Purdue University
Purdue e-Pubs

Department of Computer Science Technical Reports | Department of Computer Science

2001

Configuration Space Computations for Polyhedra With Planar Motions

Elisha Sacks
Purdue University, eps@cs.purdue.edu

Report Number:
01-004

Sacks, Elisha, "Configuration Space Computations for Polyhedra With Planar Motions" (2001). *Department of Computer Science Technical Reports.* Paper 1502.
https://docs.lib.purdue.edu/cstech/1502

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.

CONFIGURATION SPACE COMPUTATION FOR POLYHEDRA WITH PLANAR MOTIONS

Elisha Sacks

Department of Computer Science
Purdue University
West Lafayette, IN 47907

CSD TR #01-004
February 2001

Configuration space computation for polyhedra with planar motions

Elisha Sacks
1398 Computer Science Building
Purdue University
West Lafayette, IN 47907-1398
eps@cs.purdue.edu

February 21, 2001

Abstract

This report describes an algorithm that computes the configuration space of a polyhedron with planar motion relative to a fixed polyhedral obstacle. The algorithm decomposes the configuration space along the orientation axis into intervals of cross-sections with the same structure. It computes the angles where the structure changes. In each interval between two changes, it computes the (fixed) free-space structure from a representative cross-section then lifts the contact curves to construct an adjacency graph of contact patches.

1 Introduction

This report describes an algorithm that computes the configuration space of a polyhedral part with planar motion relative to a fixed polyhedral obstacle. The obstacle is specified in a global frame (x, y, z). The part is specified in a local frame that is aligned with the global frame (same orientation) and whose origin lies in the xy plane. The part translates parallel to the xy plane and rotates around its z axis. The configuration space is a cylinder whose coordinates are the position (u, v) of the part frame in the xy plane and the angle ψ between the part x axis and the global x axis.

The algorithm constructs a boundary representation of free space comprised of contact patches. The patches are represented implicitly and parametrically. Each patch is bounded by a simple loop of boundary curves, which are represented parametrically. The boundary representation is represented as a contact graph whose nodes describe the patches and whose links describe the boundary curves of adjacent patches.

The contact graph is constructed by a method that I developed in prior work [3] and implemented for planar geometry [2]. The configuration space is decomposed along the ψ axis into intervals of uv cross-sections called slices. The slices within an interval have the same structure, but the structure can change at interval boundaries. In each interval, the algorithm construct a representative slice, deduces its structure, and lifts this structure to the full configuration space to obtain the patches in the interval.

2 Criticality computation

The first stage of the algorithm computes the angles where the slice structure changes, which are called critical angles. Each slice is a planar subspace of configuration space in which the part translates at a fixed orientation. Its free space consists of planar regions and its contact space consists of loops of curve segments that bound these regions. Each segment is generated by a contact between a pair of part/obstacle features (vertices, edges, or faces). Feature pairs can also generate contact curves that lie wholly or partially in blocked space.

Two slices have the same structure when every component in each slice has a unique mate in the other slice with the same structure. Two components have the same structure when there exists a cyclic ordering of the second boundary for which every curve in each boundary has the same structure as the corresponding curve in the other boundary. Two curves have the same structure when they are generated by the same pair of part features.

The contact curves of polyhedra are linear. I require that the faces of the polyhedra be convex, which guarantees that each contact curve is a single line segment. A part with n features and an obstacle with m features generate nm segments in the worst-case, but only $O(n)$ when both are convex. As ψ changes, segments can enter and leave blocked space, but segments are never created or destroyed.

A necessary condition for a structure change at ψ_0 is that an endpoint of one contact segment

lies on another segment. If the condition does not hold, every pair of segments is either a positive distance apart or intersects at a point in the interior of both segments (transversality). Both these conditions persist in an interval around ψ_0 because the segments are continuous functions of ψ. The segments that lie in blocked space at ψ_0 are a positive distance from contact space, hence stay in blocked space throughout the interval. The remaining segments form the contact space throughout the interval. Each segment intersects two neighbors in its loop at ψ_0, hence intersects the same two segments throughout the interval. These conditions mean that all the slices in the interval have the same structure.

The necessary condition translates into an equation that can be solved for ψ. Contact segment endpoints have the form $p = a + R_\psi b$ where a and b are vectors of length two and R is the planar rotation operator. The conditions that endpoint p_3 lies on the segment $[p_1, p_2]$ are $(p_3 - p_1) \times (p_2 - p_1) = 0$ and $0 \leq (p_3 - p_1) \cdot (p_2 - p_1) \leq (p_2 - p_1)^2$. The equality simplifies to $k_1 \sin\psi + k_2 \cos\psi + k_3 = 0$ where the k_i are functions of the a_i and b_i constants. This equation yields at most two roots, for example using the substitution $t = \tan(\psi/2)$. The roots that satisfy the inequality are potential criticalities.

We now derive the endpoint expressions. There are three types of contacts: moving vertex/fixed face, moving face/fixed vertex, and moving edge/fixed edge. The endpoints of the first two types are configurations where the vertex lies on a boundary edge of the face. The endpoints of the third type are configurations where a boundary vertex of one edge lies on the other edge. Hence, every contact segment endpoint is a moving vertex/fixed edge contact or a moving edge/fixed vertex contact.

In the vertex/edge case, the vertex has global coordinates $[u, v, 0] + R^z_\psi p$ with $p = (p_x, p_y, p_z)$ the part coordinates of the vertex and with R^z the operator that rotates around the z axis in (x, y, z) space. The edge endpoints are a and b in global coordinates and its direction vector is $d = b - a$. The vertex lies on the line generated by a and b when $[u, v, 0] + R^z_\psi p - a = kd$, which states that $p - a$ is colinear with d. This is a system of three linear equations in u, v, k. If $d_z = 0$ there is no solution, otherwise $k = (p_z - a_z)/d_z)$ and

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a_x + kd_x \\ a_y + kd_y \end{bmatrix} - R_\psi \begin{bmatrix} v_x \\ v_y \end{bmatrix}.$$

The solution lies on the segment ab (as opposed to the underlying line) when $0 \leq k \leq ||d||$. The edge/vertex analysis is the same, except that a and b vary, while v is fixed. The equations are $v - [u, v, 0] - R^z_\psi a = kR^z_\psi d$ and the solutions are $k = (p_z - a_z)/d_z$ and

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \end{bmatrix} - R_\psi \begin{bmatrix} a_x + kd_x \\ a_y + kd_y \end{bmatrix}.$$

The algorithm enumerates all contact segment/contact endpoint triples, solves for ψ, and returns the sorted results. Partial information is used to reduce computation. Here are a few examples. Two features cannot generate a contact when their z ranges are disjoint. A vertex/face contact

cannot occur if an edge incident to the vertex has negative inner product with the outward face normal. A segment endpoint cannot fall on a segment if the local geometry intersects. Criticalities that pass these quick tests are tested for part overlap via a naive polyhedron intersection algorithm.

3 Contact graph construction

After the criticalities are computed, the contact graph is constructed. The patches are constructed independently in each interval (lb, ub) between adjacent criticalities. The midpoint slice, $\psi = 0.5 * (lb + ub)$, is constructed by my planar algorithm [4], which generates all contact curves, computes their arrangement by a line sweep, and returns the connected components where the parts do not overlap. The only modification is the module that constructs the contact segments. The construction is straightforward linear algebra and resembles Latombe's treatment [1].

The midpoint slice yields the patch structure in (lb, ub). Each contact segment lifts to a patch with the same pair of features in contact. The patch contact functions (implicit and parametric) are derived from the feature pair following Latombe. Segment adjacency transfers to the patches. The boundary curve between adjacent patches is computed from the parametric contact functions of the adjacent patches. The given ψ value is substituted into the functions to obtain two linear equations in u and v that are solved for the boundary curve point (u, v, ψ).

Every patch also has a lower boundary segment in the lb plane and an upper boundary segment in the ub plane. A patch can have zero, one, or more upper neighbors along an upper boundary segment and likewise below. These adjacencies are computed after all the patches are constructed.

4 Implementation

I have implemented the configuration space computation algorithm in Lisp and have validated it on moderate size examples. For example, a part with 26 features and an obstacle with 128 features takes between 100 and 200 milliseconds on a 933 Mhz Pentium 3 running linux. A C implementation would probably take 10 to 20 milliseconds based on my extensive porting experience.

References

[1] Jean-Claude Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1991.

[2] Elisha Sacks. Practical sliced configuration spaces for curved planar pairs. *International Journal of Robotics Research*, 18(1):59–63, January 1999.

[3] Elisha Sacks and Chandrajit Bajaj. Sliced configuration spaces for curved planar bodies. *International Journal of Robotics Research*, 17(6):639–651, 1998.

[4] Elisha Sacks and Leo Joskowicz. Computational kinematic analysis of higher pairs with multiple contacts. *Journal of Mechanical Design*, 117(2(A)):269–277, June 1995.