

Towards Collision Detection and Constraints in Rigid based Animation

Juraj Onderik¹ and Roman Ďurikovič¹

¹Department of Applied Computer Science
Faculty of Mathematics, Physics and Informatics
Comenius University, Bratislava, Slovakia

Abstract

Realistic animation of rigid multi-body systems represents an integral part of the animation pipeline used in the majority of today's virtual reality systems. To achieve real-time behavior, interactive environments often require fast, robust and scalable techniques for collision, constraint detection, resolution during the dynamic simulation. In this tutorial we cover several major approaches, issues and problems emerged when developing solid, physically based, rigid body animation kernel. We explain advanced modular design of the animation pipeline, which breaks the simulation process into several independent modules and thus simplifies the overall complexity of this process.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computational Geometry and Object Modeling]: Physically based modeling

1. Introduction

Together with other natural phenomena, fast, stable and realistic animation of rigid bodies is still a challenging problem. [Erl05, Gue06]. (see figure 1)

Today's virtual realities consist of enormous number of dynamic objects, however the majority of them are still *rigid bodies*. The "rigidity" concept states, that their shape never changes during the dynamic simulation or more precisely they are "infinitely rigid". Although, this idealization not corresponds with proper laws of physics, it helps us to lower the computational cost of the simulation and achieve visually plausible results.

The overall research around physically based animation of rigid bodies consists of two main parts, namely the *collision detection* and *rigid body dynamics*. Collision detection usually solves a pure geometrical problem of object interpenetration, whereas in dynamic simulation we compute proper motion of objects undergoing several collisions and constraints, while modeling frictional effects and other artifacts (e.g. gravity, wind, user interaction etc.).

Both of these topics have a long research history.

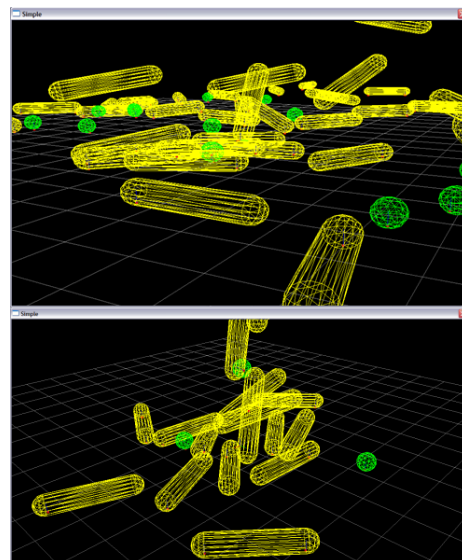


Figure 1: Sample demonstration of dynamic animation of rigid bodies.

2. Summary

The concept of this tutorial is to split the rigid body animation pipeline into several almost independent components and modules. For each part of the modular design we define its functionality, key concepts and provide a detailed description of several important techniques necessary for an efficient implementation.

The overall process of the physically based animation of rigid bodies can be roughly expressed as an infinite loop of the three main stages. Each stage is then implemented in its respective component.

First part of the tutorial describes the *User Interaction Component*, where we collect input from users and present the outcome of the previous animation frame. We assume that the animation is simply a sequence of successive frames during this course. In each frame the state (position, orientation etc.) of all objects should be computed. For simplicity, we do not describe any particular user interface, conversely all required user input is described within other components.

We proceed with the *Collision Detection Component*, where we test objects for collisions, i.e. we quickly identify set of bodies in close proximity or interpenetration. We describe how to split this component into three basic layers (modules), where each one is responsible for a more refined collision detection. We start with *Broad Phase* module, where we approximate objects with simple bounding volumes and establish efficient conservative algorithms, which prune pairs of distant objects. Next, we compare various hierarchies of bounding volumes, which are the essential part of the *Mid Phase* module. Finally, in the *Narrow Phase* module we explain fast and robust algorithms for exact collision queries of convex polytopes and show the application of *Signed Distance Maps* to non-convex objects. We also propose how to gather additional contact information to correctly resolve these collisions.

In the last part of the tutorial we provide the *Simulation Component*, which is responsible for handling collision and user-defined constraints or any external influence using Newtonian dynamics is described. Again we split this component into three main modules, each focusing on different aspects of the resolution process. At this part, we start with the *Motion Solver* module, where we describe the simplest unconstrained motion of rigid bodies, i.e. they are influenced only with external forces such as gravity or wind forces. We also derive necessary properties of rigid body dynamics. Next, in the *Collision Solver* module we concentrate on a short *impulse-based* collision resolution, where we explain several *algebraic* and *incremental* collision laws. Finally, in the *Constraint Solver* we focus on the *constraint-base* multi body dynamics, where we handle 6-DOF user-defined joint constraints and long-term resting contact constraints, using efficient iterative MLCP solver.

3. Proposed Length

We recommend this tutorial as half-day tutorial.

4. Topics Covered

- Hierarchical Grids and Octrees
- Semi-Adjusting BSP Tree
- Sweep and Prune (SAP)
- Bounding Volume Hierarchies
- k-Discrete Orientation Polytopes (kDOPs)
- Oriented Bounding Boxes (OBB)
- Quantized Orientation Slabs with Primary Orientations
- Bounding Spheres (BS)
- Swept-Sphere Volumes (SSV)
- Minkowski Space
- Proximity Queries
- The GJK Algorithm
- The Expanding Polytope Algorithm (EPA)
- The V-Clip Algorithm
- Dobkin-Krikpatrick Hierarchies (DKH)
- Signed Distance Maps (SDM)
- Coulomb Friction Model
- Impulse Space
- Algebraic and Incremental Collision Resolution
- Spatial Notation
- Unilateral and Bilateral Constraints
- System Jacobian
- Iterative MLCP Solver
- Friction Pyramid
- Kinematic Constraint and Contact Jacobians
- Symmetrization of System Matrix
- 6-DOF Joint Constraint

5. Tutorial Outline

According to the structure of the enclosed course notes, this tutorial consists of the following seven main parts.

Introduction to Rigid Body Animation Pipeline. The majority of the rigid body animation pipelines follow similar concepts and strategies. According to the original work [Eri02] in this part we define a modular design of a generic rigid body animation pipeline and present important principles, problems and paradigms about each module.

Broad Phase Collision Detection Techniques. Every broad phase collision detection technique should compute pairs of objects in the scene that are in close proximity. Given N objects in the scene, in the worst case this can be an $O(N^2)$ problem. However, in practice this almost never happen. In this part of the tutorial we describe simple regular partitioning schemes based on *Hierarchical Hash Tables* and *Loose Octrees* [Mir96, THM*03, Tha00]. Next we present the standard *Sweep and Prune (SAP)* technique based on the dimension reduction principle [Bar92, CLMP95] and finally introduce a more complicated spatial decomposition called *Semi-Adjusting BSP Trees* [LCF05]

Mid Phase Collision Detection Techniques. In this part we adopt the general approach for data localization known as the *bounding volume hierarchy (BVH)*. First we design a theoretical model of a generic BVH, which is a useful tool for a comparison of various real hierarchies. Next we briefly describe several state-of-the-art BVHs including *k-Discrete Orientation Polytopes (kDOPs)* [KHM*98], *Oriented Bounding Boxes (OBB)* [GLM96], *Quantized Orientation Slabs with Primary Orientations* [He99], *Bounding Spheres (BS)* [Bra02] and *Swept-Sphere Volumes (SSV)* [LGLM00]. For each hierarchy we show the choice of the *bounding volume*, the *hierarchy construction design*, the *collision query algorithm* and the *hierarchy update* (due to motion transformations).

Narrow Phase Collision Detection Techniques. We explain here how to find exact colliding pairs of objects and how to compute necessary contact (proximity) information. Assuming that the mid phase module has decomposed object geometry into convex subparts, we focus here on the well known convex-based algorithms. We start with the family of simplex based *Gilbert, Johnson and Keerthi (GJK)* algorithms [vB99, vB01] and continue with a modified feature based distance computation algorithm called *The Voronoi Clip (V-Clip)* [Mir98]. As an alternative for highly non-convex objects we introduce *Signed Distance Maps (SDM)* [JBŠ06]. We end up this part with simple projection and tracking based algorithms for contact generation.

Solving Unconstrained Motion of Rigid Bodies. This part starts the second half of the tutorial. Here we try to briefly reintroduce necessary physical properties of rigid bodies [DB03] and derive the well known Newton-Euler equation of unconstrained motion. From the users point-of-view the overall animation consists of a sequence of successive frames. However, the actual simulation process is far more complex. We show, that each frame must be split into an appropriate number of sub-frames, divided into several sub-steps. Therefore, at the end of this part we suggest some natural subdivision schemes of the animation time line.

Solving Collisions between Rigid Bodies. Given proper collision/contact information from the collision detection module, we focus in this part on the computation of forces and impulses that prevent objects from non-realistic interpenetration while modeling frictional effects. Trying to find a balance between the computational costs and physical correctness we further discuss three usual collision resolution strategies [Mir96]: simple *Algebraic Resolution* and more complex *Incremental Resolution*. However, a detailed description of the physically correct *Full Deformation Resolution* is beyond the scope of this course.

Solving Constraints between Rigid Bodies. The last part of this tutorial consider the most complex scenario, where objects rest onto each other or have some user-defined constraints. We reformulate the motion equation of single rigid body to the whole multi body system. Using the spatial no-

tion we successively develop several contact and joint constraints and express the solution of the underlying equations as an *Mixed Linear Complementarity Problem (MLCP)*. Finally we discuss several efficient iterative MLCP solvers. [Erl05]

6. Prerequisites

The participant should be familiar with the concepts of hierarchical spatial data structures, basic properties of rigid body dynamics and numerical solutions of Ordinary Differential Equations (ODE) and Linear Complementarity Problems (LCP).

7. Organizer

Roman Ďurikovič

Department of Applied Computer Science
Faculty of Mathematics, Physics and Informatics
Comenius University, Bratislava, Slovakia

Mlynska dolina
842 48 Bratislava
Slovakia

phone: +421 2 60295879

fax: +421 2 65422263

mail: durikovic@fmph.uniba.sk

web: <http://www.sccg.sk/~durikovic>

8. Speaker

Roman Ďurikovič obtained his RNDr. in Numerical Analysis from Comenius University in Slovakia 1998, Ph.D. in Computer Science from the Hiroshima University in Japan in 1996. He has been an invited researcher at the Kyushu University, Japan and invited professor at the University of Aizu in Japan. Currently he is an associate professor of Computer Graphics at Comenius University, and at University of Saint Cyril and Metod, Slovakia. His research interests include graphics, rendering with GPU, shape modeling, and physical based animations. Dr. Durikovic serves as reviewer for following conferences GRAPHICON Russia, CGI, SCCG Slovakia, WSCG Czech Republic, GRAPP Spain, and EUROGRAPHICS.

Juraj Onderik received his MSc degree in Applied Computer Science from the Comenius University in 2006. He is currently a PhD student at the department of Applied Computer Science at Comenius University and a software developer at Abyss Studios Ltd. His research is focused on collision detection, physically based simulation of rigid bodies and fluids. During his studies he has organized and tutored several lectures in computer science for various audience.

9. Course Notes Description

Course notes for this tutorial are based on the state of the art report required for doctoral degree of Juraj Onderik. All additional materials and bibliographical references can be found in the course notes as well.

References

- [Bar92] BARAFF D.: *Dynamic Simulation of Nonpenetrating Rigid Bodies*. PhD thesis, Department of Computer Science, Cornell University, Ithaca, NY 14853-7501, 1992.
- [Bra02] BRADSHAW G.: *Bounding Volume Hierarchies for Level-of-Detail Collision Handling*. PhD thesis, Dept. of Computer Science, Trinity College Dublin, 2002.
- [CLMP95] COHEN J. D., LIN M. C., MANOCHA D., PONAMGI M.: I-collide: An interactive and exact collision detection system for large-scale environments. In *SI3D '95: Proceedings of the 1995 symposium on Interactive 3D graphics* (New York, NY, USA, 1995), ACM Press, pp. 189–196, 218.
- [DB03] DAVID BARAFF A. W.: Physically based modeling. In *Proceedings of SIGGRAPH 2003* (2003).
- [Erl02] ERLEBEN K.: *Module Based Design for Rigid Body Simulators*. Tech. Rep. DIKU 02/06, Department of Computer Science, University of Copenhagen, 2002.
- [Erl05] ERLEBEN K.: *Stable, Robust, and Versatile Multi-body Dynamics Animation*. PhD thesis, Department of Computer Science, University of Copenhagen (DIKU), Universitetsparken 1, DK-2100 Copenhagen, Denmark, March 2005.
- [GLM96] GOTTSCHALK S., LIN M. C., MANOCHA D.: Obbtree: A hierarchical structure for rapid interference detection. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1996), ACM Press, pp. 171–180.
- [Gue06] GUENDELMAN E.: *Physically-Based Simulation Of Solids And Solid-Fluid Coupling*. PhD thesis, Department of Computer Science, Stanford University, June 2006.
- [He99] HE T.: Fast collision detection using quospo trees. In *Symposium on Interactive 3D Graphics* (1999), pp. 55–62.
- [JBŠ06] JONES M. W., BÆRENTZEN A., ŠRÁMEK M.: Discrete 3D distance fields: Techniques and applications. *IEEE Transactions on Visualization and Computer Graphics* 12, 4 (2006), 581–599.
- [KHM*98] KLOSOWSKI J. T., HELD M., MITCHELL J. S. B., SOWIZRAL H., ZIKAN K.: Efficient collision detection using bounding volume hierarchies of k -DOPs. *IEEE Transactions on Visualization and Computer Graphics* 4, 1 (1998), 21–36.
- [LCF05] LUQUE R. G., COMBA J. L. D., FREITAS C. M. D. S.: Broad-phase collision detection using semi-adjusting bsp-trees. In *SI3D '05: Proceedings of the 2005 symposium on Interactive 3D graphics and games* (New York, NY, USA, 2005), ACM Press, pp. 179–186.
- [LGLM00] LARSEN E., GOTTSCHALK S., LIN M., MANOCHA D.: Fast distance queries with rectangular swept sphere volumes. In *IEEE International Conference on Robotics and Automation* (San Francisco, CA, USA, 2000), vol. 4, pp. 3719–3726.
- [Mir96] MIRTICH B.: *Impulse-based Dynamic Simulation of Rigid Body Systems*. PhD thesis, Dept. Elec. Engin. Comput. Sci., Univ. California, Berkeley, CA, 1996.
- [Mir98] MIRTICH B.: V-Clip: Fast and robust polyhedral collision detection. *ACM Trans. Graph* 17, 3 (1998), 177–208.
- [Tha00] THATCHER U.: Loose octrees. In *Game Programming Gems*. Charles River Media, 2000, ch. 4, pp. 444–453.
- [THM*03] TESCHNER M., HEIDELBERGER B., MUELLER M., POMERANETS D., GROSS M.: Optimized spatial hashing for collision detection of deformable objects. In *Proceedings of Vision, Modeling, Visualization VMV'03* (Munich, Germany, November 2003), pp. 19–21.
- [vB99] VAN BERGEN G.: A fast and robust GJK implementation for collision detection of convex objects. *Journal of Graphics Tools: JGT* 4, 2 (1999), 7–25.
- [vB01] VAN BERGEN G.: Proximity queries and penetration depth computation on 3d game objects. In *Proc. Game Developers Conf.* (2001).