curve is **identical** (i.e., rigidly attached to the moving solid) on all instances, hence the mapping between the instances of the curves at consecutive time samples is the identity. The choice of using piecewise screw approximations of an arbitrary motion is justified by the fact that the **instantaneous approximation of any motion is a screw** (an instantaneous 3D motion cannot be modeled as a pure rotation or a pure translation) and by the fact that **smooth polyscrew** (piecewise-screw) motion that interpolates a given sequence of key-poses may be trivially obtained by subdivision in realtime [Powell05].

With this approach, we can apply the "generate, split, and test" paradigm used in solid modeling for **boundary evaluation** to the computation of the boundaries of swept volume. We believe that this has not been done previously. This paradigm simplifies the implementation and leverages commercially available code for computing intersections of surfaces and curves. In our approach, we generate a sufficient set of **candidate surfaces** as explained above, then, to **trim** them to the portion on the boundary of the swept volume, we **divide** them into cells along their curves of intersection with other surfaces, finally, we **classify the cells** to discard those in the interior of the swept volume and to retain those on the boundary of the swept volume. We use a commercial system to compute the surface/surface intersections robustly and efficiently.

Since, during the computation of the boundary of the swept volume, the boundary is not known, we cannot use it for testing whether the given point lies in the interior of the swept volume. However, the candidate set is split into cells, such that each cell is either on the boundary of the swept volume or entirely in its interior. One may be tempted to conclude that cells that

cannot be reached from infinity without crossing another cell are necessarily interior. Unfortunately, the assumption is wrong, since the boundary of the swept volume may comprise several shells, some of which are bounding interior cavities. For example, consider a torus T swept by a ball of radius r initially positioned at (2r,0,0) as it rotates 360 degrees around the z-axis. Now, let W be the volume swept by T as T is rotated around the x-axis. W is a ball of radius 3r that has an interior spherical cavity of radius r. Hence, previously proposed approaches based on this assumption are wrong. Algorithms for correctly classifying a point against a swept volume are challenging. We introduce here a simple solution, which is practical for the case of screw motions. For the **classification** of the cells against the swept volume, we use a novel **screw-shooting technique**, which builds upon the classic "ray-shooting" approach for point-membership classification used in graphics and modeling. To classify a cell, we trace a helix from a point of the cell and test whether the helix intersects the initial instance of the solid. This helix-solid intersection test is well supported in commercial CAD systems.

To summarize, the research contributions reported here are:

- The formulation of the swept volume boundary computation as a "generate, split, and test" **paradigm**
- The **equations** identifying the characteristic curves on bi-parametric patches under **screw motion**
- A practical technique for **constructing parametric patches** that approximate the envelop surfaces
- An efficient **algorithm for testing** whether a **point or cell** lies on the boundary of the swept volume
- A formulation that makes it possible to **use commercial packages** for computing the needed intersections

In the remainder of the paper, we define our terminology (2), review the construction and smoothing of screw motions (3), overview the underlying principles upon which our approach is based (4), explain how to compute the characteristic curves on smooth surfaces (5) and on sharp edges (6), explain how to extrude them into surfaces (7), how to subdivide them into cells (8), and how to classify each cell (9). Then we discuss the details of the particular implementation we have produced (10).

2. Poses, motions, and swept volumes

To define the swept volume, we first define a pose and a motion. A ***pose*** corresponds to a rigid body transformation, which is a combination of a rotation with a translation. Let W be a solid and let $\boldsymbol{P}$ be a pose. Often $\boldsymbol{P}$ is represented by a 3×4 matrix [I J K O], whose column vectors define the images by $\boldsymbol{P}$ of the three basis vectors and of the origin of the global coordinate system. Hence the three vectors (I,J,K) form the ortho-normal basis of the local coordinate system and point O is its origin. By W@$\boldsymbol{P}$ we denote the ***instance*** of W transformed by $\boldsymbol{P}$. Hence, $\boldsymbol{P}$ is a bijection that maps each point (x,y,z) of W onto the point $O+xI+yJ+zK$ of W@$\boldsymbol{P}$. A ***motion*** may be defined by a pose-valued function $\boldsymbol{P}(t)$ of time-parameter t. We use the notation $\boldsymbol{P}(.)$ when referring to the family of poses for all values of t in a given domain. Hence, W@$\boldsymbol{P}(.)$ describes a motion of W, i.e., an animation. In this paper, we study motion ***segments*** that each interpolate two poses, an ***initial pose*** $\boldsymbol{A}$ and a **final pose** $\boldsymbol{B}$. To simplify discussion and without loss of generality, we re-parameterize time so, that during the motion, $t\in[0,1]$ with $\boldsymbol{P}(0)=\boldsymbol{A}$ and $\boldsymbol{P}(1)=\boldsymbol{B}$. To further simplify notation, the pose at time t of such an interpolating motion segment (hereafter simply called motion) will be denoted $\boldsymbol{P}(t,\boldsymbol{A},\boldsymbol{B})$. As W is moved by $\boldsymbol{P}(t,\boldsymbol{A},\boldsymbol{B})$, its instance at time t will be denoted by W@$\boldsymbol{P}(t,\boldsymbol{A},\boldsymbol{B})$. W@$\boldsymbol{A}$ and W@$\boldsymbol{B}$ are called the ***initial*** and ***final*** instances. A moving object may be represented by a hyper-solid H in four-dimensional space, such that a slice of H by a hyper-plane of constant t corresponds to W@$\boldsymbol{P}(t,\boldsymbol{A},\boldsymbol{B})$. The 3D ***volume swept*** by W during this motion is the normal projection of this hyper-solid into the three-dimensional space $t=0$. It will be denoted $S(W,\boldsymbol{A},\boldsymbol{B})$. Note that $S(W,\boldsymbol{A},\boldsymbol{B})$ is the infinite union of the W@$\boldsymbol{P}(t,\boldsymbol{A},\boldsymbol{B})$ instances for all values of $t\in[0,1]$.

3. Screw motions

Two arbitrary key-poses may be interpolated by a variety of motions. If no restriction are imposed on the type of motion, it may prove expensive to compute a boundary representation of $S(W,\boldsymbol{A},\boldsymbol{B})$ [Abdel-Malek98, Blackmore97, Blackmore99, Hu94a, Hu94b, Parida94, Martin90, Rossignac84, Kim04]. Many systems resort to a graphic superposition of the instances of W@$\boldsymbol{P}(t,\boldsymbol{A},\boldsymbol{B})$ for a finite set of samples of t or restrict the domain to polyhedra [Schroeder94, Rossignac01].

For more rigorous applications such as machining, the true boundary of the volume swept by a free-form solid may be required. To provide such a functionality, we propose a method for computing a representation of the precise boundary $\partial S(W,\boldsymbol{A},\boldsymbol{B})$ of $S(W,\boldsymbol{A},\boldsymbol{B})$ when W is a free-form solid and when $\boldsymbol{P}(t,\boldsymbol{A},\boldsymbol{B})$ is a screw motion, exploiting the properties of screw motions to simplify the underlying algorithm and to reduce its computational cost. We chose to focus on ***screw motions*** [Ohwovoriole81] for their simplicity, uniqueness, independence of coordinate system, and computational advantages, which are reviewed below. Screw motions extend the commonly used rotations and translations, while retaining their representational conciseness and specific geometric properties. Screw motions have been used in numerous applications, including collision prediction in manufacturing, robotics, or animation [Redon00, Redon01, Kim03, Buss05], and free-form shape deformations [Llamas03, Llamas05].

A ***polyscrew*** (i.e. piecewise-screw motion) [Powell05] can be specified in terms of a sequence of user-controlled ***key-poses***. Each pair of consecutive key-poses, $\boldsymbol{A}$ and $\boldsymbol{B}$, is interpolated by a screw-motion segment $\boldsymbol{P}(.,\boldsymbol{A},\boldsymbol{B})$ [Zefran98]. Linear (and even polynomial) mappings between the time parameter of each screw motion and the global time may be used to control the overall animation speed and acceleration. A minimal angle screw segment is uniquely defined by its two key-poses (except for ambiguities created when two consecutive poses correspond to a 180 degrees rotation) and is coordinate system independent. Furthermore, the motion is independent of construction steps through which the key-poses were specified. A polyscrew is a

continuous motion, but usually exhibits derivative discontinuities at the key-poses. To reduce these discontinuities, one may increase the sampling (number of key-poses) or perform ***polyscrew smoothing*** [Powell05] through an interpolating "Split&Tweak" subdivision process [Rossignac04], which introduces additional key-poses converging to a smooth motion which interpolates the initial **control** poses. Throughout this paper, we assume that a sufficiently dense set of key-poses has been obtained (either through sampling or smoothing) and focus on the computation of the volume swept by the solid as it moves along a screw motion segment that interpolates two such consecutive poses.

A screw motion segment $\boldsymbol{P}(t,\boldsymbol{A},\boldsymbol{B})$ is a constrained combination of two simultaneous motions: a linear ***translation*** by distance td along a vector $\mathbf{s}$ and a ***rotation*** by tb around a constant axis (***screw axis***) parallel to $\mathbf{s}$ and passing through an ***axis point*** $\mathbf{p}$. Note that these two transformations (rotation and translation) commute. Since a screw motion segment is defined by the four parameters: **direction** $\mathbf{s}$, axis point $\mathbf{p}$, ***total displacement*** d and ***total angle*** b, we denote it by $\boldsymbol{M}(\mathbf{s},\mathbf{p},d,b)$. Note that an infinite number of pairs $(\boldsymbol{A},\boldsymbol{B})$ of poses define the same screw motion segment $\boldsymbol{M}(\mathbf{s},\mathbf{p},d,b)$. Also note that $\boldsymbol{M}(\mathbf{s},\mathbf{p},d,0)$ is a pure translation and that $\boldsymbol{M}(\mathbf{s},\mathbf{p},0,b)$ is a pure rotation. Hence a screw motion generalizes these two simple motions.

A simple and efficient algorithm for computing the screw motion segment parameters ($\mathbf{s}$, $\mathbf{p}$, d and b) from a pair of initial and final poses ($\boldsymbol{A}$, $\boldsymbol{B}$) was described by Rossignac and Kim [Rossignac01] (Fig. 2). Let the matrices of the starting and ending poses be $[\ I_A\ J_A\ K_A\ O_A]$ and $[\ I_B\ J_B\ K_B\ O_B]$. To obtain the parameters of the interpolating screw, let $I:=I_B-I_A$, $J:=J_B-J_A$, $K:=K_B-K_A$, and $O:=O_B-O_A$. At least one of the three cross-products I×J, J×K, and K×I is not a null vector. Without loss of generality, assume that I×J is the longest of the three (otherwise rotate the symbols). Compute $\mathbf{s}:=I\times J$ and normalize it with $\mathbf{s}:=\mathbf{s}/\|\mathbf{s}\|$. Compute b as the angle between $\mathbf{s}\times I_A$ and $\mathbf{s}\times I_B$. Compute d as the dot product $O\bullet\mathbf{s}$. Finally, compute $\mathbf{p}:=(O_B+O_A+(\mathbf{s}\times O)/\tan(b/2))/2$. This construction ensures that d is positive and that b is between 0 and 180 degrees.

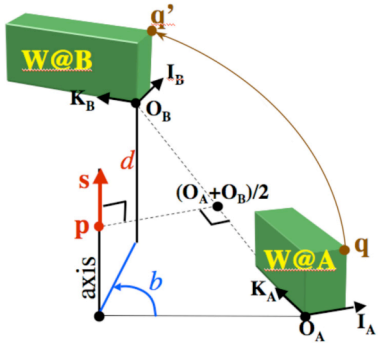

Figure 2: *Construction of the screw motion parameters* ***s***, ***p***, *d and b from the initial and final poses A and B.*

The velocity $\mathbf{v}(\mathbf{q})$ of a point $\mathbf{q}$ that is subject to a screw motion $\boldsymbol{M}(\mathbf{s},\mathbf{p},d,b)$ is the vector sum of a displacement by d along $\mathbf{s}$ and a displacement by a quantity br along the vector $\mathbf{t}$ that is tangent to the circle which is the planar projection along $\mathbf{s}$ of the helix upon which $\mathbf{q}$ travels. The quantity r is the radius of that circle, i.e., $r=\|\mathbf{hq}\|$, where $\mathbf{h} = \mathbf{p} + (\mathbf{pq} \bullet \mathbf{s})\ \mathbf{s}$ is the closest projection of $\mathbf{q}$ onto the screw axis. Thus, we have $\mathbf{v}(\mathbf{q})= d\mathbf{s} + br\mathbf{t}$. The vector $r\mathbf{t}$ is orthogonal to both $\mathbf{s}$ and $\mathbf{hq}$ and has for magnitude $\|\mathbf{hq}\|$. Consequently, $r\mathbf{t} = \mathbf{s} \times \mathbf{hq}$ and $\mathbf{v}(\mathbf{q}) = d\mathbf{s} + b\mathbf{s} \times \mathbf{hq}$.

4. Overview of the proposed approach

Our approach for computing $\partial S(W,\boldsymbol{A},\boldsymbol{B})$ and our contributions are based on the following series of observations expanded in the remainder of the paper. Note that (except for the last point in bullet G), these observations have been published and used previously. We review them here to justify the validity of the proposed approach.

A. The ***velocity*** $\mathbf{v}(\mathbf{q})$ of a point $\mathbf{q}$ of W is **constant** with respect to the object's local coordinate system. To be more precise, the velocity vector $\mathbf{v}(\mathbf{q})$ may be computed using the screw motion parameters for a point $\mathbf{q}$ on one instance of W. We can express its coordinates in the global coordinate system. These coordinates will of course change with time as W moves along the motion. We can also express the coordinates of vector $\mathbf{v}(\mathbf{q})$ in the local coordinate system of W, by simply computing its dot products with the basis vectors of that local coordinate system. These local coordinates remain constant throughout each screw segment of the motion. The simplicity, accuracy, and performance of the proposed approach are all based on this crucial observation. The ***trajectory*** of a point $\mathbf{q}$ in the global coordinate system is a ***helix arc*** defined as the set $\{\mathbf{q}@\boldsymbol{P}(t,\boldsymbol{A},\boldsymbol{B})$ with $t \in [0,1]\}$.

B. Consider a point $\mathbf{q}$ of the smooth portion of ∂W. Let $\mathbf{n}(\mathbf{q})$ be the normal to ∂W at $\mathbf{q}$. We say that $\mathbf{q}$ is an ***ingress points*** when $\mathbf{n}(\mathbf{q})\bullet\mathbf{v}(\mathbf{q})<0$, a ***grazing point*** when $\mathbf{n}(\mathbf{q})\bullet\mathbf{v}(\mathbf{q})=0$, and an ***egress point*** when $\mathbf{n}(\mathbf{q})\bullet\mathbf{v}(\mathbf{q})>0$ (see Fig. 3). Now consider a point $\mathbf{q}$ on a sharp convex edge $\mathbf{e}$ of ∂W. We say that $\mathbf{q}$ is a ***motion silhouette*** (or simply *silhouette*) point (with respect to the motion direction) if in an infinitely small neighborhood around $\mathbf{q}$ one of the faces incident on $\mathbf{e}$ contains egress points and the other ingress points. The ***generator*** $G(W,\boldsymbol{A},\boldsymbol{B})$ is the set of the grazing and silhouette points of ∂W. In general, it is composed of loops of **curves** that separate the egress from the ingress points. Occasionally, it may contain

two-dimensional regions or isolated points with tangential velocity. Since the velocity of a point is constant in the local coordinate system, G(W,***A***,***B***) remains fixed in the coordinate system of the solid throughout the motion.

C. The ***extruded set*** E(W,***A***,***B***) is the sweep G(W,***A***,***B***)@***P***(*t*,***A***,***B***). Each one of its faces is the extrusion of a curve ***R***(*u*) of G(W,***A***,***B***)) and hence may be expressed in parametric form F(*t*,*u*).

D. The ***candidate set*** C(W,***A***,***B***), defined as ∂(W@***A***) ∪ ∂(W@***B***) ∪ E(W,***A***,***B***) may be decomposed into a geometric complex [Rossignac89] K of two-dimensional **cells** (faces) by using surface/surface intersection tools available in commercial CAD systems. Note that C(W,***A***,***B***) contains the boundary of S(W,***A***,***B***), which is the union of a set J of the cells of K.

E. Because C(W,***A***,***B***)⊂S(W,***A***,***B***), the unwanted cells in the set I=K–J are **interior** to S(W,***A***,***B***). Hence the cells that form ∂S(W,***A***,***B***) may be generated by identifying and discarding all cells of K that are in the interior of S(W,***A***,***B***).

F. To test whether a cell **Q** lies in the interior of S(W,***A***,***B***) it suffices to generate an arbitrarily chosen ***witness point*** **q** in **Q** and to test whether the **back helix segment**, **q**@***P***(*t*,***B***,***A***) for *t*∈[0,1] intersects ∂W@***A*** (Fig. 3).

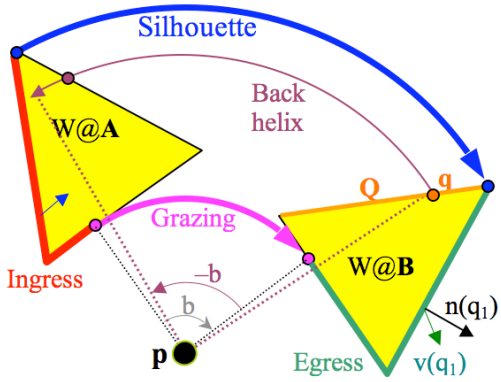

Figure 3: *In two dimensions, the boundary (thick curve) ∂S(W,A,B) of the swept region is composed of the egress points (green) of ∂W@**A**, of the ingress points (red) of ∂W@**B**, of the extrusions of silhouette vertices (blue) and of grazing points. We know that cell **Q** is inside S(W,A,B) because the back helix (magenta) starting at a witness point **q** of **Q** intersects the interior of W@**A**.*

Note that one could formulate S(W,***A***,***B***) as the regularized union of face-sweeps, each computed as the volume swept by a different face of ∂W. However, most of the faces bounding the individual face-sweeps are interior to S(W,***A***,***B***) and are trimmed away during the union. The approach we propose here avoids the unnecessary cost of generating and then trimming most of these faces.

Our swept-boundary evaluation algorithm performs the five steps listed below. We discuss the underlying derivations and their implementation in the following sections and show results at the end of the paper.

(1) We construct the curves of the generator G by trimming convex edges to their silhouette portion and by fitting spline curves through series of consecutive grazing points in each face of W.

(2) We extrude each curve of G along ***P***(*t*,***A***,***B***) obtaining a parametric face of the extruded set E.

(3) We consider the set C of faces of E, of ∂W@***B***, and of ∂W@***A***. We decompose each face F of C into maximally connected cells bounded by the intersection curves between F and the other faces of C.

(4) We discard each cell **Q** for which the helix **q**@***P***(.,***A***,***B***) from an arbitrary witness point **q** in **Q** intersects W@***B***.

(5) We combine the remaining cells to produce ∂S(W,***A***,***B***)

5. Grazing points, characteristic curves, and envelopes

Consider the general case of a **time-evolving** (i.e. deforming) **smooth surface**, U(*t*). At any moment *t*, its ***characteristic curve*** is the limit of U(*t*)∩U(*t*+*ε*) as *ε* tends towards zero. The ***envelope*** of the family U(.) is the union of the continuous set of these characteristic curves. Note that in general, the shape of these characteristic curves changes with time. Analytical formulation of envelopes has been studied for a variety of families of shapes. These include the canal surfaces of families of spheres [Monge1850, Rossignac85].

We are concerned with **rigid bodies**. The characteristic curves of rigid bodies under a continuous motion are the maximally connected one-dimensional sets of the grazing points. Their nature depends on the type of motion and on the nature of the surfaces that bound the moving object. The envelopes swept by these characteristic curves may be formulated using envelope theory [Wang86], differential equations [Blackmore92b, Blackmore99, Ganter93], and Jacobian Rank Deficiency [AbdelMalek97, AbdelMalek98]. Together with the boundaries of the initial and final instances, these envelopes form a set

that is guaranteed to contain the boundary of the volume swept by a smooth solid. Several practical techniques were introduced to compute such envelopes and to trim them to the boundaries of swept volumes of moving objects [Blackmore92, Blackmore99, AbdelMalek98, Yang04]. They are reviewed by Blackmore et al. [Blackmore97].

During general motions, the topology and shape of the characteristic curves changes over time. For **screw motions** however, they remain fixed on the object, and hence may be pre-computed.

First, we review the computation of the characteristic curve for a **planar face** in screw motion by $\boldsymbol{M}(\mathbf{s},\mathbf{p},d,b)$ [Weld90, Rossignac01]. Consider a face **F** of $\partial\mathbf{W}@\mathbf{A}$ with a normal direction **n**. When $\mathbf{n}\times\mathbf{s}=0$, no characteristic curve exists on **F**. Hence, we will now assume that **n** is not parallel to **s**. A point **q** of **F** is grazing if $\mathbf{n}\cdot\mathbf{v}(\mathbf{q})=0$, which yields $d\mathbf{n}\cdot\mathbf{s}=b\mathbf{n}\cdot(\mathbf{s}\times\mathbf{q}\mathbf{h})$ and

$$b(\mathbf{s}\times\mathbf{n})\cdot\mathbf{hq} = d\mathbf{s}\cdot\mathbf{n} \qquad [1]$$

Note that points **q** that satisfy equation [1] lie on plane **U** having normal $\mathbf{n}\times\mathbf{s}$ and passing through **p**. Hence, the characteristic curve is the line segment $\mathbf{L}=\mathbf{F}\cap\mathbf{U}$. It may be efficiently computed as a plane/face intersection. We avoid the cost of this process when the four vertices of a rectangle that contains **F** all lie on the same side of **U**. The characteristic curve is shown (the left of Fig. 4) for a polyhedron in screw motion.

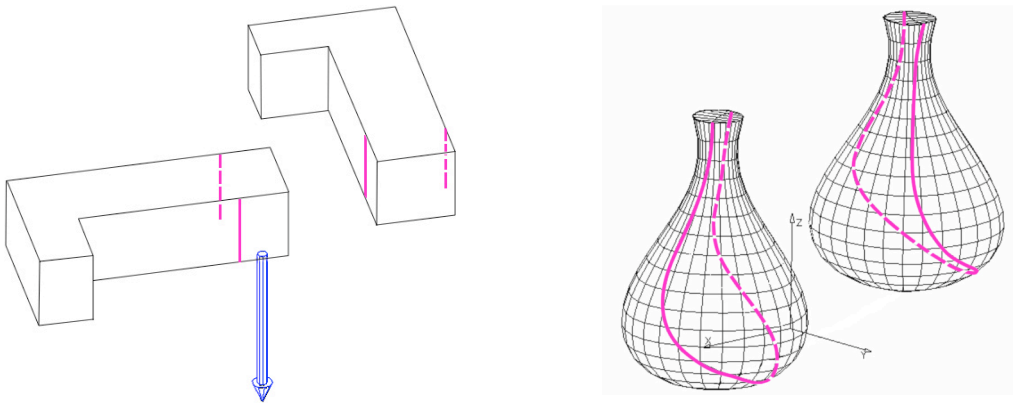

Figure 4: *The characteristic curves are drawn in magenta on the planar faces (left) and on curved faces of a solid (right). Both screw motions have a vertical axis (not shown for the motion on the right, which has a small rotation angle b).*

Now, let us turn to the computation of the characteristic curves on **curved surfaces.** A closed form solution for the restricted set of **natural quadric** surfaces undergoing an instantaneous screw motion was proposed by Hu and Ling [Hu94, Hu94b]. They have further combined this technique with the local approximation of an arbitrary motion by an **instantaneous screw motion**. At each time step, they compute an instantaneous screw motion approximation from the velocities and positions of three vertices, then they use the approximating screw motion to compute the characteristic curves on the faces and edges of the object in its current position, and finally produce a polygonal surface that interpolates two consecutive characteristic curves. This interpolation may prove delicate, since characteristic curves may have different topologies. The resulting polygonal surface is then trimmed to the part that bounds the swept volume by computing a series of cross-sections, performing a 2D selection of bounding curves, and then interpolating the resulting 2D contour by a triangular surface. The result is a *polyhedral approximation* of the boundary of $S(W,\boldsymbol{A},\boldsymbol{B})$. To complement this prior art, we propose an approach for computing the grazing points on polynomial parametric patches undergoing a screw motion segment and for interpolating them with a spline curve (the right of Fig. 4).

When the face F is curved, the normal **n** varies with **q** and thus will be denoted $\mathbf{n}(\mathbf{q})$. The set of points **q** with velocity orthogonal to $\mathbf{n}(\mathbf{q})$ is the surface defined by $\mathbf{n}(\mathbf{q})\cdot\mathbf{v}(\mathbf{q})=0$. Developing this equation yields

$$b(\mathbf{n}(\mathbf{q})\times\mathbf{s})\cdot\mathbf{qp} = d\mathbf{n}(\mathbf{q})\cdot\mathbf{s} \qquad [2]$$

Consider a parametric surface F(u,v). For example, F(u,v) may be a bi-cubic Bezier or a bi-quintic polynomial patch. We treat it as a family of iso-parameter curves, $C_v(u)$. We also define the field $\mathbf{t}_v(u)$ of surface tangent bi-normal to C_v and the field $C'_v(u)$ of tangents to C_v. For a given value of v, we substitute **q** by $C_v(u)$ and $\mathbf{n}(\mathbf{q})$ by $\mathbf{t}_v(u)\times C'_v(u)$ in [2] and solve for u, obtaining zero or more solutions u_i and hence corresponding grazing points $C_v(u_i)$. We repeat this for a dense set of v values and then switch the role of u and v and do this again. Hence, we have decomposed the parameter space of F into square cells, each bounded by two horizontal (constant u) and two vertical (constant v) edges. We split cells into 4 quadrants recursively until none of their edges contain more than one grazing point (Fig. 5) or until a prescribed resolution is reached. To split a cell, we create mid-parameter curve segments in u and v and solve [2] to compute the grazing points they contain. Then, we order the grazing points along the characteristic curves by following them in parameter space from one cell to an adjacent cell, in a manner similar to the surface/surface intersection solution proposed for the set of natural quadrics and tori [Rossignac85]. We fit a spline curve in 3D through each run of ordered points and use it as an approximation of the corresponding characteristic curve. If face F is a trimmed subset of a complete patch, we use the trimming curves in the parametric domain to trim the characteristic curve.

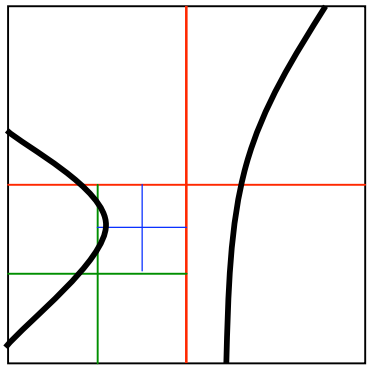

***Figure 5**: Each cell in the parametric domain of a patch is split recursively until no edge of a cell contains more than one grazing point.*

6. Silhouette edges

In this section, we focus on the identification of the silhouette portions of the sharp edges of a solid in screw motion. Before we define sharp edges, consider an infinitely small tube **T** around the interior of an edge **E** of **W**. Let $\mathbf{N}=(\mathbf{T}\cap\partial\mathbf{W})-\mathbf{E}$. We say that the connected components of N are the ***face-borders*** incident upon **E**. Because **W** is a solid, **E** has an even number, $2k$, of borders. **E** is ***manifold*** when k=1 and ***non-manifold*** otherwise. It is possible to treat a non-manifold edge with k pairs of incident face-borders as k coincident manifold edges [Rossignac99] by matching face-borders into pairs of compatible orientation. Hence, we will now assume that such a matching has been performed and that all edges have exactly two incident face-borders. Note that although unusual, these two face-borders may belong to the same face.

Consider a point **q** in the interior segment of an edge **E**. Let **t** denote the tangent to **E** at **q**. Let $\mathbf{n}_L$ and $\mathbf{n}_R$ denote the limits of the outward pointing normals to points on the two face-borders (L and R) as they converge to **q**. Furthermore, assume that the binormal $\mathbf{b}=\mathbf{n}_L\times\mathbf{t}$ points towards the interior of face-border L. We say that **E** is **convex** at **q** if $\mathbf{b}\bullet\mathbf{n}_R>0$, **flat** if $\mathbf{b}\bullet\mathbf{n}_R=0$, and **concave** if $\mathbf{b}\bullet\mathbf{n}_R<0$. We assume that all sharp curves of $\partial\mathbf{W}$ have been split into edges at points where the number of face-borders or the convex/flat/concave classification changes.

Note that concave edges cannot contribute to $\partial S(W,\boldsymbol{A},\boldsymbol{B})$ and that grazing points and characteristic curves that happen to coincide with flat edges are computed using the technique described in the previous section.

Let $\mathbf{v}(\mathbf{q})$ denote the velocity of **q**. We say that **q** is a ***silhouette point*** if $(\mathbf{v}(\mathbf{q})\bullet\mathbf{n}_R)(\mathbf{v}(\mathbf{q})\bullet\mathbf{n}_L)\le 0$. As **q** slides along **E**, its status (silhouette or not) changes at places where $\mathbf{v}(\mathbf{q})\bullet\mathbf{n}_R=0$ or $\mathbf{v}(\mathbf{q})\bullet\mathbf{n}_L=0$. In each one of these equations, we substitute $\mathbf{v}(\mathbf{q})$ with $d\mathbf{s}+b\mathbf{s}\times\mathbf{hq}$ and the normal $\mathbf{n}_R$ or $\mathbf{n}_L$ with the biparametric expression of $\mathbf{t}_v(u)\times\mathbf{t}_u(v)$, using the (u,v) parameters of **q** in the corresponding patch. We solve the equations, compute the corresponding points and use them to split **E** into segments that are silhouette or not silhouette. The silhouette edges are shown (Fig. 6) for a polyhedron and a curved object in screw motion.

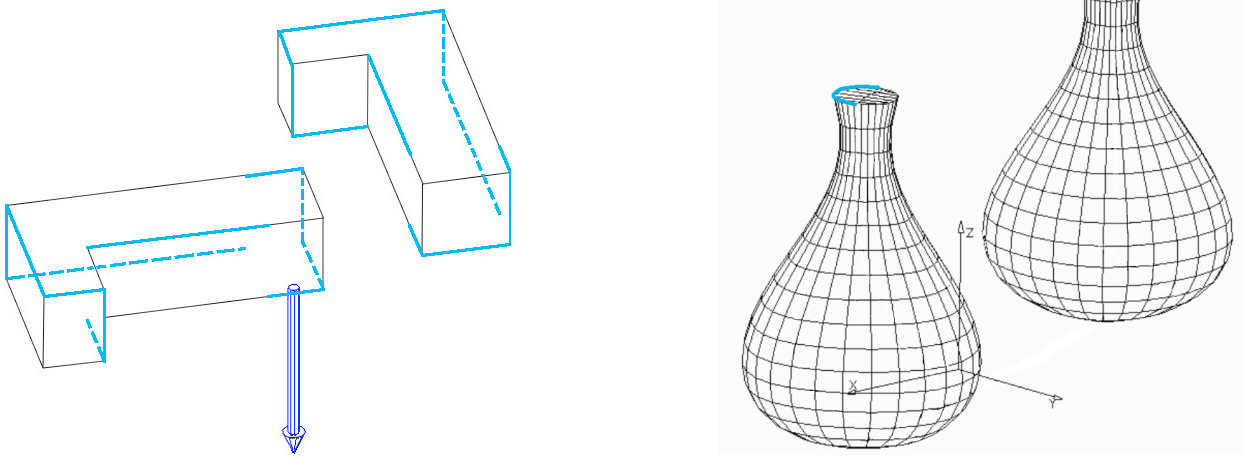

***Figure 6**: The silhouette edges are drawn in cyan on the polyhedron (left) or on the free-form shape (right). Note that the concave edges are not included in the silhouette edges.*

7. Extrusions

$\partial S(W,\boldsymbol{A},\boldsymbol{B})$ is a subset of the union of 4 sets: (1) the egress points of $\partial W@\boldsymbol{B}$, (2) the ingress points of $\partial W@\boldsymbol{A}$, (3) the faces swept by the characteristic curves, and (4) the faces swept by the silhouette edges.

We split ∂W into its egress and ingress parts using the all the characteristic curves and silhouette edges computed as explained above. The algorithm used for splitting the boundary if W is similar to the one, discussed in the following section, for computing cells of the arrangement. Then, we apply the B pose to the egress points and the ***A*** pose to the ingress points, obtaining the first two sets (Fig. 7).

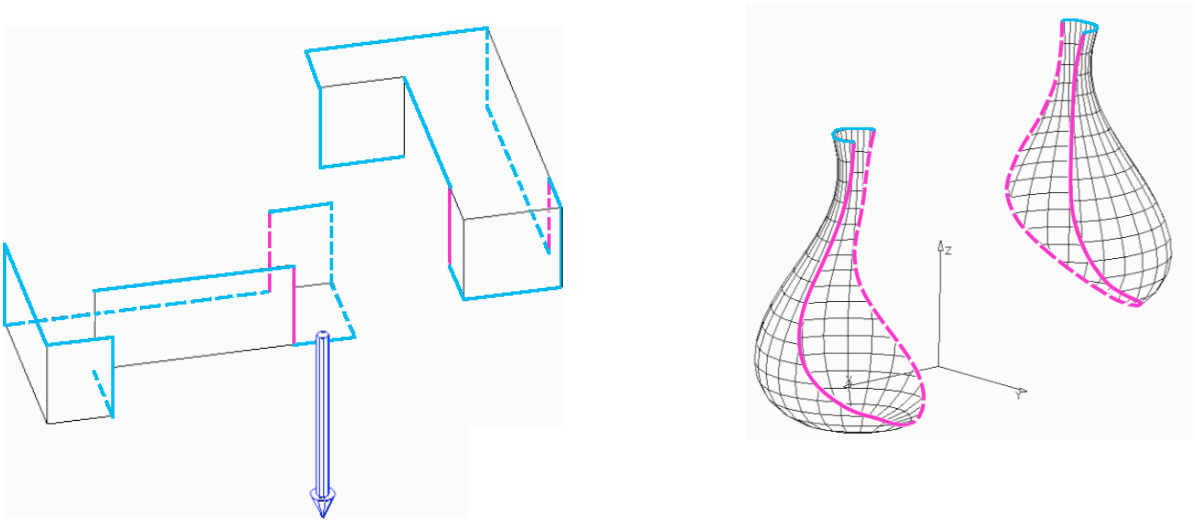

Figure 7*: The boundary of each solid is split along the characteristic curves and silhouette edges into its ingress or egress parts, which are placed at the initial and final poses respectively.*

When a characteristic curve or silhouette edge is a straight line segment, its extrusion along a screw motion is a ruled surface (Fig. 8, left) for which we can obtain a closed form expression. However, for the sake of simplicity, we ignore these special cases and convert them to splines and treat them as general curves when computing the sweep. To extrude a general curve G along a screw motion segment, we sweep the control points of its approximation. Each control point travels along a helix. We approximate this helix with a spline. Hence, we obtain a two-dimensional array of control points for each extruded face. We use this control grid to define a parametric patch that approximates the extruded face (Fig. 8, right).

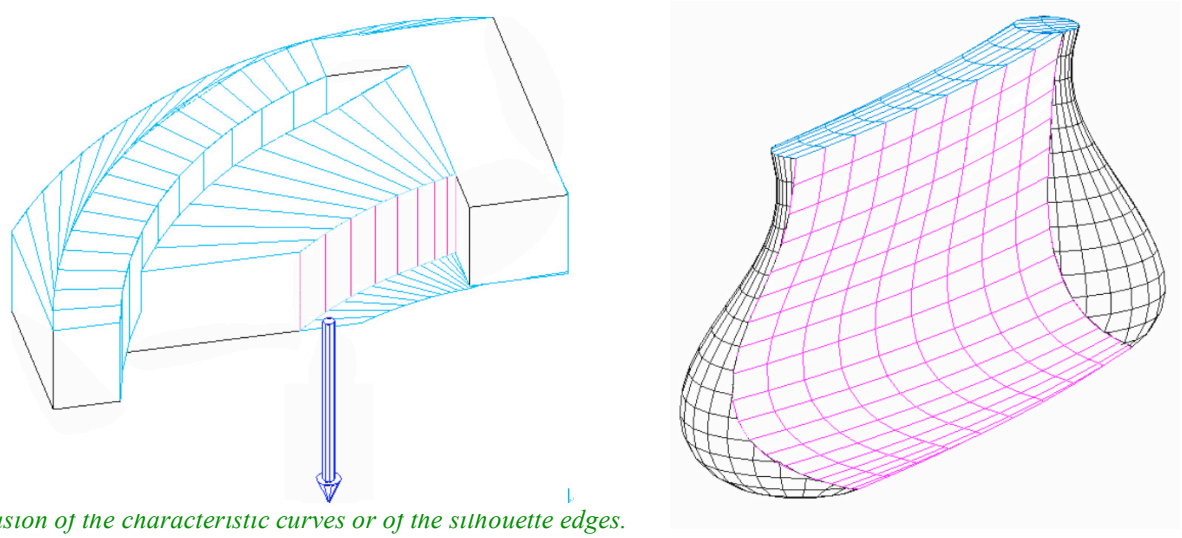

Figure 8*: Extrusion of the characteristic curves or of the silhouette edges.*

8. Subdivision

To subdivide each extrusion face and each face of $\partial W@\boldsymbol{A}$ and $\partial W@\boldsymbol{B}$, we use commercially available tools for computing intersections of parametric patches. These tools report not only the curves where two faces intersect, but also the curves where the boundary of one face intersects another face. We compute all the pair-wise intersections (Fig. 9). The resulting intersection curves are represented as trimming curves in the parametric domain of each patch. Then, we combine them with the initial

trimming curves of each face and compute the connected component (two-cells) of the complement of the union of these curves in the parameter domain. We call these relatively open components the ***cells*** of the subdivision of the face. We compute the boundaries of these cells by walking along the curves in parametric space, never crossing a curve. A good metaphor to understand this process is to think of the intersection curves as streets and to walk all the sidewalks never crossing a street. Each walk is a circuit and defines a bounding loop of a cell. Note that it is possible for a cell to have more than one bounding loop. We say that it is a cell with one or more holes.

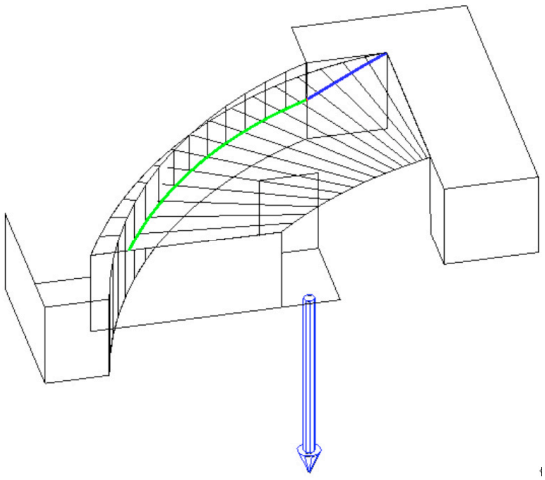

Figure 9: *A "staircase" ruled face is swept by the silhouette portion of an edge. It is split in two cells by a curve that is composed of its intersection (blue) with a square face and of its intersection (green) with a portion of a nearly cylindrical "wall" that follows the staircase.*

Note that a cell either lies entirely on $\partial S(W,\boldsymbol{A},\boldsymbol{B})$ or entirely in the interior of $S(W,\boldsymbol{A},\boldsymbol{B})$. In the next section, we explain how to discard cells that lie in the interior of $S(W,\boldsymbol{A},\boldsymbol{B})$.

9. Trimming

To test whether a particular cell **Q** lies inside $S(W,\boldsymbol{A},\boldsymbol{B})$, we generate an arbitrary **witness point q** in the cell by shooting an inward ray from the midpoint of one of its edges in the parametric domain. If **q** lies in the interior of $S(W,\boldsymbol{A},\boldsymbol{B})$, we conclude that **Q** does too and hence discard it. Unfortunately, we cannot directly test whether **q** lies in the interior of $S(W,\boldsymbol{A},\boldsymbol{B})$ because we do not have an explicit representation of $S(W,\boldsymbol{A},\boldsymbol{B})$. Hence, we use a new approach for *point-in-sweep* testing (Fig. 10), which we call **helix-shooting**, in analogy to the standard ray-shooting approach for point-in-shape testing.

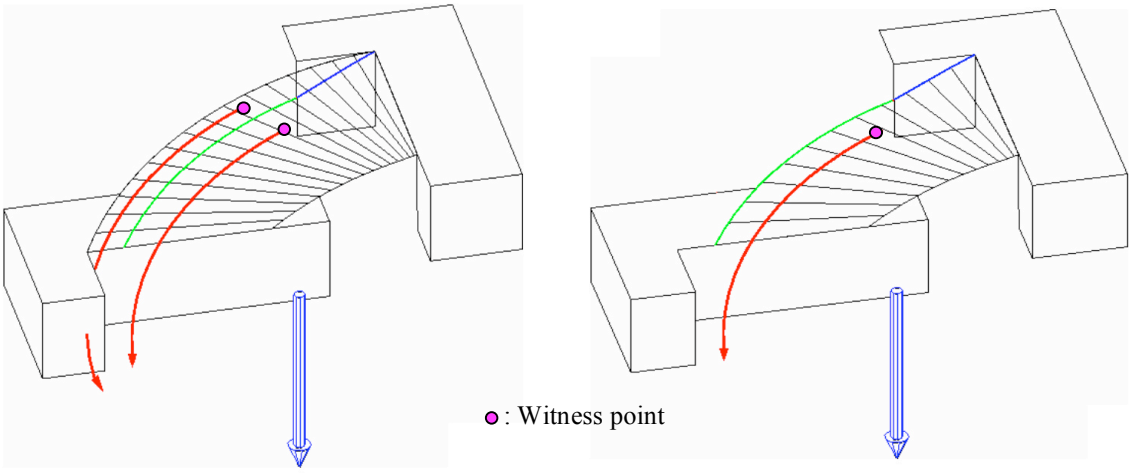

Figure 10: *The extrusion face is subdivided into the two cells (left) by the two intersection curves connected (green and blue). Then the point-in-swept test is applied to the two cells by shooting helices from a witness point of each cells. The figure (left) The helix shot from the upper cell intersects the initial instance (left), which implies that the cell lies in the interior of the swept volume and therefore must be discarded. The helix shot from the other cell misses the initial instance, indicating that the cell is on the boundary of the swept volume (right).*

Our *point-in-sweep* test, is based on the observation that **q** lies in the interior of $S(W,\boldsymbol{A},\boldsymbol{B})$ if and only if the back back helix

q@*P*(.,***B***,***A***) intersects the interior of W@***A***. To justify this result, note that **q** lies in the interior of S(W,***A***,***B***) if W collides with **q** during the motion. In the local coordinate system moving with W, point **q** follows the backward motion ***P***(.,***B***,***A***). Hence, we can test for collision in the local coordinate system of W@A by testing whether the back helix arc swept by **q** hits the boundary of W@***A***. To do this, we approximate the back helix **q@*P***(.,***B***,***A***) by a spline curve and use commercially available tools for testing whether that helix intersects W@A.

10. Summary of the algorithm

We have described above the underlying principles and the details of each step. For clarity, we summarize in this section the algorithmic steps of our approach.

1) Given a solid W, its initial pose A, and its final pose B for a portion of the motion of W, we compute the parameters of the **screw** taking A to B.
2) For each patch P bounding W, we compute, in the parametric domain of P, a parametric approximation of the **characteristic curves** in P, defined as the set of points where the instantaneous velocity vector is tangent to P.
3) For each edge E bounding W, we compute the **motion silhouettes** (edge-segments that separate the ingress and egress portions of the incident patches).
4) We extrude the characteristic curves and edge segments along the screw and obtain a set of **candidate** parametric patches.
5) We add the **boundary** of W at pose A and at pose B to the candidate set, which now is guaranteed to contain the boundary of the swept volume.
6) We split each patch of the candidate set into **cells** using the intersection curves of that patch with all other patches in the candidate set.
7) We **discard** cells that lie in the **interior** of the swept volume. To test each cell, we shoot a screw segment from it and check whether the segment hits W at pose A.

10. Implementation details

We discard cells that are in the interior of the swept volume by shooting a screw segment from an arbitrary point in each cell and testing whether the screw hits W at pose A.

We have implemented the algorithm described here in C using the API of CATIA [CATIA97a, CATIA97c]. The 3D images shown in the paper were generated using CATIA viewing facilities [CATIA97b]. The initial free-form shapes were bounded by bi-quintic Bezier patches. The original sampling (before recursive refinements) splits the parameter domain of each patch into 20x20 squares. We formulated the normal field of these patches as a patch of degree 9 in u and v. We used splines (piecewise quintic polynomials) to interpolate the characteristic curves and the helices. To compute grazing points, we need to find the roots of a 14^{th} degree polynomial, which we do using Maple [Maple]. The algorithm takes an average of 8 seconds on an IBM RS/6000 workstation for each one of the sweeps generated of the simple objects used for illustration in the paper.

Fig. 11 shows the egress points and ingress points as well as the extrusion faces for the free-from objects of Fig. 1. Fig. 12 illustrates the application of the algorithm to a NC simulation. It shows the initial and final poses of a square-shaped cutter, and the swept volume generated between the two poses. Since the swept volume is represented by a valid boundary model, it can be used as an argument of a Boolean operation to compute the material removed by the cutter.

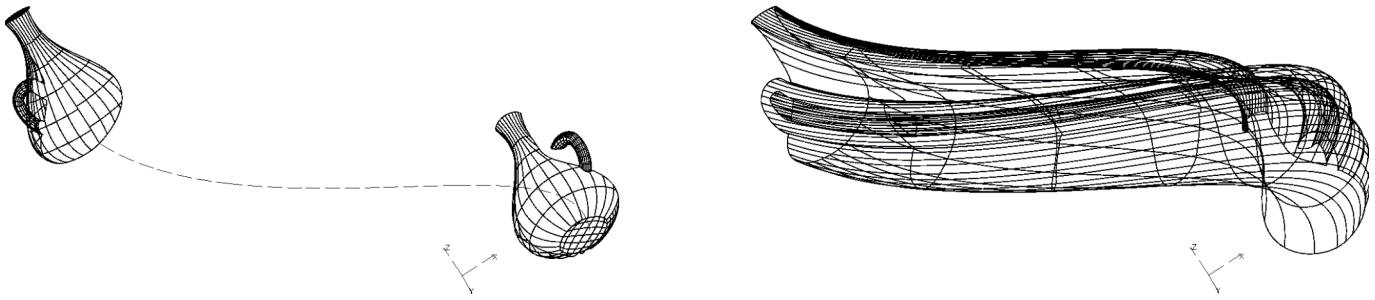

Figure 11*:. The egress points and ingress points of the curved object of Figure 1 (left). The extrusion faces by sweeping the characteristic curves and the motion silhouette edges (right).*

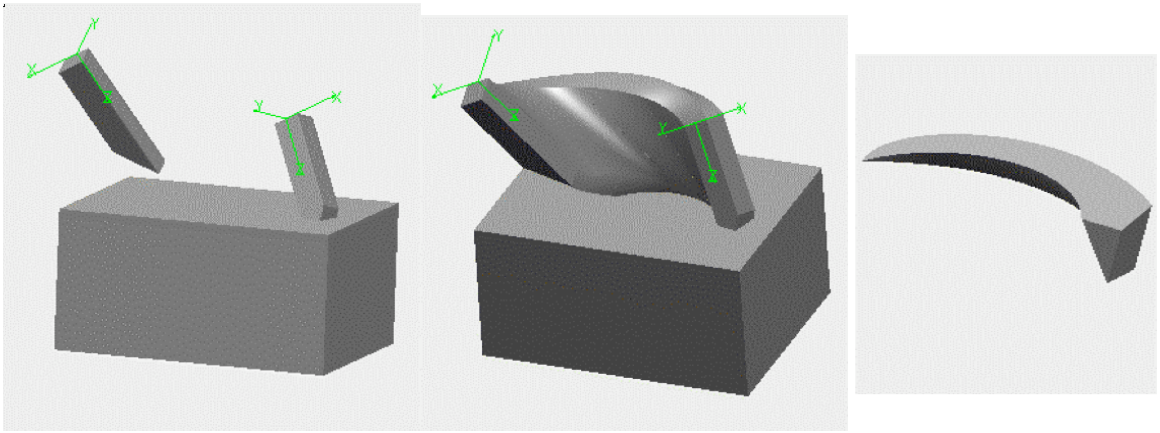

***Figure 12**: From left to right: The initial and final poses of the square cutter and the material. The volume swept by the cutter in screw motion. The material removed by the cutter computed as a Boolean intersection .*

12. Summary and conclusion

We present an algorithm for computing the boundary of the volume swept by a free-form solid in screw motion. The algorithm exploits the fact that the characteristic curves and the motion silhouette edges are fixed on the moving solid. It extrudes these curves to produce a set of faces, which, along with the faces of the initial and final instances, are subdivided at their intersections into cells. Each cell is tested by shooting a helix segment from an arbitrary point in the cell. If the helix segment hits the final instance, the cell is discarded. Our implementation supports solids bounded by parametric patches and approximates extruded faces by parametric patches and helices by parametric splines. It uses the tools provided in CATIA for computing intersections.

References

[Abdel-Malek97] Abdel-Malek, K. and Yeh, H.J., Geometric Representation of the Swept Volume Using Jacobian Rank-Deficiency Conditions, Computer Aided Design, 1997, Vol. 29, No. 6, pp. 457-468.

[Abdel-Malek97] Abdel-Malek, K., Yeh, H. J. and Othman, S., Swept volumes: void and boundary identification. Computer-Aided Design, 1998, 30(13), 1009-1018.

[Blackmore92a] Blackmore, D., Leu, M., Wang, L. and Jiang, H., Analysis of swept volumes via Lie groups and differential equations. Int, J. of Robotics Research, 1992, vol. 11, 516-537.

[Blackmore92b] Blackmore, D., Leu, M., Wang, L., Applications of flow and envelopes to NC machining. Annals of CIRP, 1992, Vol. 41, 493-496.

[Blackmore97] Blackmore, D., Leu, M., Wang, L. and Jiang, H., Swept volumes: a retrospective and prospective view. Neural Parallel and Scientific Computations, 1997, Vol. 5, 81-102.

[Blackmore99] Blackmore, D., Samulyak, R. and Leu, M., Trimming swept volumes. Computer-Aided Design, 1999, 31(3), 215-214.

[Buss05] Samuel Buss. Collision Detection with Glide Rotations. The Visual Computer 21 (2005) 41-58.

[CATIA97a] CATIA Advanced Surface Design API Reference Manual, Dassault Systemes, 1997

[CATIA97b] CATIA 3D Wireframe API Reference Manual, Dassault Systemes, 1997

[CATIA97c] CATIA API Reference Manual, Dassault Systemes, 1997

[Chiou04] Chiou, John., Accurate tool position for five-axis ruled surface machining by swept envelope approach, Computer-Aided Design, 2004, 36(10), 967-974

[ElMounayri98] El Mounayri, H.; Spence, A.D.; Elbestawi, M.A., 1998, Milling process simulation - a generic solid modeller based paradigm", Journal of Manufacturing Science and Engineering, Transactions of the ASME, 1998, vol. 120, no. 2, pp. 213-221.

[Ganter93] Ganter, M A, Storti, D W, and Ensz, M T 'On Algebraic Methods for Implicit Swept Solids with Finite Extent' ASME DE Vol 65(2) (1993) pp389-396.

[Hu94a] Hu, Z. and Ling, Z., Swept volumes generated by the natural quadric surfaces. Computers & Graphics, 1994, Vol. 20, No. 2, 263-274.

[Hu94b] Hu, Z., and Ling, Z. Generating swept volumes with instantaneous screw axes", Proc. 94 ASME Design Technical Conference, Part 1. Minneapolis, MN. 70(1), 7-14, 1994.

[Keiffe91] Keiffe, J. and Litvin, L., Swept volume determination and interference of moving 3-D solids, ASME J. of Mechanical Design, 1991, vol. 113, 456-463.

[Kim03] ByungMoon Kim and Jarek Rossignac (2003). **Collision Prediction for Polyhedra under Screw Motions.** *ACM Symposium in Solid Modeling and Applications*, pp. 4-10.

[Kim04] Kim, Y. J., Varadhan, G., Lin, M. C. and Manocha, D., Fast swept volume approximation of complex polyhedral models, Computer-Aided Design, 2004, 36(11), 1013-1027

[Llamas03] Ignacio Llamas, ByungMoon Kim, Joshua Gargus, Jarek Rossignac, and Chris D. Shaw (2003) **Twister: A space-warp operator for the two-handed editing of 3D shapes.** *ACM Trans. on Graphics (TOG), Proc. ACM SIGGRAPH*, 22(3), pp. 663-668.

[Llamas05] Ignacio Llamas, Alex Powell, Jarek Rossignac, and Chris Shaw (2005). **Bender: A Virtual Ribbon for Deforming 3D Shapes in Biomedical and Styling Applications.** *ACM Symposium on Solid and Physical Modeling (SPM),* pp. 89-99.

[Maple] http://www.maplesoft.com/

[Martin90] Martin, P. R. and Stephenson, P. C., Sweeping of three-dimensional objects. Computer-Aided Design, 1990, 22(4), 223-234.

[Monge1850] Monge, G. Applications de l'analyse a la geometrie, Paris, Bachelier, 5th edition, 1850.

[Ohwovoriole81] M. Ohwovoriole and B. Roth. An extension of screw theory. Transaction of ASME Journal of Mechanical Design, 103:725–735, 1981.

[Parida94] Parida, L and Mudur, S P 'Computational Methods for Evaluating Swept Object Boundaries' Visual Computer Vol. 10(5) (1994) pp. 266-276.

[Powell05] Alex Powell and Jarek Rossignac (2005). **ScrewBender: Polyscrew Subdivision for Smoothing Interpolating Motions.** GVU Technical Report GIT-GVU-05-05.

[Redon00] S. Redon, A. Kheddar, and S. Coquillart. An algebraic solution to the problem of collision detection for rigid polyhedral objects. In Proceedings of IEEE International Conference on Robotics and Automation, pages 3733–3738, April 2000.

[Redon01] S. Redon, A. Kheddar, and S. Coquillart. Contact: arbitrary in-between motions for continuous collision detection. In Proceedings of IEEE ROMAN, September 2001.

[Rossignac01] Rossignac, J. and Kim, J., Computing and visualizing pose-interpolating 3D motions, , Computer Aided Design, 2001, 33(4), pp. 279-291.

[Rossignac04**]** Jarek Rossignac (2004). **Education-Driven Research in CAD.** *Computer-Aided Design Journal (CAD),* Volume 36, Issue14. pp 1461-1469.

[Rossignac84] Rossignac, J. and Requicha, A. A. G., Constant radius blending in solid modeling. Computers in Mechanical Engineering, July 1984, 65-73.

[Rossignac85**]** Jarek Rossignac (1985). **Blending and Offsetting Solid Models.** PhD Dissertation, Electrical Engineering Department, University of Rochester, NY, June 1985.

[Rossignac89**]** Jarek Rossignac and Michael O'Connor (1989). **SGC: A Dimension-Independent Model for Pointsets with Internal Structures and Incomplete Boundaries**. In *Geometric Modeling for Product Engineering,* Proceedings of the IFIP Workshop on CAD/CAM, Eds. M. Wosny, J. Turner, K. Preiss, North-Holland, pp. 145-180, 1989.

[Rossignac99] Rossignac, J. and Cardoze, Matchmaker: Manifold BReps for non-manifold r-sets. Proceedings of the ACM Symposium on Solid Modeling, pp. 31-41, 1999.

[Sambandan89] Sambandan, K and Wang, K K, Five-axis Swept Volumes for Graphic NC Simulation and Verification, ASME DE, vol. 19(1) (1989) pp143-150.

[Schroeder94] Schroeder, W J, Lorensen, W E, and Linthicum, S, Implicit Modeling of Swept Surfaces and Volumes, Proceedings of the IEEE Visualization Conference, Los Alamitos, CA (1994) pp40-45.

[Wang86] Wang, W.P. and Wang, K. K., Geometric Modeling for Swept Volume of Moving Solids, IEEE Computer Graphics and Applications, 1986, 6(12), 8-17.

[Weinert04] Weinert, K., Du, S., Damm, P. and Stautner, M., Swept volume generation for the simulation of machining processes, International Journal of Machine Tools & Manufacture, 2004, 44(6), 617-628.

[Weld90] Weld J., and Leu M., Geometric representation of swept volumes with applications to polyhedral object. Int. J. Robotics Res, 1990; 9, 105-106.

[Yang05] Yang, J. and Abdel-Malek, K., Approximate swept volumes of NURBS surfaces or solids, Computer Aided Geometric Design, 2005, 22(1), 1-26.

[Zefran98] M. Zefran and V. Kumar. Interpolation schemes for rigid body motions. Computer-Aided Design, 30(3):179–189, 1998.