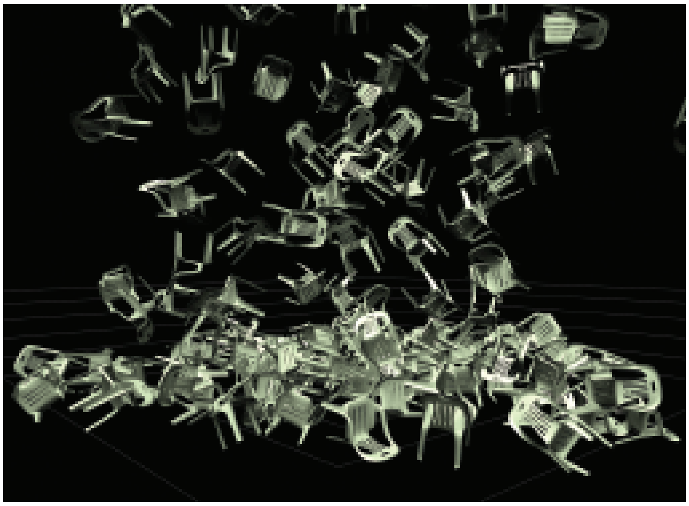
Output-Sensitive Collision Processing for Reduced-Coordinate Deformable Models

Experimental Animation, Length 1:52



Contributors
Doug L. James
Dinesh K. Pai
Christopher Twigg

Contact
Doug L. James
5000 Forbes Avenue
Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
USA
412-268-5505
djames@cs.cmu.edu

This animation shows offline animations of deformable collision phenomena created using a new collision-detection bounding volume hierarchy called a Bounded Deformation Tree, or BD-Tree (James & Pai, 2004). This algorithm can make collision detection for reduced-coordinate deformable models as asymptotically cheap as with rigid models. The bounding volumes of a BD-Tree can be updated in any order following deformation, and without need for explicit access to the deformed geometry. Consequently, collisions with very large and/or numerous models can be processed at very modest costs on the CPU, with deformations synthesized in programmable graphics hardware. In all of our examples, deformable collision, contact, and multibody dynamics processing are typically faster than rendering all models using programmable hardware (NVIDIA Quadro FX 3000). Our animation shows that large-scale physically based deformable simulations can be achieved at little more than the cost of rendering. The final chair sequence involves 3,601 chairs, with more than 60 million deforming triangles and 1.6 billion sphere-collision events, and has a mean collision processing cost of two seconds per time-step.

REFERENCE

James, D.L. & Pai, D.K. 2004. BD-tree: Output-sensitive collision detection for reduced deformable models. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2004).

PRODUCTION

Modeling: Finite element deformation. Rendering technique used most: Vertex programs for graphics hardware accelerated synthesis of physically based deformation (for example, DyRT) and lighting (for example, pre-computed radiance transfer). Average CPU time for rendering per frame: for scenes involving approximately 60 million triangles, hardware rendering took several seconds per frame. Total production time: final video frames and modeling, approximately one week (not including R&D). Production highlight: No fish were harmed in this production.

SOFTWARE

Modeling: Emacs. Animation: proprietary software. Rendering: OpenGL (GL4Java). Dynamics: proprietary, BD-Tree. Compositing: None. Additional software: Emacs, Java 2. Custom software: All of it. OS: Windows XP Pro.

HARDWARE. Dual Xeon 3 GHz CPU, 2 GB RAM. Graphics card: NVIDIA Quadro FX 3000. Hardware rendering was used for final renders.