Survey Paper and Thesis Proposal
Approximate Boolean Operations for Subdivision Surfaces

Henning Biermann[*]
Advised by Denis Zorin[†]

1 Abstract

Subdivision representations of surfaces are convenient for many common tasks such as rendering, topology-preserving editing and simulation.

However, some important editing operations modify the topology of surfaces, for example, computing a union, intersection or difference of solids represented by closed surfaces. In general, it is impossible to represent the resulting surface as a subdivision surface precisely. Nevertheless, it is possible to find a subdivision surface which approximates the result sufficiently well. This approach is attractive for computer graphics applications, as high accuracy is often less important than high efficiency.

In this talk, we describe how to approximate the result of a boolean operation applied to a pair of solids represented by closed subdivision surfaces. We present algorithms for generating a control mesh for a subdivision surface approximating the result, and for fitting the surface defined on the new mesh to the geometry of the original surfaces.

Our algorithms aim to minimize the size of the new control mesh, and to control the valence of the newly created vertices. All algorithms are quite fast as they modify the control meshes only in a neighborhood of the intersection. Our algorithms can be applied to subdivision-based multiresolution surfaces, for which it is possible to trade performance for accuracy of the approximation.

2 Introduction

Subdivision surfaces are rapidly becoming popular in computer graphics. This surface representation is so attractive as it leads quite natural to multiresolution techniques which allow to interact with a geometric model on various levels of resolution. Over the last years, subdivision surfaces have been used in different contexts: rendering, multiresolution editing [56], animation [11] and simulation [17]. Also, a number of commercial systems use subdivision as a surface representation: Alias|Wavefront's Maya, Pixar's Renderman, Nichimen's Mirai, and Micropace' Lightwave 3D, to name just a few.

However, the creation of a complex surface remains a challenging task. Typically, the surface is manipulated via the control mesh (i.e. Maya). Designers move control points or manipulate the mesh by adding or removing polygons - a tedious task even for simple surfaces. As designers work on the control mesh rather than the surface they have to know about the properties of the particular

---

[*] `biermann@mrl.nyu.edu, www.mrl.nyu.edu/biermann/research`

[†] `dzorin@mrl.nyu.edu`

subdivision scheme. Typically, the quality of the surface depends strongly on what polygons are used and how they are connected. Typically, a subdivision scheme works best on a certain regular topology.

Nevertheless, Pixar created and animated subdivision surfaces for their beautiful short fim “Geri’s Game” [11]. Subdivision surfaces are particularly appealing for animations as one can model complex surfaces as a single piece without the danger that the model falls apart [37].

Some recent work aims at more intuitive user interfaces for surface editing. Igarashi [25] presents an interface for surface sketching. His algorithm infers a three dimensional shape form a two dimensional outline, editing operations (e.g. extrusion, indents) are indicated with mouse gestures.

The Skin project [38] has a similar objective, but proposes a different editing paradigm: The idea is to manipulate a surface which grows smoothly over underlaying masses, henceforth, the name Skin. The skin is interactively maintained as an evolving triangulation, which approximates an offset surface. The update rules for the triangulation employ some interesting heuristics to maintain an uniform edge length and to avoid skinny triangles.

2.1 Constructive Solid Geometry.

Many important editing operations can be expressed in terms of boolean operations on closed objects. Shapes are combined to form union, intersection or difference. For example, a union operation merges shapes; the difference operation scoops out a hole.

In Constructive Solid Geometry (CSG) [21], the user constructs objects by applying affine transformations and boolean operations to solids. Typically, a system provides basic geometric primitives such as box, sphere, cylinder and torus. Each object remembers its boolean expression, and the user can modify the expression anywhere.

While conceptually easy, a CSG system has to deal with many problems. As objects are represented as boolean expression, the boundary is defined only implicitly. However, for many tasks (i.e. rendering, collision, . . .) it is crucial to efficiently determine the boundary.

Due to its powerful editing primitives CSG is popular for professional geometric design, where specialized hardware is available. Only recently volume graphics hardware has become affordable for many users [40]. Still, the crucial intersection step for CSG in inherently ill-conditioned and difficult to implement. There are approaches using exact or precise arithmetic, but there is not much hope of an efficient system [53, 50].

The work of Rappoport et al. [42] is concerned with an CSG implementation on commodity graphics hardware. A CSG object is represented as a *Convex Differences Aggregate*, a set of nested negative or positive convex cells. These aggregates can be efficiently rendered using the screen buffer: An object with a hole would be rendered as a positive cell, where one masks out the negative cell representing the hole. This approach works works only with polyhedron; curved objects need to be approximated.

2.1.1 CSG and Subdivision

Our goal is to support boolean operations for subdivision surfaces. We imagine an interactive setting, where a designer can chose between mesh editing and boolean operations. Therefore, it is important that a boolean operation creates a control meshes which is manageable by the designer, i.e. it should not be more complex than the input meshes.

We observe that the class of geometric objects represented by subdivision surfaces is not closed under boolean operations. However, we could try to approximate the result. Clearly, a very fine control mesh results in a good approximation. The challenge is to find a coarse control mesh which produces a sufficient approximation.

In the rest of this paper, we propose a method to construct a control mesh for result of the boolean operations. For the reasons above, we aim for a coarse mesh which produces a similar shape. Moreover, we discuss how multiresolution approaches could approximate the surface with arbitrary precision.

Moreover, our algorithms allow to trade off precision for performance: each editing operation produces instantaneously a result which can be improved in a more expansive computation. For example, a designer should be able to translate object and observe how the difference changes ("slide a hole over an object"). Once the position of the objects are fixed, the system could optimize the shape, potentially a more expansive computation.

Many computer graphics applications do not require precision as much as performance and robustness. We believe that approximate boolean operations are valuable in such applications, i.e. conceptual design and animation.

3 Overview

This section gives a high-level description of the algorithm. Given two subdivision surfaces, we would like to find a control mesh for a surface approximating the result of a boolean operation.

Our algorithm proceeds in three steps. At first, we compute where the surfaces intersect. In the second step, we cut the control meshes along the intersection curves, and merge the pieces with respect to the boolean operation. These topological changes affect the shape of the subdivision surface, therefore we optimize the shape in the final fitting step (c.f. Figure ref:steps).