Computers & Graphics 73 (2018) 70–79

Contents lists available at ScienceDirect

Computers & Graphics

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

Technical Section

Accurate self-collision detection using enhanced dual-cone method[☆,☆☆]

Tongtong Wang[a], Min Tang[a,b,*], Zhendong Wang[a], Ruofeng Tong[a]

[a] Zhejiang University, China
[b] Alibaba-Zhejiang University Joint Institute of Frontier Technologies, China

ARTICLE INFO

Article history:
Received 2 October 2017
Revised 3 April 2018
Accepted 9 April 2018
Available online 14 April 2018

Keywords:
Self-collision detection
Dual-cone culling
BVH

ABSTRACT

We present an accurate and robust algorithm for self-collision detection in deformable models. Our method is based on the normal cone test and is suitable for both discrete and continuous collision queries on triangular meshes. We propose a novel means of employing surface normal cones and binormal cones to perform the normal cone test. Moreover, we combine our culling criteria with bounding volume hierarchies (BVHs) and present a hierarchical traversal scheme. Unlike the previous BVH-based dual-cone method, our method can reliably detect all self-collisions, and it achieves appreciable speedup over other high-level culling methods.

1. Introduction

To ensure the generation of physically plausible results, collision detection (CD) algorithms have been widely used in various applications, including physically based simulations, computer-aided design and computer-aided manufacturing (CAD/CAM), and robot motion planning. Such algorithms can be classified as either self-collision detection (SCD) for a single object or inter-collision detection among multiple objects. A *false negative* occurs when a CD algorithm misses a collision; a *false positive* occurs when a CD algorithm conservatively classifies a non-collision instance as a collision. An accurate CD method should not result in any *false negatives*.

Most CD algorithms use bounding volume hierarchies (BVHs) for acceleration. These methods work well for inter-object CD, but they incur high computation times for SCD for deformable objects because the adjacent primitives of a deforming mesh are in close proximity and cannot be culled through bounding volume tests. Even if a mesh (with n triangles) has no self-intersection, checking for self-collision is still quite expensive ($O(n^2)$ complexity).

Many approaches have been proposed to improve the efficiency of SCD. Volino and Thalmann [1] introduced the normal cone test for discrete collision detection (DCD). This approach was extended to continuous collision detection (CCD) by Tang et al. [2], leading to more efficient execution of self-intersection queries. Heo et al. [3] proposed a dual-cone culling method based on the normal cone test, which has lower computational overhead but may result in false negatives in practice. To address this problem, they proposed an extension that includes internal boundary edges in [3]. However, although this method results in no false negatives, maintaining such internal boundary edges can significantly reduce the performance.

Main Results: In this paper, we propose a new method that not only does not miss collisions but also accelerates the performance of the extension of the original dual-cone method. First, we introduce a sufficient set of criteria for determining whether a surface exhibits self-collisions based on two types of cones and the boundary contours of four sub-surfaces making up the entire surface (Fig. 5). The two cone types are surface normal cones and binormal cones. Second, we design a BVH-based hierarchical culling method for use in combination with our culling criteria and present a new bounding volume test tree (BVTT) traversal scheme for our culling criteria, which can significantly reduce the number of redundant tests performed. We evaluate the accuracy of our method on many complex benchmarks involving deformable models and cloth. Unlike the previous dual-cone method [3], our method can accurately detect all self-collisions. Moreover, we observe considerable speedup compared with other SCD methods.

2. Related work

In this section, we present a brief review of previous works on CD.

High-level culling: The simplest culling algorithms compute geometric bounds and use BVHs to accelerate CD. Many alternative culling methods have been proposed to reduce the number of

---

[☆] http://min-tang.github.io/home/DCC/

[☆☆] This article was recommended for publication by D Panozzo.

[*] Corresponding author at: Zhejiang University Computer Science College, Zhejiang, Hangzhou, China.

E-mail addresses: wtt923@zju.edu.cn (T. Wang), tang_m@zju.edu.cn (M. Tang).

https://doi.org/10.1016/j.cag.2018.04.001

queries. Volino and Thalmann [1] proposed the normal cone test for DCD, which takes advantage of the topology and connectivity of the input mesh and checks for self-collision by means of normal cones and 2D contour tests. Many self-collision culling techniques [2–6] have been developed based on the normal cone test. In addition, Barbič and James [7] presented a self-collision culling method for subspace deformable models, but their method does not support general deformations. Based on this method, Zheng and James [8] proposed an energy-based culling method that is applicable to general deformable models. Moreover, many clustering strategies have been proposed to improve the culling efficiency. Most of these techniques are used as preprocessing steps [9,10]. Wong et al. [11] presented a continuous SCD algorithm for skeletal models and extended it to check for collisions between a deformable surface and a solid model [12]. However, these techniques have several shortcomings during animation, and their cost reduction for CD is limited. A modified framework was proposed in [13] to improve the culling efficiency of these methods. He et al. [14] recently presented a fast decomposition algorithm in which the mesh boundary is represented using hierarchical clusters and only inter-cluster collision checks are necessary; this algorithm achieves a small speedup over previous CCD algorithms.

Low-level culling: Many techniques have been proposed to reduce the number of elementary tests between triangle pairs for CCD. Govindaraju et al. and Wingo et al. [15,16] eliminated redundant elementary tests for CCD. Hutter and Fuhrmann [17] used the bounding volumes of primitives to reduce false positives. Other methods, such as representative triangles [18] and orphan sets [2], have also been used to reduce the number of duplicate elementary tests. These low-level culling algorithms can be combined with our high-level culling method.

Reliable collision queries: Brochu et al. [19] used exact computations for reliable CCD, thereby ensuring no false negatives or false positives. Tang et al. [20] presented another exact algorithm based on Bernstein sign classification (BSC) that offers speedups of a factor of $10 - 20$ over [19]. Wang [21] introduced a useful approach based on the derivation of tight error bounds for floating-point computations. Wang et al. [22] derived tight error bounds on the BSC formulation [20] for elementary tests.

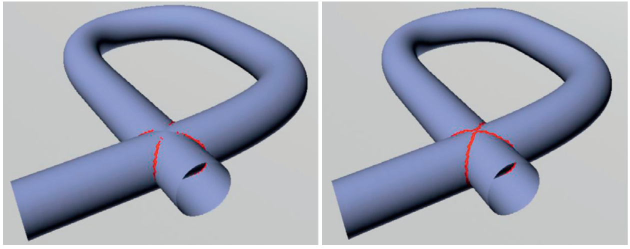
3. Overview

In this section, we present the problem definition and introduce the notation used throughout the rest of this paper. We also present an overview of the normal cone test algorithm proposed in [1].

3.1. Problem definition

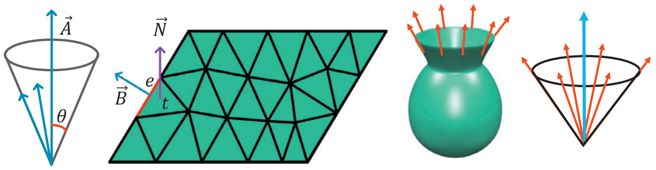
We assume that the scene of interest consists of one or many deformable objects. Each object is represented by a triangle mesh for simulation. Given two discrete time instances in a simulation, we assume that the vertices of the objects move at a constant velocity during the time interval between them. Our goal is to check whether any object exhibits any self-collision. Our approach can be used to perform both DCD and CCD on triangular meshes. For DCD, our method returns the number of potentially colliding triangle pairs. For CCD, our culling method computes the number of elementary collisions between vertex-face (VF) pairs and edge-edge (EE) pairs.

3.2. Notation



(a) Dual-Cone Method (b) Our Method

Fig. 1. *Pipe Benchmark.* We illustrate the benefits of our SCD algorithm using the Pipe benchmark (78K triangles). The colliding triangle pairs are highlighted in red. Unlike the previous dual-cone method (a), our method (b) can detect all the collisions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) (b) (c)

Fig. 2. *Binormal Vectors and BNCs.* (a) The definition of a cone. (b) The binormal vector $\vec{B}$ computed from the boundary edge e on triangle t. (c) An example of a BNC computed from a mesh. The BNC contains all the red vectors. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

We use the following acronyms throughout the rest of the paper: BV, BVH, and BVTT stand for bounding volume, bounding volume hierarchy, and bounding volume test tree, respectively. We define a cone $(\vec{A}, \theta)$ in terms of $\vec{A}$, the axis, and θ, half of the apex angle of the cone (Fig. 2a). Unless otherwise specified, the angle of a cone refers to θ. For a BVH node N, N_l and N_r represent its left and right child nodes, respectively; N_{ll} and N_{lr} represent the left and right child nodes of N_l; and N_{rl} and N_{rr} represent the left and right child nodes of N_r.

3.3. Normal cone test

Several widely used SCD algorithms are based on the normal cone test algorithm proposed by Volino and Thalmann [1]. Given a continuous surface S bounded by a contour C, a sufficient set of criteria for no self-collision consists of both of the following sequential conditions:

- *Surface normal test:* There exists a vector $\vec{V}$ for which $(\vec{N} \cdot \vec{V}) > 0$ at every point on S, where $\vec{N}$ is the normal vector at each point on the surface.
- *Contour test:* The projection of C along the vector $\vec{V}$ does not have any self-intersections on a plane orthogonal to $\vec{V}$.

Provot [4] presented an efficient method for evaluating whether the first condition is satisfied based on normal cones, which can be computed by combining the normal vectors of individual triangles in a triangular mesh. However, the contour test has a worst-case time complexity of $\Theta(N^2)$, where N is the number of edges on the projected plane. To improve the efficiency of the normal cone test, Heo et al. [3] proposed a dual-cone culling method based on surface normal cones (SNCs) and binormal cones (BNCs). However, this method may result in false negatives in practice when it is combined with a BVH-based CD method.

4. Dual-cone culling method

In this section, we briefly review the previously proposed dual-cone culling method [3] and highlight several cases in which this method may result in false negatives.