Interactive and Exact Collision-Detection Systems

Department of Computer Science   University of North Carolina at Chapel Hill   February 2004

The Challenge

Collision detection is a fundamental problem in computer animation, computer graphics, physically based modeling, and robotics. For applications and simulations in these domains to be convincing, they should not only render realistic images but should also precisely model object interactions such as pushing, striking, or deforming other objects. Detecting collisions, computing distances, and determining exact contact manifolds are of fundamental importance to the ability to model these interactions accurately. Researchers have written extensively about the issues concerning collision detection and distance computation, but actually building a general-purpose, robust, and efficient system for complex scenes and deformable objects remains an outstanding research challenge.
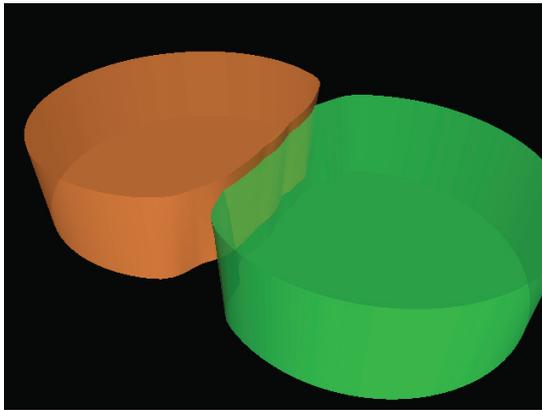
Proximity information being used for collision response between dynamically deformable bodies.

The Approach

We have developed efficient algorithms for collision detection between polygonal and curved models in large environments. Different algorithms have been proposed for convex polyhedra, unstructured models (polygon soups), curved models represented as NURBS, and large environments composed of thousands of moving objects. These algorithms utilize temporal and spatial coherence and hierarchical representations. Based on these algorithms, we have developed multiple proximity systems: I-Collide, RAPID, V-Collide, PQP, PIVOT, SWIFT, and DEEP.

I-Collide is an interactive and exact collision-detection library for environments composed of convex polyhedra. It quickly determines the contact status by exploiting special characteristics of convex polyhedra. It takes advantage of temporal coherence, so that the collision queries are extremely fast when the models move only a little between successive frames. The algorithm uses a sorting-based sweep-and-prune technique. To begin with, the number of object pair interactions is reduced to only the pairs within close proximity by sorting bounding boxes surrounding the objects. Then for each pair with overlapping bounding boxes, I-Collide determines whether

Highlights

- **Interactive and exact proximity query algorithms and systems for complex, dynamic virtual environments, and electronic prototyping.**

- **Use of locality, coherence, and hierarchical data structures to achieve efficiency and accuracy simultaneously.**

- **Widely used by researchers and commercial vendors in a number of applications including robot motion planning, dynamic simulation, haptic rendering, virtual prototyping, surgical simulation, and computer games.**

or not the actual objects are colliding by traversing the external Voronoi region of each object.

RAPID is a robust and accurate polygon collision detection library for unstructured models. RAPID constructs a hierarchy of oriented bounding boxes (OBBs) around the input triangles of a model. It then follows a two-stage approach to detect collisions: a hierarchical OBB test to find possibly colliding pairs of triangles, followed by an exact test to determine whether or not the pair of triangles actually overlaps. RAPID is numerically robust; the algorithm is not subject to conditioning problems and requires no special handling of degenerate cases.

V-Collide is a collision detection library for managing a large number of unstructured models. It uses a three-stage collision detection architecture. First it constructs axis-aligned bounding boxes around each model and uses a sorting-based sweep-and-prune algorithm to find possibly colliding pairs of objects. The final two stages are the same as RAPID. V-Collide uses coherence between successive time steps of a simulation; hence it performs well in animations and motion simulations

PQP is a proximity query package for unstructured models. Besides collision detection, it can also compute the Euclidean minimum distance as well as the approximate
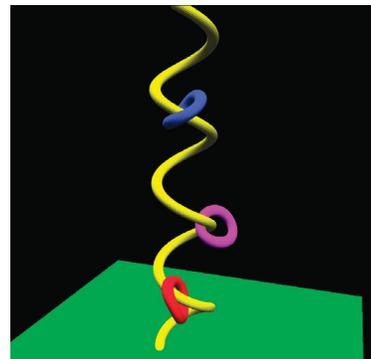
A frame from a dynamic simulation of non-convex rigid bodies.

distance between objects undergoing rigid motion. The system makes use of a new proximity query algorithm that uses a hierarchy of swept sphere volumes to accelerate these queries. It also makes use of coherence and priority directed searches to further enhance performance.

PIVOT is a general proximity query engine that performs queries using a hybrid geometry and image-based approach that balances computation between the CPU and graphics subsystems. Geometric object-space techniques are used to localize potential intersection regions between two objects, and image-space techniques compute the low-level proximity information in these regions. Our algorithm relies on the computation of graphics hardware-accelerated distance fields using multi-pass rendering techniques. The main features of our approach include a unified framework for all proximity queries (collisions, Euclidean distance, penetration depth computation etc.), generality to non-convex objects, no required pre-computation or complex data structures, computational efficiency allowing interactive queries on current PCs, robustness requiring no special-case handling of degeneracies, portability across various CPU/graphics combinations, and error-bounds on the approximations.

SWIFT is an exact collision-detection library for environments composed of general polyhedral solids undergoing rigid motion. It handles multiple types of queries such as intersection detection, distance computation, and contact determination, in a unified framework. It first decomposes the input polyhedra into convex surface elements. It uses these elements along with a hierarchy of convex hulls to quickly answer any of the queries. In particular the contact determination query can find local minima in the distance function between two polyhedra quickly.

DEEP is a technique based on Dual-space Expansion for Estimating the Penetration depth between convex polytopes in 3D. The algorithm incrementally seeks a "locally optimal solution" by walking on the surface of the Minkowski sums. The surface of the Minkowski sums is computed implicitly by constructing a local Gauss map. In practice, the algorithm works well when there is high motion coherence in the environment and is able to compute the optimal solution in most cases.

System Availability. The source code for some of the systems is available for noncommercial use via ftp or the Web. For details, please visit gamma.cs.unc.edu/collide

Technology Transfer. More than 5,000 users have downloaded these systems from our Web site. The technology has been transferred to a number of commercial vendors, including ADAC Labs, Blaxxun, Division LTD, Ford, Intel Corp., Kawasaki, Mechanical Dynamics, Oz.com, and a number of game developers.

Project Members
Ming C. Lin, professor
Dinesh Manocha, professor
Young Kim, postdoctoral researcher
Kenny Hoff, graduate student

Past Research Assistants
Jonathan Cohen, Stephen Ehmann, Stefan Gottschalk, Arthur Gregory, Tom Hudson, Shankar Krishnan, Eric Larsen, Ajith Mascarenhas, Gopi Meenakshisundaram, Amol Pattekar, M. K. Ponamgi, Andrew Wilson, Andrew Zaferakis

Research Sponsors
Intel Corp.; National Science Foundation; Office of Naval Research; U. S. Army Research Office

Selected Publications
Kim, Y., M. C. Lin, and D. Manocha. "DEEP: Dual-space Expansion for Estimating Penetration Depth Between Convex Polytopes," *Proc. IEEE International Conference on Robotics and Automation*, 2002.

Ehmann, S. A., and M. C. Lin. "Accurate and Fast Proximity Queries between Polyhedra Using Surface Decomposition," *Computer Graphics Forum* (Proc. Eurographics 2001).

Hoff, K., A. Zaferakis, M. Lin, and D. Manocha. "Fast and Simple 2D Geometric Proximity Queries Using Graphics Hardware," *Proc. ACM Interactive 3D Graphics Conference*, 2001.

Ehmann, S. A., and M. C. Lin. "Accelerated Proximity Queries Between Convex Polyhedra by Multi-Level Marching," *Proc. International Conference on Intelligent Robots and Systems*, 2000.

Larsen, E., S. Gottschalk, M. Lin, and D. Manocha. "Fast Proximity Queries Using Swept Sphere Volumes," Department of Computer Science technical report TR89–018, University of North Carolina, 1999.

Hudson, T., M. Lin, J. Cohen, S. Gottschalk, and D. Manocha. "V-COLLIDE: Accelerated Collision Detection for VRML," *Proc. VRML Conference*, 1997, 119–125.

Gottschalk, S., M. Lin, and D. Manocha. "OBB-Tree: A Hierarchical Structure for Rapid Interference Detection," *Proc. ACM SIGGRAPH*, 1996, 171–180.

Key Words
Collision detection; distance computation; tolerance verification; dynamic simulation; hierarchical data structures.
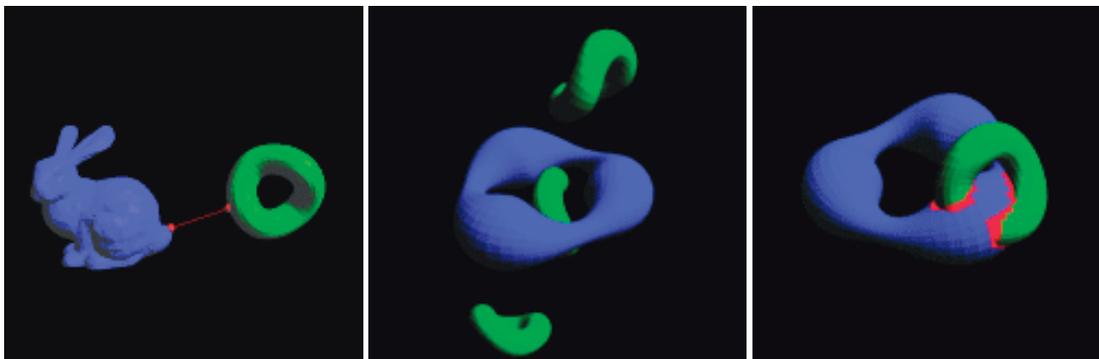
Left: A line connects the closest points of a bunny and a torus model. Middle: Three steps of a collision-free path. The path was computed by a path planner using tolerance verification. Right: The overlapping triangles of two intersecting tori are shown in red.

gamma.cs.unc.edu/collide/