

NVIDIA®

GPU Physics

Simon Green NVIDIA Developer Technology

What is Game Physics?



Enhances game experience by simulating the motion of objects based on the laws of physics

Simulates objects and interactions between them
 Rigid bodies, "rag dolls", particles, cloth, fluids, etc.
 Collisions, constraints, fluid forces, etc.

Current state of the art in game physics:
~1-2K colliding objects on current CPUs

Goal: Scalable Game Physics



Physics-based effects on a massive scale

- 10,000s of objects
- Rigid bodies
- Particles
- **Fluids**
- Cloth
 - and more

Physics effects should scale with capability of platform

Why Physics on GPUs?



Pixel quality has improved tremendously over the last few years

Material shaders, lighting and shadowing

Still much room for improvement in physics
 Small number of objects, limited interaction

Many games today are CPU limited

Makes sense to perform simulation close to rendering

Why Physics on GPUs?



GPU: very high data parallelism

- G70: 24 pixel pipelines, 48 shading processors
- 1000s of simultaneous threads
- Very high memory bandwidth
- SLI enables 1-4 GPUs per system

Physics: very high data parallelism

- 10,000s of objects
- 1000s of collisions to resolve every frame
- Requires 1000s of floating point operations per collision

NVIDIA GPU Pixel Shader GFLOPS Image: Constraint of the set of t



General-Purpose Computation on GPUs



GPUs have been used to accelerate many highly parallel applications

- physically-based simulation
- image processing
- scientific computing
- computer vision
- computational finance
- medical imaging
- bioinformatics



Physically-based Simulation on GPUs





Particle Systems –

Fluid Simulation



Jens Krüger, TU-Munich









Soft-body Simulation

Doug L. James, CMU

What About Rigid Body Physics?



Fluids, particles, cloth map naturally to GPUs

Highly parallel, independent data

Rigid body physics is more complicated

- Arbitrary shapes
- Arbitrary interactions and dependencies
- Parallelism is harder to extract

Physics Refresher Course





Bread Phase Collision Detection





Narrow Phase Collision Detection



Solve collisions









Anatomy of a clock tick









Anatomy of a clock tick









Anatomy of a clock tick





Slide courtesy of Andrew Bond, Havok





Slide courtesy of Andrew Bond, Havok

Physics Is A Data Parallel Task

Havok and NVIDIA

Havok is a world leader in physics middleware

- NVIDIA launched R&D project with Havok in 2005 to investigate physics on GPUs
 - Optimized for NVIDIA platforms

Havok FX

Havok FX is World's first GPU-accelerated game physics solution
Part of Havok 4 SDK

Enables a new class of large-scale physics effects in games

Already being adopted by game developers

Havok FX Features Overview

Havok FX Physics

Game Physics on Multiple GPUs

SLI - READY- Second GPU can be used for SLI graphics,

NVIDIA GPU Physics

Multi-GPU configurations, mixed or same GPU type

- One GPU does both graphics and physics
- One GPU for graphics, one for physics
 - Enables extra GPU for rendering when FX is not active
- Two GPUs for graphics, one for physics
 - Full speed rendering with full speed physics simulation

GeForce 7600 GS

GeForce 7950 GX2 + GeForce 7600

GeForce 7900 GTX SLI

Custom Behaviors

Gameplay physics interaction

NVIDIA Technology for Physics

SLI multi-GPU technology

New driver technology for physics

SLI Performance Scaling

Frame

Rate

Performance Measurement 15,000 Boulders with Shadows

Single GPU

Dual Core P4EE 955 - 3.46GHz GeForce 7900GTX CPU Multi-threading enabled

Dual GPU

Dual Core P4EE 955 - 3.46GHz 2 x GeForce 7900GTX SLI CPU Multi-threading enabled

Rendering is fully controlled by application

Havok FX returns vertex buffers with position, velocity and optional user data

Supports OpenGL and Direct3D

Rigid bodies rendered using instancing
Direct3D or OpenGL NVX_instanced_arrays extension

Particles rendered as point sprites

- Supports motion blur
- Can modify particle color or size over time
- Can use texture atlases for particle animation

The Future of GPU Physics

Distributing physics across multiple GPUs

 e.g. 2 GPUs for physics, 2 for rendering

 Brittle fracture
 Advanced smoke/cloud rendering

 Volumetric shadowing

 Advanced fluids

 Smoothed particle hydrodynamics
 Isosurface extraction using DirectX 10 Geometry Shader

Questions?

