Advances in Engineering Software 95 (2016) 16–27

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Boolean and smoothing of discrete polygonal surfaces

Adam Updegrove[a], Nathan M. Wilson[b], Shawn C. Shadden[a,*]

[a] *Department of Mechanical Engineering, University of California, 5126 Etcheverry Hall, Berkeley, CA 94720-1740, United States*
[b] *Open Source Medical Software Corporation, Santa Monica, CA, United States*

ARTICLE INFO

Article history:
Received 4 December 2015
Accepted 24 January 2016
Available online 19 February 2016

Keywords:
Boolean
Discrete solid model
Smoothing
Surface reconstruction
Triangulated surfaces
Visualization toolkit

ABSTRACT

The development of discrete solid models from imaging has become a common practice in reverse engineering. This has motivated the need for tools to combine and manipulate discrete boundary representation of objects. Despite the importance of this problem in computer aided engineering, there is a lack of robust and efficient open-source implementations of Boolean operations for discrete geometries. This paper discusses the development of new Visualization Tool Kit (VTK) classes for the Boolean and local mesh control of triangulated solid models. The implementation presented in this paper maintains the same base classes for Boolean operations in VTK version 6.2.0, but develops new procedures within these classes. Improvements are made on the robustness and performance of the Boolean algorithms for discrete surfaces. For example, for Boolean operations consisting of order 10,000 intersecting edges, the new implementation runs an order of magnitude faster than the current Boolean implementation in VTK, and is able to handle test cases the current implementation fails to handle. In addition, surface manipulation operations were created in order to deal with issues such as surface roughness, poor quality triangles, and sharp junctions that are often encountered in discrete solid modeling. These operations are implemented locally to give increased control. A unique smoothing method is also developed to address the issue of global model distortion common to prior smoothing procedures.

1. Introduction

Evolving technologies have shifted how computational solid models are often constructed. Increasingly, computer models are reverse engineered from imaging data, rather than being constructed from traditional CAD. For example, high resolution scanners provide a set of points, which can be connected to form a surface representation of an object [1]. Using similar techniques, medical imaging (e.g., computed tomography, magnetic resonance imaging and ultrasound) can enable accurate construction of computer models representing internal organs, soft tissue, vasculature, and bones [2].

Motivation for the work herein stemmed from the need to create vascular models from medical image data, especially those suitable for computational modeling. Image-based vascular modeling is challenging given the natural tortuosity, branching and multiscale nature of cardiovascular structures [3]. In order to develop a computer model from medical image data, two main methods are in use (1) direct 3D segmentation and (2) lofted 2D segmentation. In the former approach, a surface is evolved in 3D space to obtain an appropriate boundary representation (B-Rep) of the segmented region; in the later approach, the region being segmented is skeletonized by paths, and a B-Rep is formed from lofting together a series of 2D image segmentations along the skeletal paths [2]. Direct 3D segmentation typically results in discrete polygonal surfaces, whereas prior implementation of the lofted 2D segmentation method employed commercial software to create analytic (Non-Uniform Rational B-Spline) surfaces [4]. However, in order to develop and utilize open source tools, as well as to interface into a discrete solid modeling framework, we have recently developed a method to enable 2D image segmentations to be lofted and represented as a discrete polygonal surface [5], consistent with the output of most 3D segmentation softwares.

In many simulation-based modeling applications in engineering and design it is common to perform Boolean operations between objects or models. Such operations may be necessary in the image-based model construction process (e.g., combining members segmented separately), or in the development of a multi-domain model from separate objects. One example is the virtual placement of a medical device into an anatomical model derived from image data.

There are a variety of Boolean implementations for B-Reps described in the literature. They can be classified in four categories by the computational approach: (1) tolerance and exact arithmetic,

---

* Corresponding author. Tel: +1 510 664 9800.
E-mail address: shadden@berkeley.edu (S.C. Shadden).

http://dx.doi.org/10.1016/j.advengsoft.2016.01.015

(2) approximate arithmetic, (3) volumetric, and (4) image space technqiues [6]. Tolerance and exact arithmetic methods both compute the intersection between two solids on their exact boundary, but contain different techniques for dealing with geometric robustness. Tolerances restrict floating point numbers to a specified decimal place for geometric tests; whereas, exact integer arithmetic methods convert floating point numbers to an integer-based system in which computations can be carried out exactly [7–10]. Approximate arithmetic methods reduce the computational complexity before running the geometric algorithm [11,12]. This will make the computation simpler and quicker, but lacks the exactness of the previous approaches. Volumetric techniques represent the solids first as a volume [13,14], and then perform the Boolean [15]. This results in a robust implementation; however, the boundary between the two solids is typically not resolved well in the output and a loss of geometric detail is seen [16]. Lastly, image space techniques take advantage of graphics hardware to quickly provide a boundary evaluation of the Boolean [17,18]; these methods are typically used for object collision detection. Many of these algorithms use Layered Depth Images (LDIs) to store information about the depth piercing of ray tracing from the viewpoint and have the same pitfall as the volumetric methods in which the geometric detail is not exact; however, they do provide a fast and robust method to obtain a visual of the Boolean boundary [19,20].

The Boolean implementations in literature also differ by the type of data used in the computation. Many implementations focus on NURBS surfaces [6,21]; however, there are others that investigate the procedure for polygonal surfaces [22], which have been the leading representation for discrete solid models derived from image data. There are a limited number of libraries providing open source, available, and usable Boolean operations for polygonal surfaces. The Visualization and Computer Graphics Library (VCG) has an implementation of the Boolean operation, which is implemented within the software MeshLab [23]. This is a volumetric implementation, and thus lacks the exactness of geometric detail described above. Another implementation is the GNU Triangulated Surface Library (GTS, http://gts.sourceforge.net), which we have found to be robust, but this package is no longer maintained and is difficult to include in software projects that require customization. VTK maintains an implementation for the Boolean of triangulated surfaces [24], however we have found that the implementation is not robust and often fails in various manners described below.

Rarely does the result of a Boolean give a surface that is ready for meshing and simulation, and typically other surface preparation methods must be performed. Besides defining faces for specification of boundary conditions or material properties, methods providing smoothing, blending, and manipulation of surfaces are necessary to give a solid that both accurately represents the image data and is valid for computational modeling. Moreover, discrete solid models obtained through image segmentation are often limited in quality by the resolution of the image data and inherent noise. Therefore surface manipulation tools are also necessary to improve the quality and representation of discrete solid models that serve the basis for quantitative postprocessing and simulation.

In this paper we describe the development of new VTK libraries for the Boolean, and local mesh control, of triangulated solid models. First the details of the developed Boolean procedures are discussed in Section 2. Each step of the Boolean is described followed by an analysis of the algorithm's performance. Next, customized surface manipulation operations created for local mesh control are presented in Section 3, and compared to current methods. Finally, concluding remarks are made in Section 4. The classes developed are included in the open-source SimVascular 2.0 software (simvascular.org); however, since they are implemented within the VTK framework they are modular and easily extendable elsewhere.

2. Boolean procedure

VTK maintains an implementation for the Boolean of triangulated surfaces [24], however the current implementation is not robust and often fails in various manners as demonstrated in Fig. 1. Because of VTK's broad utility, modularity, and inclusion in numerous software packages, a decision was made to adapt on the VTK implementation to create a new, more robust Boolean procedure for model creation. New methods for re-triangulation and sub-surface definition using flood fill operations were created similar to the work of Mei and Tipper [25] and found in the GTS implementation.

The current VTK version 6.2.0 Boolean operation [24] consists of two separate VTK filters, *vtkIntersectionPolyDataFilter* for finding the intersection loops and re-triangulating the surfaces, and *vtkBooleanOperationPolyDataFilter* for determining the output surface. The implementation presented in this paper maintains the same two classes, but develops new procedures within these classes. A breakdown of the classes and functions implemented is located in the Appendix. For the new implementation, a computational approach following an exact boundary representation using a tolerance was chosen. The motivation for our work was the development of computational domains from medical image data in which the exactness of the model is paramount to correct results. Thus, an approximate or image space technique is not appropriate. A

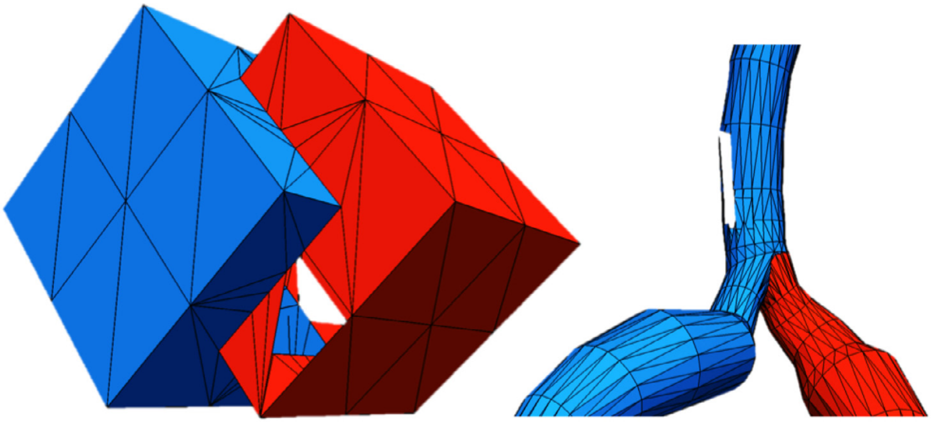

Fig. 1. The current VTK implementation often fails due to incorrect intersection point surface origin determination (left) or due to incorrect sub-surface determination using a signed distance calculation (right). Issues such as these diminish the practical utility of this implementation.