

# Continuous Collision Detection and Physics

Erwin Coumans, Sony Computer Entertainment  
August 2005, Draft revision 5

## 0 Overview

A novel approach to Angular and Linear Continuous Collision Detection is presented. It is a form of Conservative Advancement, and can be viewed as a generalization of a raycast against a deforming Minkowski sum [2]. The in-between motion happens either in world space, or in relative space with a deforming Minkowski sum.

## 1 Introduction

The Time of Impact (TOI) Problem can be stated as:

*Given: The current transformations and velocities of two convex bodies.  
Compute: A lower bound on the time of impact of the two bodies, assuming they continue their current trajectories.*

In a lot of game environments the TOI can be estimated by a raycast or a linear convex cast, like a swept sphere. This can be sufficient to prevent missing collisions and leaving the environment. We present a method that calculates the TOI exactly, within any chosen accuracy. This makes it applicable to a wider range of scenarios.

Two different solutions to compute the TOI are discussed in this paper. One solution is based on Brian Mirtich's [7] concept of Conservative Advancement. The linear convex cast as described by Gino van den Bergen is an specific case of this. The other solution is featured based, and calculates the TOI per feature pair. In particular we discuss Redon's Algebraic Continuous Collision Detection method.

In order to use the TOI in a physics engine, we need to make sure that the motion used by the TOI estimation is consistent with the motion used by the physics engine integrator. For example, when relative screwing motion is used [10], the in-between motion is very different from an ordinary Euler integration scheme with piece-wise constant linear and angular velocity. This means that when more then two objects are involved, the relative motion can not be used.

But even if we choose constant linear and angular velocities for the in-between motion, we have to be careful using relative motion. Reason for this is:

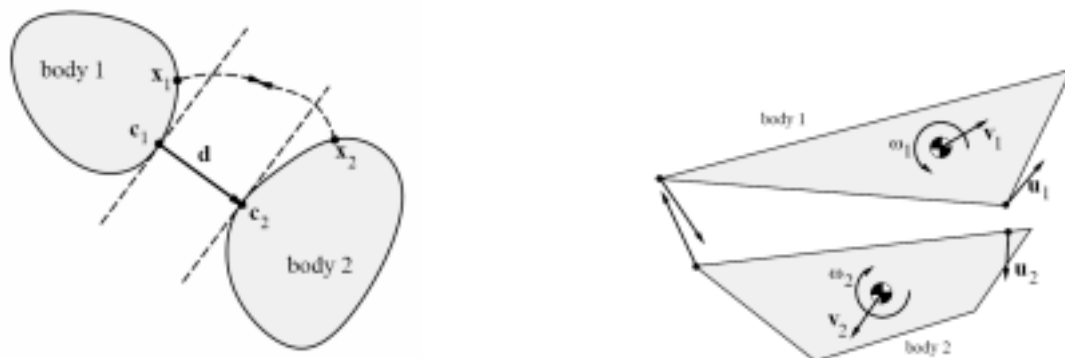
*For two objects with constant linear and angular motion, its combination doesn't have this property of constant angular and linear velocity, as Stephane Redon pointed out. <todo: proof?>*

Hence we propose using world space motion, or a special version of the Minkowski sum where during every iteration both objects orientations is updated. This means deformation of the Minkowski sum. We don't build this sum explicitly, instead the

sum is sampled using the support mapping. Due to updated transforms for every iteration, this support mapping changes.

## 2 Previous Work

Brian Mirtich describes the Conservative Advancement method in his PhD thesis, section 2.3.2. In addition to the current transformation and velocities, he also keeps track of the current closest points. Given  $D$  the upper bound formula for the distance travelled by any point on the body projected on the separating vector  $c_2 - c_1$ . The TOI can be calculated by solving the formula  $D1(t) + D2(t) = d$ . Mirtich uses ballistic motion to describe  $D$ .



Instead of ballistic motion, Gino van den Bergen [2] uses pure linear motion for  $D$ , and transforms the problem into a raycast on the Minkowski sum.

Another approach for calculating the TOI is by iterating over all feature pairs, one from each object, and taking the minimum TOI of all those feature pairs. The only feature pairs that need to be taken into account are Vertex of one object versus the Face of the other objects, or an Edge of one object versus an Edge of the other object. John Canny [3] finds the TOI for the feature pairs by numerically solving a low order polynomial. He assumes constant angular velocity.

Redon [10] introduces screwing motion and this allows for solving the polynomial algebraically. For each feature pair (Edge-Edge and Vertex-Face), an algebraic formula is derived that takes angular and linear motion into account. The roots of this algebraic formula, a polynomial of degree 3 after reduction, represent the possible TOI's for this feature pair. In later work Redon uses Interval Arithmetic and global motion instead of relative motion for aforementioned reasons.

## 4 Conservative Advancement and Algebraic Rootfinding

Our contribution is a variant of Brian Mirtich's Conservative Advancement method. Instead of using ballistic motion, we use constant linear and angular velocity to describe  $D$ . Similar to van den Bergen [2], we use the Minkowski Sum of both bodies, so we work on relative linear velocities. This means instead of solving the  $D1(t) + D2(t) = d$  we solve for  $Drel(t) = d$ . We either keep the transforms for both objects in world space, or as alternative we use relative positions but absolute

orientations. In both cases the Minkowski Sum might deform due to rotations of the involved objects.

The maximum distance from any point on the body to the bodies center of mass,  $r_{max}$ , is precomputed for each body. Along with the constant angular velocity  $w$  this angular component of  $D$  is  $r_{max} * w$ , so the  $D$  is defined as  $v \cdot n + r_{max} * w$ . Due to the conservative nature of each step, each iteration will either bring  $x$  closer to  $c$  or terminate.

*Algorithm 1: Iterative Method for finding the Time of Impact (lambda) between two bodies, represented as their Minkowski Sum. In the algorithm  $x$  stands for relative transform,  $n$  for normal,  $c$  for closest point,  $\lambda$  is the Time of Impact fraction in range  $[0..1)$ . Notice that the Minkowski Sum changes during the iterations due to rotation.*

```

 $\lambda \leftarrow 0$ ;  $x \leftarrow s$ ;  $n \leftarrow 0$ ;
 $c \leftarrow$  "the point of C closest to  $x$ ";
while not " $x$  is close enough to  $c$ " do
  begin
     $n \leftarrow x - c$ ;
     $dist \leftarrow |n|$ 
    if  $n \cdot r \geq 0$  and  $dist > r_{max}$  then
      return false
    else
      begin
         $D_{rel} = v \cdot n + w * r_{max}$ 
         $\lambda \leftarrow \lambda + dist / D_{rel}$ ;
         $x \leftarrow$  "interpolate( $\lambda$ )";
         $c \leftarrow$  "the point of C closest to  $x$ "
      end
    if  $\lambda > 1$ 
      return false
  end;
return true

```

[Algebraic implementation: todo]

## 5 Persistent Manifold and Contact Tolerances

Adam Moravanzky describes a method for persistent manifold generation in [8]. Contact points generated using Discrete or Continuous Collision Detection are added to a Manifold and are kept persistent over frames. As a rough approximation of the 2D convex hull, a 2D Axis Aligned Bounding Box (AABB) is formed in the manifold plane. Only a small fixed number of points with maximum projections along this AABB's axis are kept, in order to reduce the number of contact points. Also the

contact point with the deepest penetration depth is kept. We successfully used this approach in Bullet. But instead of using an AABB we take the collection of 3 points with maximum area. In our experiments this simplification provides enough quality for stable RigidBody Dynamics with stacking.

Similar to Conservative Advancement, the contact points penetration depth is updated using the relative motion projected on the contact normal. As described in Redon, several tolerances are kept, one breaking tolerance and a tolerance to avoid the more expensive penetration depth case. As we provide Retroactive Detection (penetration recovery) we don't use the rescue tolerance. This allows for a hybrid system with Conservative Advancement for fast moving bodies, and Retroactive Detection for slow moving bodies. A similar Hybrid system is described in Baraff [1] where he combines Regula Falsum and Bisection to estimate the TOI.

## 6 Results and Conclusion

The methods are implemented in the Bullet Continuous Collision Detection library, which is available from [www.continuousphysics.com/Bullet](http://www.continuousphysics.com/Bullet). The project is open source and free for commercial use. Bullet has been successfully used in RigidBody Dynamics Simulation, with stable efficient stacking, and preventing the bullet-through-paper problem. The Simulator uses Gauss Siedel or Succesive Overrelaxation Iterative LCP solver as described in K. Erleben [5]. We used Jefferson's optimistic scheduling approach described in Time Warp [7] to allow for Parallel scheduling of Time of Impact discontinuities.

<todo: timings, other comparisions and parallel implementation details>

## 7 References

- [1] Baraff Siggraph 1990, Computer Graphics, Volume 24, Curved Surfaces and Coherence for Non-penetrating Rigid Body Simulation.
- [2] Gino van den Bergen "Ray Casting against General Convex Objects with Application to Continuous Collision Detection", GDC 2005. [www.dtecta.com](http://www.dtecta.com)
- [3] Canny, Collision Detection for Moving Polyhedra, 1986. IEEE Transaction on Pattern Analysis and machine Intelligence, 8(2).
- [4] Michael Cline 2003, "Post Stabilization for Rigid Body Simulation with Contact and Constraints"
- [5] Kenny Erleben 2005, PhD Thesis "Stable, Robust, and Versatile Multibody Dynamics Animation", <http://www.diku.dk/forskning/image/research/opentissue/>
- [7] Brian Mirtich 1996, PhD Thesis, "Impulse Based Dynamic Simulation of Rigid Body Systems"
- [8] Brian Mirtich ,2000, "Time Warp Rigid Body Simulation"

[9] Adam Moravanszky and Pierre Terdiman, 2004. Fast contact reduction for dynamics simulations. Game Programming Gems 4. Charles River Media.

[10] Stephane Redon, 2000, “Algebraic Solution to the Problem of Collision Detection for Rigid Polyhedral Objects”, <http://www.inrialpes.fr/i3d/people/redon>