ARTICLE IN PRESS

Computer-Aided Design ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Computer-Aided Design

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

Robust free space construction for a polyhedron with planar motion[☆]

Elisha Sacks [a,*], Nabeel Butt [b], Victor Milenkovic [c]

[a] Computer Science Department, Purdue University, West Lafayette, IN 47907-2066, USA
[b] Autodesk, Pier 9 The Embarcadero, San Francisco, CA 94111, USA
[c] Department of Computer Science, University of Miami, Coral Gables, FL 33124-4245, USA

ARTICLE INFO

Keywords:
Configuration space
Robust computational geometry
Path planning

ABSTRACT

We present a free space construction algorithm for a polyhedron that translates in the *xy* plane and rotates around its *z* axis, relative to a stationary polyhedron. We employ the proven paradigm of constructing the configuration space subdivision defined by patches that comprise the configurations where the boundary features of the polyhedra are in contact. We implement the algorithm robustly and efficiently. The challenge is to detect degenerate predicates efficiently and to handle them correctly. We use our ACP (Adaptive Controlled Perturbation) robustness strategy to prevent degenerate predicates due to input in special position. The remaining cases are predicates that are identical to the zero polynomial because their arguments are derived from overlapping sets of input vertices. We detect and handle these cases with custom logic. We validate the implementation by computing maximum clearance paths.

1. Introduction

We present research in rigid body kinematics. We consider a polyhedron *A* that translates and rotates in a plane while avoiding a stationary polyhedron *B*. The polyhedra can have multiple components and need not be convex. These kinematic pairs model vehicles that travel on the ground while avoiding obstacles, layout of parts that rest on a base, and mechanical systems with planar motion. Fig. 1 shows a simple example in which a tetrahedron *A* navigates an obstacle *B*. Our task is to compute the configurations (positions and orientations) of *A* where it is disjoint from *B*. The analysis supports motion planning [1], part layout [2], and mechanical design [3].

In an appropriately chosen coordinate system, *A* translates in the *xy* plane and rotates by angle θ around its *z* axis. The manifold with coordinates (x, y, θ) is the *configuration space*. The constraint that *A* cannot intersect *B* restricts *A* to an open subset of configuration space, the *free space*. The boundary, the *contact space*, consists of the configurations where the boundaries of *A* and *B* intersect but the interiors are disjoint. Fig. 2 shows the contact space of the example. The free space is its exterior. The free configurations from Fig. 1 are drawn as green spheres. The tetrahedron can move in or out of the obstacle along the curve that connects these configurations.

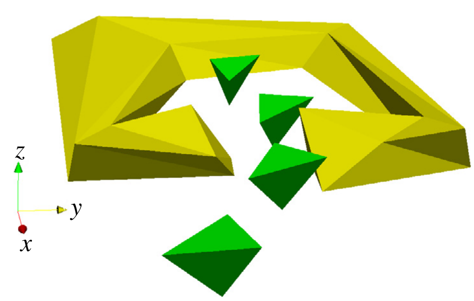

Fig. 1. Tetrahedron in four configurations and obstacle.

We present an algorithm for constructing a boundary representation of free space (Section 3). The key fact is that at a contact configuration either a vertex of one polyhedron lies on a facet of the other polyhedron or two edges share a point. The configurations where a vertex/facet or an edge/edge pair intersect form a surface called a *patch*. Fig. 3 shows the two types of patches for our example and Fig. 4 shows all the patches. The patches subdivide configuration space into open regions, called *cells*. The free space is a disjoint union of cells and the contact space is a subset of the union of the patches. Our algorithm constructs the subdivision and identifies the free space cells.

We develop the first robust implementation of the algorithm, using our ACP robustness strategy [4] (Section 4). We validate the implementation by computing maximum clearance paths (Section 5).

[☆] This paper has been recommended for acceptance by C.L. Wang and Y. Chen

* Corresponding author.
E-mail addresses: eps@purdue.edu (E. Sacks), nabeel.butt@autodesk.com (N. Butt), vjm@cs.miami.edu (V. Milenkovic).

http://dx.doi.org/10.1016/j.cad.2017.05.017
0010-4485/

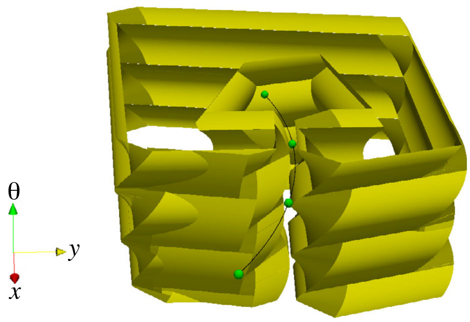

Fig. 2. Contact space of tetrahedron and obstacle.

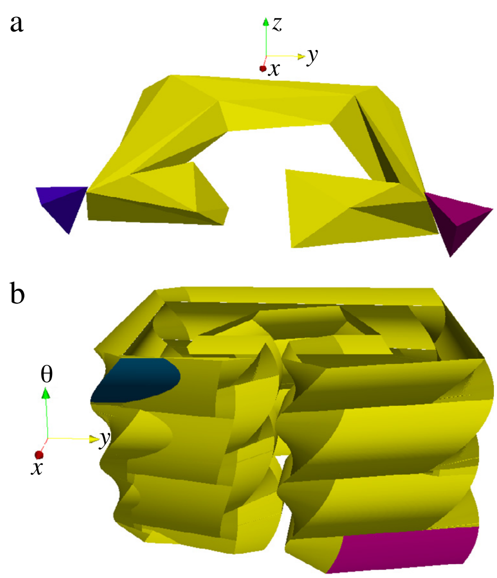

Fig. 3. Vertex/facet (left) and edge/edge (right) contacts (a) and corresponding patches in same colors (b).

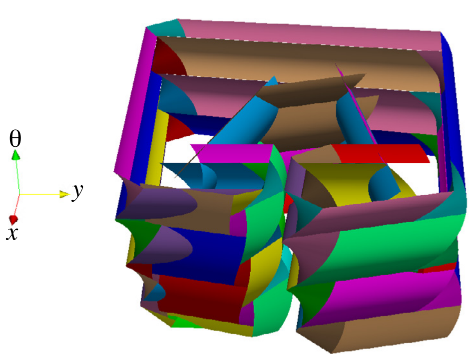

Fig. 4. Patches for tetrahedron and obstacle.

2. Prior work

The difficulty of free space construction grows sharply with configuration space dimension. Dimension two is implemented in the CGAL library [5]. For dimension three, we have devised state-of-the-art algorithms for a planar body [4] and for a translating polyhedron [6]. Both algorithms have the same structure as the current algorithm – construct the subdivision of the patches and identify the free cells – and are implemented robustly using ACP. Kim, Elber, and Kim [7] construct patch intersection curves for planar bodies bounded by spline curves. The only prior algorithm for the configuration spaces in this paper [8] is not robust and never appears to have been implemented. The best algorithm for configuration spaces of dimension four and up [9] has exponential complexity in dimension and has not been implemented.

The limited progress in free space construction has led to algorithms in which the configuration space is sampled and the free samples are linked into a graph via short paths in free space [10]. The hardest case is long narrow passages. One strategy for detecting them is local search near contact configurations [11]. Free space construction and probabilistic method can also be combined [12].

Wang, Chiang, and Yap [13] argue that exact predicate evaluation is too slow for free space construction. They approximate free spaces of planar bodies using a bounded-depth recursive axis-parallel subdivision.

The cost of the approximate approaches is proportional to ϵ^{-d} with ϵ the accuracy and d the dimension of the configuration space. This cost is prohibitive in robot path planning with narrow free space passages, in precision assembly planning, in mechanical design, and in part layout.

Free space construction is related to penetration depth computation. The penetration depth of a configuration in the complement of free space is the minimum distance to a configuration in contact space. In other words, it is the smallest motion that transforms a configuration in which the parts overlap to one in which they do not overlap. It appears that the only way to compute the penetration depth is to construct the free space. The complexity of this task motivates research on approximate and heuristic penetration depth computation. Tang and Kim [14] use a heuristic to find a contact configuration near the overlap configuration then locally minimize the distance to the overlap configuration. The error is large when the local minimum is far from the global minimum. Pan and Manocha [15] approximate the contact space with a support vector machine (SVM) that they construct from free and overlap configurations, collected using uniform sampling followed by active learning. Since an SVM defines a smooth surface, sharp features and narrow channel are poorly modeled. Kim, Manocha, and Kim [16] combine this algorithm with local refinement of the contact space. He, Pan, Li, and Manocha [17] approximate the contact space with a graph of configurations that they obtain via random sampling followed by local search around contact samples. They compute penetration depth via nearest-neighbor search and interpolation. The approach is more accurate than prior work, yet narrow passages remain problematic. None of this work provides error bounds.

3. Algorithm

The input to the free space construction algorithm is polyhedra A and B with manifold triangle mesh boundaries. The geometric part of the algorithm constructs the patches (Section 3.1), the intersection curves of two patches (Section 3.2), and the intersection points of three patches (Section 3.3). The combinatorial part of the algorithm constructs the subdivision of the patches and identifies the cells that comprise the free space (Section 3.4).

3.1. Patches

The *configuration* of A is $\mathcal{C} = (u, \theta)$ with $u = (x, y, 0)$ a translation vector and with θ a rotation angle around its z axis. A configuration $\mathcal{C}$ maps a point p to $u+\theta p$ where θp denotes p rotated by θ. The image of A is written as $A(\mathcal{C})$. A *contact* is a configuration $\mathcal{C}$ where (1) a vertex a of $A(\mathcal{C})$ lies on a facet bcd of B, (2) a vertex a of B lies on a facet bcd of $A(\mathcal{C})$, or (3) an edge ab of $A(\mathcal{C})$ shares a point with an edge cd of B. The contact is *compatible* if the interiors of $A(\mathcal{C})$ and B are disjoint in a neighborhood of the shared point (Fig. 6). The compatible contacts form a surface, called a *patch*. Fig. 5 shows

Please cite this article in press as: Sacks E., et al. Robust free space construction for a polyhedron with planar motion. Computer-Aided Design (2017), http://dx.doi.org/10.1016/j.cad.2017.05.017.