Computer-Aided Design 70 (2016) 161–170

Contents lists available at ScienceDirect

Computer-Aided Design

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

An improved star test for implicit polynomial objects[☆]

Lincong Fang [a,b,*], Dominique Michelucci [b], Sebti Foufou [c]

[a] School of Information Technology, Zhejiang University of Finance & Economics, 310018 Hangzhou, China
[b] LE2I UMR6306, CNRS, Arts et Métiers, Université Bourgogne Franche-Comté, F-21000 Dijon, France
[c] SE Department, College of Engineering, Qatar University, PO BOX 2713, Doha, Qatar

HIGHLIGHTS

- A homogeneous star test method for implicit polynomial objects.
- A linear programming optimization method to improve the efficiency of star test.
- Comparison of the naive star test and the homogeneous star test using different arithmetic.

ARTICLE INFO

Keywords:
Geometric modeling
Implicit objects
Star-shaped
Star test

ABSTRACT

For a given point set, a particular point is called a star if it can see all the boundary points of the set. The star test determines whether a candidate point is a star for a given set. It is a key component of some topology computing algorithms such as Connected components via Interval Analysis (CIA), Homotopy type via Interval Analysis (HIA), etc. Those algorithms decompose the input object using axis-aligned boxes, so that each box is either not intersecting or intersecting with the object and in this later case its center is a star point of the intersection. Graphs or simplicial complexes describing the topology of the objects can be obtained by connecting these star points following different rules. The star test is performed for simple primitive geometric objects, because complex objects can be constructed using Constructive Solid Geometry (CSG), and the star property is preserved via union and intersection. In this paper, we improve the method to perform the test for implicit objects. For a primitive set defined by an implicit polynomial equation, the polynomial is made homogeneous with the introduction of an auxiliary variable, thus the degree of the star condition is reduced. A linear programming optimization is introduced to further improve the performance. Several examples are given to show the experimental results of our method.

1. Introduction

Computing the topology of geometric objects is a fundamental problem in computer aided design (CAD), computer graphics (CG), and robotics. It plays an important role in a lot of geometric computations, such as recognition, feature extraction, simplification, motion planning, etc. Also topological information of objects has a significant impact on classification, indexation, shape description in solid modeling and shape matching.

Several research efforts have been conducted to compute the topology of geometric objects. The Cylindrical Algebraic Decomposition (CADec), due to Collins and its variants [1–5] decompose a semi-algebraic set into cells and compute its topological properties. CADec is the first method for computing the topology of a semi-algebraic set. This decomposition theoretically solves the piano mover's problem, *i.e.* the motion planning problem. Unfortunately, CADec and its variants rely on Computer Algebra tools with at least exponential cost. Thus roboticians typically prefer probably-approximately-correct (PAC) methods, like the probabilistic roadmap [6] to solve the motion planning problem. PAC methods are fast but only approximate.

Computing the topology and arrangement of planar algebraic curves has been subject to several studies [7–9], the output of the proposed methods is a graph homotopic to the input algebraic curve. However, these methods cannot process solid geometric objects.

To preserve the topology in surface extraction, a star-shaped criterion for geometric objects was presented in [10]. A star is a

☆ This paper has been recommended for acceptance by Scott Schaefer and Charlie C.L. Wang.
* Corresponding author at: School of Information Technology, Zhejiang University of Finance & Economics, 310018 Hangzhou, China.
E-mail address: lincongfang@gmail.com (L. Fang).

http://dx.doi.org/10.1016/j.cad.2015.07.005
0010-4485/

point which can see all the boundary points of a point set, and the kernel of the set consists of all star points. The star test can be reduced to testing whether the kernel is empty or not, and the kernel of polyhedral primitives can be computed using linear programming. Therefore, the authors suggested to perform a dense tessellation of the original primitive as a pre-processing step. Then an approximate kernel is computed using linear programming. At last, a point in the kernel is selected as the best candidate to perform the star test. However, the method depends on the approximation incurred in tessellation.

Recently, Delanoue et al. [11,12] proposed two exact algorithms, Connected components via Interval Analysis (CIA) and Homotopy type via Interval Analysis (HIA), resorting to interval analysis. Contrarily to CADec, their methods CIA and HIA are practicable: they only use interval computations and the classical recursive space subdivision [13]. Two other nice features of their methods is that they consider each geometric primitive independently, and they apply also to non polynomial functions (like cos, sin, exp, log, etc.). Thus, CIA and HIA can compute topological properties for Constructive Solid Geometry (CSG) shapes easily. In counterpart, for CIA and HIA, the sets, *i.e.* the free space for motion planning, must be fat (see Appendix for the definition); in practice the free space is fat, but if by mistake it is not then the methods detect it. A more serious limitation of CIA and HIA (relatively to CADec) is that they do not take into account objects defined by projections, like extrusions or parametric patches: $\boldsymbol{P} : (u, v, w) \in [0, 1]^3 \rightarrow (x, y, z) = \boldsymbol{P}(u, v, w) \in \mathbb{R}^3$ often met in CAD [14].

CIA and HIA are important for CAD because they *compute* and *certify* topologic properties of input geometric sets and objects. The need for certification (of geometric, or topological, properties of geometric programs) is currently increasing, *e.g.* Narkawicz et al. in NASA recently certified the correctness of a geometric software for the detection and avoidance of collisions between airplanes [15]. In the future, all critical geometric objects and algorithms should be certified with some proof assistant, like Coq or PVS.

We now mention three other potential applications of the star test in CAD. First, variational modeling: the idea is to specify mechanical parts or clearances with geometric constraints (incidences, tangencies, distances, angles), which a solver numerically solves. Unfortunately, topologic constraints like connectedness are not expressible into mathematical equations or inequalities. The solution is to resort to the generate-and-test paradigm of Artificial Intelligence: the solver generates all solutions, connected or not, and then CIA discards non-connected clearances.

Second, CIA and HIA can also apply to the space of feasible configurations or free motions of some given mechanisms, for example composed of articulated bars: is it possible for this flexible mechanism to move continuously from one feasible configuration to another? More generally, HIA can compute homotopy independent cycles, or homotopy independent paths between two given positions in a configuration space.

Third and last, CIA and HIA may help to study the consistence of toleranced Breps. In a nuttshell, each cell (vertex, curve, surface patch) of a toleranced BRep [16] is attached a tolerance, or thickness. Tolerances were introduced to robustly compute Boolean operations between solids: tolerances fill cracks and can solve inaccuracy issues. Shapiro [16] suggested that Leray's nerve theorem may provide a set of local conditions (like contractibility or collapsibility, some of them computable with the star test), which are necessary or sufficient to guarantee that a given toleranced BRep is consistent, for instance that it indeed separates an inside from an outside.

In this paper, we improve the method to perform the star test for geometric objects defined by implicit polynomial equations. We propose several computable and sufficient conditions to prove that a set is empty, to prove that a set is non empty, to prove that a point is a star, to prove that it is not. For implicit polynomial objects, we reduce the degree of the star condition, and compare the proposed method with the method in [11,12]. Furthermore, we implement the test using interval arithmetic [17,18] and Bernstein based methods, we then compare their performances.

The rest of the paper is organized as follows: in Section 2, we recall some basic concepts and properties of interval solvers and Bernstein based methods. Then our methods of the star test for implicit objects are introduced in Sections 3 and 4. Experimental results are shown and discussed in Section 5. We conclude the paper in Section 6.

2. Preliminaries

2.1. Star-shaped and the HIA algorithm

Here we recall some fundamental concepts and propositions [11,12].

Definition 1. A point $\boldsymbol{s}$ is a star for a subset $\boldsymbol{X}$ of an Euclidean space if $\boldsymbol{X}$ contains all the line segments connecting any of its points and $\boldsymbol{s}$.

Given a geometric set $\boldsymbol{S}$, the test whether a candidate point $\boldsymbol{s}$ is a star point of $\boldsymbol{S}$ is usually called star test.

Definition 2. If $\boldsymbol{s}$ is a star for subset $\boldsymbol{X}$ of an Euclidean space, one says that $\boldsymbol{X}$ is star-shaped or $\boldsymbol{s}$-star-shaped.

Proposition 1. *Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be two $\boldsymbol{s}$-star-shaped sets, then $\boldsymbol{X} \cap \boldsymbol{Y}$ and $\boldsymbol{X} \cup \boldsymbol{Y}$ are also $\boldsymbol{s}$-star-shaped.*

For the proof, we refer readers to the reference [11]. After Proposition 1, the star property is preserved via union and intersection. Thus, it is sufficient for HIA and CIA to consider independently each geometric primitive in the CSG tree. It is a big advantage of these methods that they do not need to compute intersections between geometric primitives, *i.e.* they avoid the (difficult) *boundary evaluation problem* [19–21].

Star test is a key component of some topology computing algorithms such as CIA, HIA, etc. To show the application of the star test, we recall the definition of the HIA algorithm briefly. Let I be a finite set of integers, $\cup_{i\in I} \boldsymbol{S}_i = \boldsymbol{S}$ is called a finite cover of $\boldsymbol{S}$, $\boldsymbol{S}_i$ are interior disjoint sets. The HIA algorithm computes a finite cover $\boldsymbol{S}_i, i \in I$ of the studied set $\boldsymbol{S}$ such that $\forall J \subset I,\ \cap_{j\in J} \boldsymbol{S}_j$ is either empty or contractible. It is contractible if it contains a star. The algorithm starts with a given initial box $\boldsymbol{B}$, $\boldsymbol{S}_i = \boldsymbol{B}_i \cap \boldsymbol{S}$, where $\boldsymbol{B}_i$ will be subboxes of $\boldsymbol{B}$. If a box $\boldsymbol{B}$ such that $\boldsymbol{B} \cap \boldsymbol{S} \neq \emptyset$ is not a member of such finite cover, it is subdivided at the center, with a random perturbation, and along the longest edge. The procedure is recursively performed until a cover satisfying the previous condition is found. Let $\boldsymbol{S}$ be a d dimensional object, then the studied boxes have dimension $d, d-1, \ldots, 0$. It may happen that one of the subdivided boxes is smaller than some given threshold ε and the expected cover is still not reached, then the whole procedure has to be re-run. Usually it is the case when some subdividing (hyper)plane is tangent or almost tangent to the given object. It is claimed that the cover can be found for most fat objects with probability 1 [11,12]. The CIA algorithm is quite similar to the HIA algorithm, more details can be found in [11,12].

Fig. 1 illustrates the rule to connect star points of the HIA algorithm in a box: let $\boldsymbol{S}$ be the object, and $\boldsymbol{B}$ be the studied box. Point E is a star for $\boldsymbol{S} \cap \boldsymbol{B}$. Point A is a star for the intersection of $\boldsymbol{S}$ and the upper edge of $\boldsymbol{B}$. Point B is a star for the intersection of $\boldsymbol{S}$ and the left edge of $\boldsymbol{B}$. Point D is a star for the intersection of $\boldsymbol{S}$ and the bottom edge of $\boldsymbol{B}$. The intersection of $\boldsymbol{S}$ and the right edge is empty. Finally, A (and C) are trivial stars for $A \cap \boldsymbol{S}$ (and $C \cap \boldsymbol{S}$). The simplicial