Computer-Aided Design 78 (2016) 14–25

Contents lists available at ScienceDirect

Computer-Aided Design

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

Continuous penetration depth computation for rigid models using dynamic Minkowski sums☆

Youngeun Lee [a], Evan Behar [b], Jyh-Ming Lien [b,a], Young J. Kim [a,*]

[a] Ewha Womans University, Seoul, South Korea
[b] George Mason University, Fairfax, VA, USA

ARTICLE INFO

Keywords:
Penetration depth
Minkowski sum
Collision detection
Convolution

ABSTRACT

We present a novel, real-time algorithm for computing the continuous penetration depth (CPD) between two interpenetrating rigid models bounded by triangle meshes. Our algorithm guarantees gradient continuity for the penetration depth (PD) results, unlike conventional penetration depth (PD) algorithms that may have directional discontinuity due to the Euclidean projection operator involved with PD computation. Moreover, unlike prior CPD algorithms, our algorithm is able to handle an orientation change in the underlying model and deal with a topologically-complicated model with holes. Given two intersecting models, we interpolate tangent planes continuously on the boundary of the Minkowski sums between the models and find the closest point on the boundary using Phong projection. Given the high complexity of computing the Minkowski sums for polygonal models in 3D, our algorithm estimates a solution subspace for CPD and dynamically constructs and updates the Minkowski sums only locally in the subspace. We implemented our algorithm on a standard PC platform and tested its performance in terms of speed and continuity using various benchmarks of complicated rigid models, and demonstrated that our algorithm can compute CPD for general polygonal models consisting of tens of thousands of triangles with a hole in a few milli-seconds while guaranteeing the continuity of PD gradient. Moreover, our algorithm can compute more optimal PD values than a state-of-the-art PD algorithm due to the dynamic Minkowski sum computation.

1. Introduction

Measuring the amount of interpenetration between overlapping models is an important problem in geometric modeling, computer graphics, computational geometry, and algorithmic robotics. A widely used distance measure for interpenetration is penetration depth (PD), which is defined as a minimum translation to separate overlapping models [1,2]. In simulated environments such as dynamics simulation, assembly planning, robot motion planning or six-degree-of-freedom haptic rendering, model overlap happens frequently due to numerical/control errors, interface latency or user-in-the-loop inherent to the environments. Such a penetration state is often considered an invalid state in computer simulation, and PD often plays a major role in recovering the invalid state to a valid, collision-free state; this type of approach is broadly classified as a penalty-based system.

It is well-known that the PD can be computed using the Minkowski sums. The Minkowski sums between $\mathcal{A}$ and $\mathcal{B}$ are defined as [3,4].

$$\mathcal{A} \oplus \mathcal{B} = \{\mathbf{a} + \mathbf{b} \mid \mathbf{a} \in \mathcal{A}, \mathbf{b} \in \mathcal{B}\} \tag{1}$$

$$\mathcal{A} \oplus -\mathcal{B} = \{\mathbf{a} - \mathbf{b} \mid \mathbf{a} \in \mathcal{A}, \mathbf{b} \in \mathcal{B}\}. \tag{2}$$

If $\mathcal{A}$ and $\mathcal{B}$ overlap, their penetration depth $PD(\mathcal{A}, \mathcal{B})$ is equivalent to finding the minimum distance between the common origin of $\mathcal{A}$ and $\mathcal{B}$, $\mathbf{o}$ and the boundary surface of the Minkowski sums, $\partial(\mathcal{A} \oplus -\mathcal{B})$ [2]:

$$PD(\mathcal{A}, \mathcal{B}) = \{\min \|\mathbf{q}\| \mid \mathbf{q} \in \partial(\mathcal{A} \oplus -\mathcal{B})\}. \tag{3}$$

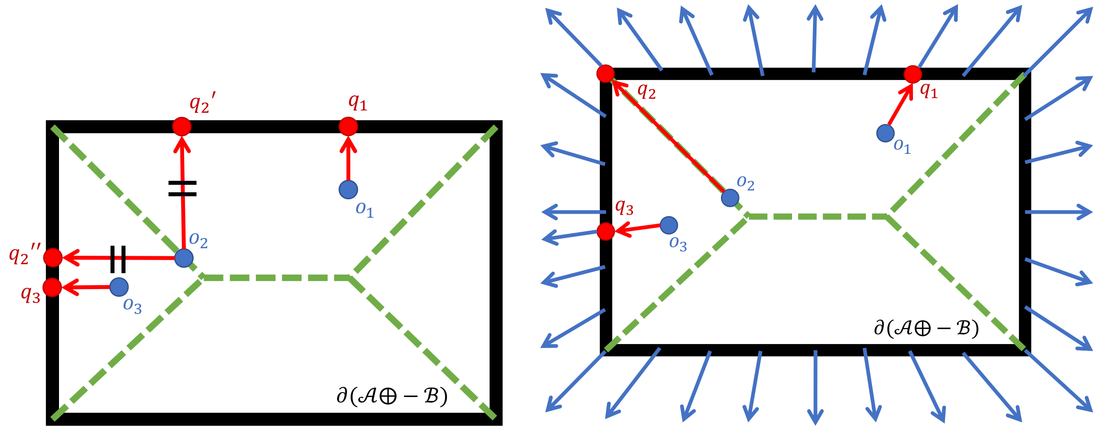
In other words, we can compute PD between $\mathcal{A}$ and $\mathcal{B}$ by projecting $\mathbf{o}$ onto the surface of Minkowski sums $\partial(\mathcal{A} \oplus -\mathcal{B})$.

However, it is known that the definition of PD in Eq. (3) can lead to a discontinuity in the direction of PD (i.e. the gradient of PD), when $\mathbf{o}$ crosses the medial axis of $\partial(\mathcal{A} \oplus -\mathcal{B})$ [5], as illustrated

☆ This paper has been recommended for acceptance by Scott Schaefer and Charlie C.L. Wang.

* Corresponding author.
E-mail addresses: youngeunlee@ewhain.net (Y. Lee), behare@gmail.com (E. Behar), jmlien@cs.gmu.edu (J.-M. Lien), kimy@ewha.ac.kr (Y.J. Kim).

http://dx.doi.org/10.1016/j.cad.2016.05.012



(a) PD using Euclidean projection.

(b) CPD using Phong projection.

Fig. 1. In both figures, the thick rectangle is the surface of Minkowski sums and the dashed green lines represent its medial axis. The blue dots show the origin $\mathbf{o}$ as it moves from $\mathbf{o}_1$ to $\mathbf{o}_2$, and then $\mathbf{o}_3$. The red dots $\mathbf{q}_i$ show the projections of $\mathbf{o}$ on the surface of Minkowski sums, and the red arrows denote the projection directions. (a) The direction of PD using conventional Euclidean projection is not continuous when $\mathbf{o}_2$ crosses the media axis. For example, the projection suddenly jumps from $\mathbf{q}_2$ to $\mathbf{q}_2'$ even though the magnitude of PD is still continuous. (b) The surface normals (the blue arrows) on the surface of Minkowski sums are continuously defined using Phong interpolation. The projection results using Phong projection are continuous even if $\mathbf{o}_2$ is on the medial axis as the projection direction changes continuously along the corresponding surface normal. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

in Fig. 1(a). This discontinuity problem is a critical issue in any penalty-based system [6] that relies on PD as a measure of penalty response, for instance, six-degree-of-freedom haptic rendering, as it can induce serious simulation instability.

Recently, there has been a limited amount of research efforts to put into solving the discontinuity problem in PD, a new PD approach known as continuous penetration depth (CPD). Zhang et al. [7] first proposed a possible solution for CPD using spherical parameterization of configuration space. Lee and Kim [8] proposed another CPD approach using Phong projection [8]. Unfortunately both approaches are rather preliminary and limited in terms of practical applicability as they do not work when the models rotate or contain holes.

*Main results:*We present a novel algorithm to compute continuous penetration depth in real-time between two intersecting models. Our new CPD is defined using Phong projection on the subspace of Minkowski sums. This guarantees the continuous change of a PD gradient value with respect to infinitesimal rigid motion of the underlying model. In order to design an efficient CPD algorithm, we conservatively predict a solution space of CPD using a conventional PD algorithm based on Euclidean projection and construct an ϵ-ball around the predicted solution space, then compute Minkowski sums dynamically and locally inside the ϵ-ball. This dynamic Minkowski-sum method is achieved by repairing the damage on the Minkowski-sum surface due to infinitesimal rotation via the new concepts of convex map (defined in Section 4) and gap filling (Section 5). To the best of our knowledge, this is the first practical method that handles general polyhedra for Minkowski-sum computation. Then, we search for a CPD solution in the local Minkowski sums. If such a solution does not exist, we incrementally enlarge the ϵ-ball until a solution is found. We have implemented our CPD algorithm on a conventional PC platform and tested our algorithm with various benchmarking scenarios including diverse motion sequences and models of complicated geometry and topology. In our experiments, our algorithm can compute CPD for models consisting of tens of thousands triangles with holes in a few msecs while guaranteeing the continuity of PD.

*Organization:*The rest of the paper is organized as follows. We briefly survey previous work relevant to PD and dynamic Minkowski sum computation in Section 2. We discuss preliminary information and give an overview of the algorithm in Section 3. In Sections 4 and 5, we propose our algorithm to construct a local Minkowski sum, and explain how to perform Phong projection efficiently in Section 6. We show experimental results in Section 7. Finally, we conclude our work and provide a discussion on future work in Section 8.

2. Previous work

In this section, we briefly survey the work relevant to our research, namely penetration depth computation and dynamic Minkowski sum for rigid models.

2.1. Penetration depth computation for rigid models

Penetration depth. The penetration depth (PD) can be computed using a Minkowski sum-based formulation [1]. It is well-known that complexity of Minkowski sum computation is $O(n^2)$ and $O(n^6)$ for convex and non-convex models in 3D, respectively, where n is the number of facets in the objects. For convex models, exact PD can be computed using a Dobkin–Kirkpatrick hierarchy-based acceleration structure [2] and a randomized algorithm [9]. There exist various algorithms to compute approximate PD for convex models. Cameron [4] computed PD based on upper and lower bounds. Bergen [10] and Kim et al. [11] computed approximated PD in real-time.

For non-convex models, Hachenberger [12] proposed an exact PD computation algorithm based on Minkowski sums, but it is relatively slow. Various approximation algorithms have been developed for non-convex objects, but they are rather slow in practice [13,14]. Later, Je et al. [15] proposed a real-time algorithm based on iterative projection on the Minkowski sum.

Continuous penetration depth. To the best of our knowledge, there are only two works related to CPD. Zhang et al. [7] first introduced the notion of CPD and defined the CPD using a spherical parameterization of configuration space. Since this algorithm is based on sampling and machine learning, its runtime performance is very fast. However, the algorithm requires heavy preprocessing for configuration space approximation and spherical parameterization that makes it very challenging when models rotate, as the algorithm needs to recompute the entire preprocess from scratch. Moreover, the spherical parameterization does not work when the configuration space is not homeomorphic to a sphere. To circumvent these issues, Lee and Kim [8] compute CPD