Title:
Fast and Accurate Local Minimal Distance Computation between Industrial CAD Models for Interactive Dynamics Simulations

Authors:
Sébastien Crozet, sebastien.crozet@cea.fr, CEA, LIST, Interactive Simulation Laboratory, Gif-sur-Yvette
Jean-Claude Léon, jean-claude.leon@grenoble-inp.fr, INRIA, Grenoble University
Xavier Merlhiot, xavier.merlhiot@cea.fr, CEA, LIST, Interactive Simulation Laboratory, Gif-sur-Yvette

Keywords:
Collision Detection, BRep, Smooth Constraints, Interactive Simulation, Local Minimal Distances

DOI: 10.14733/cadconfP.2017.191-195

Introduction:
Computing contact points between two CAD models is important for many applications including motion-planning, physical simulation, haptic rendering, and robotics. Penetration depth [8] and distance [4] computation are the two main approaches for defining and computing contact points. The method described on this paper is distance-based and produces results (contact points and contact normals) that evolve continuously with respect to motion parameters. Such continuity is desirable in the context of interactive physical simulation of scenarios involving rolling and sliding motions where jumps in the normals directions may induce numerical artifacts that affect the behavior of the simulated objects. We achieve this by working directly with the smooth representation of the CAD models instead of resorting to a tessellation. Alternative methods based on the computation of a *continuous* penetration depth exist [7] but only consider an approximation of the contact space, and are limited to translational motions only.

The main contributions of this paper are a complete method for computing contact points between CAD models without tessellation, an off-line geometrical analysis for grouping several exact distance computations into a single one, and a new bounding volume for normals which is much tighter than existing alternatives. We also show that, in some practical applications, our method can feature better performances and accuracy compared to recent distance-based tessellation-based approaches [4] and to our previous work [1].

Main Idea:
Problem statement and overview

Our goal is to compute contact points between industrial CAD models represented by their smooth boundaries. Such a boundary representation (BRep) is a 2-manifold CW-complex, connected, closed, oriented, and without self-intersections. It is composed of features with various dimensions: vertices (0-dimensional), edges (1-dimensional), and surfaces (2-dimensional). The geometry of edges are curves while the geometry of faces are parametric surfaces trimmed by restriction curves. Note that in practice industrial models are mostly composed of *canonical faces*, i.e., planes, spheres, cylinders, cones, and torus (see Fig. 1). Our distance-based collision detection method performs special cases for each type of surface in order the select the most efficient dedicated algorithms.

Proceedings of CAD'17, Okayama, Japan, August 10-12, 2017, 191-195

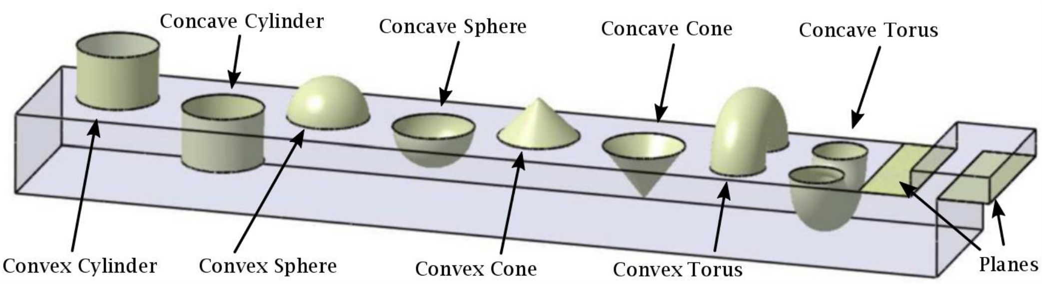

Fig. 1: Convex and concave canonical surfaces.

Given two non-intersecting BRep models, we define contact points as the local minimizers of their distance function. Thus, our method is distance-based and relies on the definition of a gap function as in Fig. 2. Our method is split into two stages:

1. The pre-computation stage where a geometrical analysis of the input models is performed and a hierarchical acceleration data structure is constructed.
2. The run-time algorithm then traverses this data structure in order to locate and compute all local minimal distances efficiently.

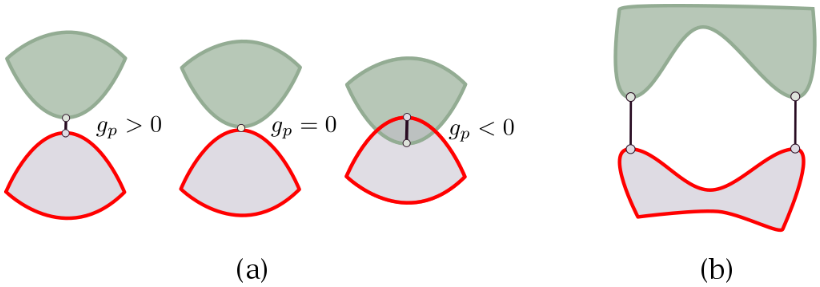

Fig. 2: (a) Definition of a contact point from a gap function which is positive when objects are separated, zero when touching, and negative when penetrating. (b) Concave objects may generate more than one local minimal distance.

Pre-computation

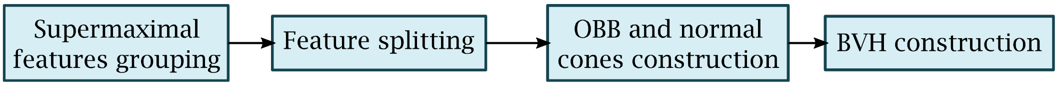

Fig. 3: Operations performed during the preprocessing stage.

The pre-computation stage (see Fig. 3) starts with the identification and merging process to generate *supermaximal* features. Those are faces and edges whose geometry share identical intrinsic properties independently from their domain of definition. In other words, those are features that would be exactly superimposed if they were not trimmed. Fig. 4 depicts an example of BRep model for which cylindrical and planar parts had to be cut into several pieces by the modeler to comply with the CW-complex structure of a BRep. The grouping process identifies which pieces of surfaces (resp. curves) are part of the same supermaximal surface (resp. curve) and groups them as such in a dedicated data structure. This proves useful for the run-time phase because only one exact contact point

Proceedings of CAD'17, Okayama, Japan, August 10-12, 2017, 191-195

determination algorithm will have to be executed for a given pair of supermaximal (canonical) curves/surfaces, instead of one execution per pair of piece of curve/surfaces.

Because the efficiency and accuracy of our run-time algorithm relies on tight bounding volumes and good approximate localization of potential contact points, the second stage of our preprocessing step split all curved supermaximal feature into *almost-flat* pieces. A piece of surface or curve is said almost-flat if it meets some *flatness conditions* as in [2]; namely, the apex angle of their normal (revolution) cone has to be smaller than a given user-defined threshold. Note that this splitting is only logical for all canonical curves and surfaces: their domain definition is cut into pieces but the geometry itself is left untouched. Complex surfaces like Bézier and NURBS curves and surfaces on the other hand have their geometry actually split using subdivision techniques [5] because the newly computed control points allow efficient computation of bounding volumes.

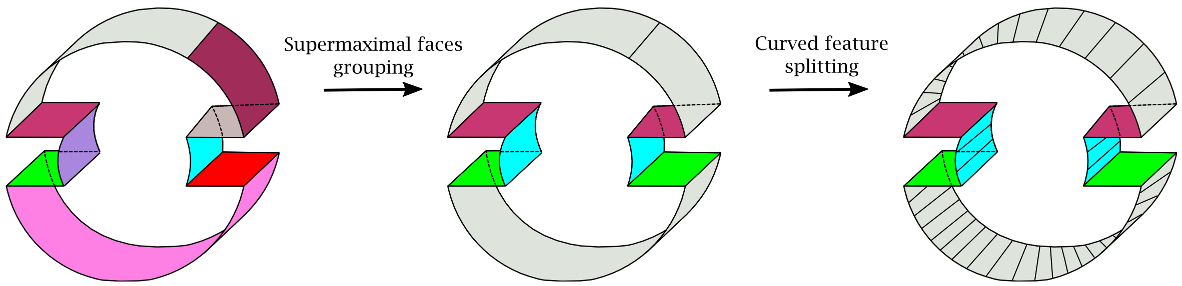

Fig. 4: Leftmost: BRep model with one color per face exported by the modeler. Right: surface areas part of the same supermaximal face share the same color.

Third, we compute a binary Bounding Volume Tree (BVT) where each leaf contains one piece of feature previously split together with a reference to the supermaximal feature it belongs to. As for the Spatialized Normal Cone Hierarchy introduced by Johnson [3], each node of the BVT contains a pair of bounding volumes: one that takes into account the spatial extent of the bounded features pieces, and another one that takes into account its orientation. We chose Oriented Bounding Boxes as a spatial bound, and designed a new normal bound that is significantly tighter than the revolution normal cones used by [3]. Our new normal bounding volume is a finitely-generated polyhedral normal cone which can be seen as the convex polyhedron built by intersecting a finite number of half-spaces whose planar boundaries all contain the origin (see Fig. 5). We designed an efficient intersection test between two polyhedral normal cones and an efficient way to compute them for faces, edges, and vertices of the BRep models. In particular, computing the polyhedral normal cone of a free-form surface like a Bézier surface or a NURBS is non-trivial and requires a convex hull algorithm on the unit sphere $\mathbf{S}^2$ that we designed as well.

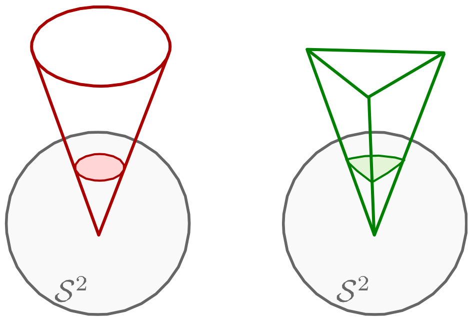

Fig. 5: Example of revolution cone (left) and a polyhedral cone (right). Their intersection with the unit sphere $\mathbf{S}^2$ is shown.

Run-time algorithm

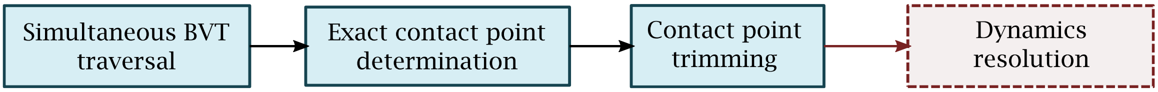

Fig. 6: Various steps (blue) performed by the run-time algorithm before outputting contact points to the dynamics constraints solver (red).

Three steps (see Fig. 6) are followed in order to compute all local minimal distances between two BRep. First, their BVT are simultaneously traversed in order to use bounding volumes to filter out efficiently pairs of feature pieces that are too far from one another to be of interest, and those that are rotated in such a way that they cannot contain any solution. Note that if two pairs of feature pieces involve the same supermaximal features, only one pair is kept to avoid redundant computations.

Second, we attempt to find actual local minimal distances on each feature pairs (that has not been filtered out) using methods that depend on the exact type of feature involved. In particular, we find closest points analytically for feature pairs involving only lines, planes, spheres, cylinders, cones, and points. For more complex curves and surfaces like torus, Bézier, and NURBS surfaces, local minimal distances are found iteratively by an unconstrained Newton-like local minimization method.

Finally, all contact points thus found must be checked to be actually inside of the trimmed domain of the corresponding surfaces and curves. Each face of the BRep model is associated with a set of trimming curves that cut the surface parametric domain. Testing that a closest point is valid amounts to testing that its parametric coordinates are inside of the valid domain of both features. This can be done efficiently by existing methods [6].

Results

Within the context of telerobotics we show how our method using BRep models directly for collision detection may perform better than our previous works [1] and than an alternative distance-based approach on polyhedrons [4]. The scenario shown by Fig. 7 is a simulation of the slave arm side of a force-feedback teleoperation system.

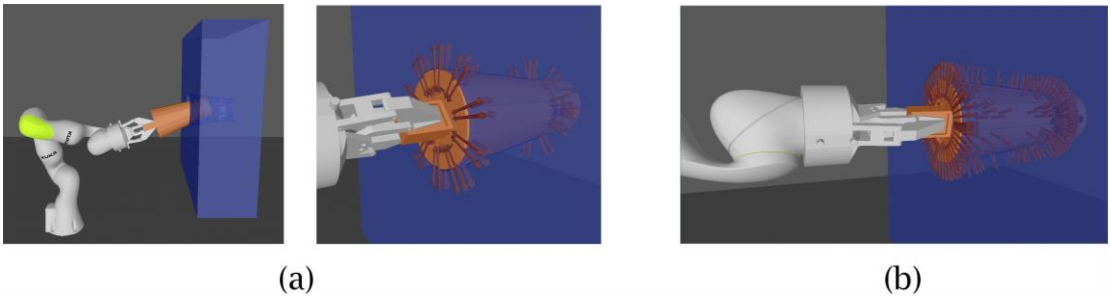

Fig. 7: A task of insertion with small mechanical clearance in the context of telerobotics. Red arrows (right) are the contact points between the conical object and the conical hole. (a) Our method based on BReps. (b) Existing distance-based method [4] on tessellated versions of the models.

In order to properly simulate such insertion tasks with small mechanical clearance, we need to use the exact, smooth, geometry of both models. Tessellated geometries become acceptable only with a sufficiently high numbers of triangles; otherwise the insertion cannot be completed due to discretization artifacts. Fig. 7(a) shows the 116 contact points generated in 19.8ms by our method using smooth geometries. In the same configuration, Fig. 7(b) shows the 433 contacts points generated in 29.3ms by [4] using the tessellated models. Thus, for such use-cases with sliding motions, our method can be both faster and generate less contact points. Moreover, this lowers significantly the time taken by the rest of the physics simulation pipeline for solving the contacts dynamics.

Conclusion:

In this paper, we present a complete and efficient method for computing contact points between objects modeled by their boundary representation (BRep). Our method is distance-based and uses geometrical analysis together with a new bounding volume for normals in order to find potential contact pairs efficiently. Exact contact points are then computed efficiently using dedicated algorithms, depending on the actual geometry types of the involved features. Our applications show that compared to tessellation-based algorithms our method based on the exact, smooth, CAD models geometry can be very beneficial, both in terms of performance and accuracy to some scenarios involving sliding or rolling motions. It also show a significant improvement of computation times compared to our previous works [1].

References:

[1] Crozet, S.; Léon J.; Merlhiot, X: Fast computation of contact points for robotic simulations based on CAD models without tessellation, International Conference on Intelligent Robots and Systems 2016, IEEE, Daejeon, 2016, pp. 937-944

[2] Johnson, D.; Cohen, E: Distance extrema for spline models using tangent cones, Proceedings of Graphics Interface 2005, Canadian Human-Computer Communications Society, Victoria, British Columbia, 2005, 169-175.

[3] Johnson, D.; Cohen, E.: Spatialized normal cone hierarchies, Symposium on Interactive 3D Graphics, ACM, New York, NY, USA, 2001, 129-134. https://doi.org/10.1145/364338.364380

[4] Merlhiot X.: A robust, efficient and time-stepping compatible collision detection method for non-smooth contact between rigid bodies of arbitrary shape, in Multibody Dynamics ECCOMAS Thematic Conference, Milano, Italy, 2007.

[5] Piegl, L.; Tiller W.: The NURBS Book, 2nd ed., Springer-Verlag, New York, NY, USA, 1997. https://doi.org/10.1007/978-3-642-59223-2

[6] Schollmeyer, A.; Fröhlich B.: Direct trimming of NURBS surfaces on the GPU, ACM Transactions on Graphics, 28(3), 2009, 1-9. https://doi.org/10.1145/1531326.1531353

[7] Zhang, X.; Kim, Y. J.; Manocha, D.: Continuous penetration depth, Computer-Aided Design, 46(1), 2014, 3-13. https://doi.org/10.1016/j.cad.2013.08.013

[8] Zhang, L.; Kim, Y.J.; Varadhan, G.; Manocha, D.: Generalized penetration depth computation, Computer-Aided Design, 39(8), 2007, 625-638. https://doi.org/10.1016/j.cad.2007.05.012