

Dynamic and Exact Collisions Detection for Paths

Jeff M. Phillips

April 29, 2005

A common subroutine in robotics, especially path planning, is to check for collisions between the robot and obstacles along a planned route of the robot. In fact, this is often the bottleneck in most path planning algorithms. Until recently, the most common collision detectors would statically check a 'soup' of triangles for intersection by using a bounding box hierarchy. However, in reality spaces are seldom static. Furthermore, these techniques do not guarantee that if they return that a path is collision free that it indeed is collision free. The problem is that these algorithms subdivide the configuration space \mathcal{C} while checking for collisions in the workspace \mathcal{W} .

Recently the *Stanford Collision Detector*[1] was introduced which efficiently checks for collisions along a dynamic path (where the shape of the robot is changing and the obstacles are moving) which guarantees that if it returns a path is collision free it is indeed collision free. In addition, this method is much faster in practice than other more refined static techniques which harness the power of graphical processing units. The key to this algorithm is a simple geometric argument bounding the difference between distances in \mathcal{C} and in \mathcal{W} .

References

- [1] Fabian Schwarzer, Mitul Saha, and Jean-Claude Latombe. Exact collision checking of robot paths. *WAFR*, 2004.