Computer Aided Geometric Design 31 (2014) 602–612

Contents lists available at ScienceDirect

Computer Aided Geometric Design

www.elsevier.com/locate/cagd

Guaranteed collision detection with toleranced motions

Hans-Peter Schröcker [*,1], Matthias J. Weber [1]

University of Innsbruck, Unit Geometry and CAD, Austria

ARTICLE INFO

Article history:
Available online 28 August 2014

Keywords:
Toleranced motion
Collision detection
Bounding ball
Bounding volume

ABSTRACT

We present a method for guaranteed collision detection with toleranced motions. The basic idea is to consider the motion as a curve in the 12-dimensional space of affine displacements, endowed with an object-oriented Euclidean metric, and cover it with balls. The associated orbits of points, lines, planes and polygons have particularly simple shapes that lend themselves well to exact and fast collision queries. We present formulas for elementary collision tests with these orbit shapes and we suggest an algorithm, based on motion subdivision and computation of bounding balls, that can give a no-collision guarantee. It allows a robust and efficient implementation and parallelization. At hand of several examples we explore the asymptotic behavior of the algorithm and compare different implementation strategies.

1. Introduction

Collision detection between moving solids (Ericson, 2005) is a ubiquitous issue in robotics and other fields, such as computer graphics or physics simulation. General-purpose strategies are available as well as algorithms tailored for specific situations. In general, any algorithm is a trade-off between reliability, accuracy, and computation time. In some scenarios, for example in trajectory planning for high-speed heavy robots, collisions might have devastating effects. Therefore, it is necessary to exclude them with certainty. This article presents an efficient algorithm that can give such a guarantee for collisions between polyhedra under certain assumptions on the motion.

Typical techniques of collision detection include time discretization (Ericson, 2005, Section 2.4), motion linearization, and the use of bounding volumes (Ericson, 2005, Chapter 4) or hierarchies of bounding volumes (Ericson, 2005, Chapter 6) and (Hubbard, 1995; Dingliana and O'Sullivan, 2000). Careful application of discretization and linearization is critical, as collisions with a short time interval of interference may be missed. The algorithm we propose in this article uses hierarchies of bounding volumes but it exhibits some specialties with respect to their construction. When using bounding volumes, several potentially contradicting requirements have to be fulfilled (Ericson, 2005, p. 76): the computation of the bounding volumes must be efficient, the representation of moving and fixed object by not too many bounding volumes has to be accurate, and the collision queries for bounding volumes should be fast.

In this article, we present ideas for a collision detection algorithm that satisfies these criteria, at least to a certain extent. The novelty is to use a ball covering of the motion which we regard as a curve in the 12-dimensional space of affine displacements. This ball covering can be converted into a system of simple bounding volumes of elementary objects (points, lines, planes, and polygons). Unlike other methods for collision detection based on bounding balls, this requires only the

* Corresponding author.
E-mail addresses: hans-peter.schroecker@uibk.ac.at (H.-P. Schröcker), matthias.weber@uibk.ac.at (M.J. Weber).
[1] Tel.: +43 512 507 61202; fax: +43 512 507 61299.

http://dx.doi.org/10.1016/j.cagd.2014.08.001
0167-8396/

covering of a one-dimensional object (the motion) as opposed to the covering of a 3D-solid. The bounding volumes are constructed from the geometry of the moving object *and* its motion and thus accurately represent the portion of space occupied by the object as it moves. It is possible to refine the bounding volumes by subdividing the motion. With every subdivision step, the number of bounding volumes doubles while their volume shrinks roughly by the factor 2^{-3}. This is much better than for ball coverings of non-curve like objects in 3D. Furthermore, the worst case complexity of subdivision is usually not achieved because large parts of motion or moving objects can be pruned away at a low recursion level.

We expect our algorithm to work particularly well for objects that lend themselves to polyhedral approximations. Other desirable properties include:

- The possibility to report approximate time and location of potential collisions,
- simple, fast and robust implementation because of recursive function calls without the need of sophisticated data structures,
- the use of fast and robust primitive procedures which are well-known in computational geometry, and
- the possibility of parallelization.

Assuming a robust numerical implementation, collisions up to a pre-defined accuracy will always be detected correctly and reported non-collisions are guaranteed.

The ball covering in the space of affine displacements is based on an underlying object-oriented Euclidean metric (Belta and Kumar, 2002; Chirikjian and Zhou, 1998) which is derived from a mass distribution in the space of the moving object. This mass distribution already saw applications in computational kinematics, for example Hofer et al. (2004), Nawratil (2007), Schröcker et al. (2009). One of its important properties, the simplicity of certain orbit shapes with respect to balls of affine displacements, was observed in Schröcker and Wallner (2005). It is our aim, to exploit this for guaranteed collision detection under certain not too restrictive assumptions on the motion.

Since collision detection is a well-explored research topic, we want to compare the algorithm we propose with other approaches known from literature.

There is a number of techniques for detecting collisions or computing the distance of moving objects by means of covering balls or hierarchies of covering balls (Hubbard, 1995; Dingliana and O'Sullivan, 2000; Quinlan, 1994; Greenspan and Burtnyk, 1996; Martínez-Salvador et al., 1998; Weller and Zachmann, 2009a, 2009b). We cover the moving objects by different bounding volumes (portions of balls and hyperboloids of one or two sheets). However, our construction is based on a covering of a curve in $\mathbb{R}^{12}$ (the motion) by balls. This latter problem is typically simpler and of lesser complexity than constructing ball coverings for 3D solids. Depending on the particular situation, certain curve specifics, for example piecewise rationality, can be used to efficiently generate systems of bounding balls. In contrast to collision detection with covering balls, our algorithm typically requires exactly one quadratic bounding surface (sphere or hyperboloid) per vertex, edge, and face. As in case of bounding spheres, efficient collision queries are possible by evaluation of quadratic functions only.

Several authors suggest bounding volume shapes which are specifically tailored to the needs of collision detection. Typical examples are the generalized cylinders of Martínez-Salvador et al. (1998) and the s-topes of Bernabeu et al. (2001). Usually, they are derived from some ball covering. An s-tope is, for example, the convex hull of a finite set of spheres. These shapes visually resemble our bounding volumes. The crucial difference is that our construction takes into account the objects' motion in such a way that it is guaranteed to stay within the volume *during the whole motion.* In this sense, it resembles the space–time solids of Hubbard (1995) or the implementation for jammed random packing of ellipsoids described in Donev et al. (2005a, 2005b). A non-collision guarantee is often possible by means of a coarse representation. Only if a collision cannot be excluded, the bounding volumes are adaptively refined. The prize we pay for this exact collision detection with arbitrary accuracy is the necessity to re-compute the bounding volumes if the motion changes.

It has to be emphasized that our approach requires no time-discretization or motion linearization. We work with the motion as it is analytically defined over a parameter interval. Moreover, our algorithm should not be confused with algorithms for (approximate) distance calculation although it can report whether or not the distance falls beyond a certain threshold during a given motion. Naturally, our algorithm is considerably slower than many of the sophisticated existing algorithms but its outcome is of a high quality. A reported no-collision is guaranteed. Our algorithm should only be used when such a certification is crucial.

The remainder of this article is organized as follows. After presenting some preliminaries about the object-oriented metric in Section 2, we describe the orbits of geometric primitives in Section 3 and their basic collision tests in Section 4. In Section 5 we present some ideas for constructing bounding balls for motions (curves in high-dimensional Euclidean spaces) and in Section 6 we synthesize everything via motion subdivision into a collision detection algorithm. Section 7 presents examples and timing results.

2. Distance between affine displacements

Measuring the "distance" between two displacements is a fundamental problem of computational kinematics with many important applications. It is also afflicted with irresolvable defects such as the incompatibility of units in angle and lengths and dependence on the chosen coordinate frames. For our algorithm, we require a Euclidean metric in the space of Eu-