

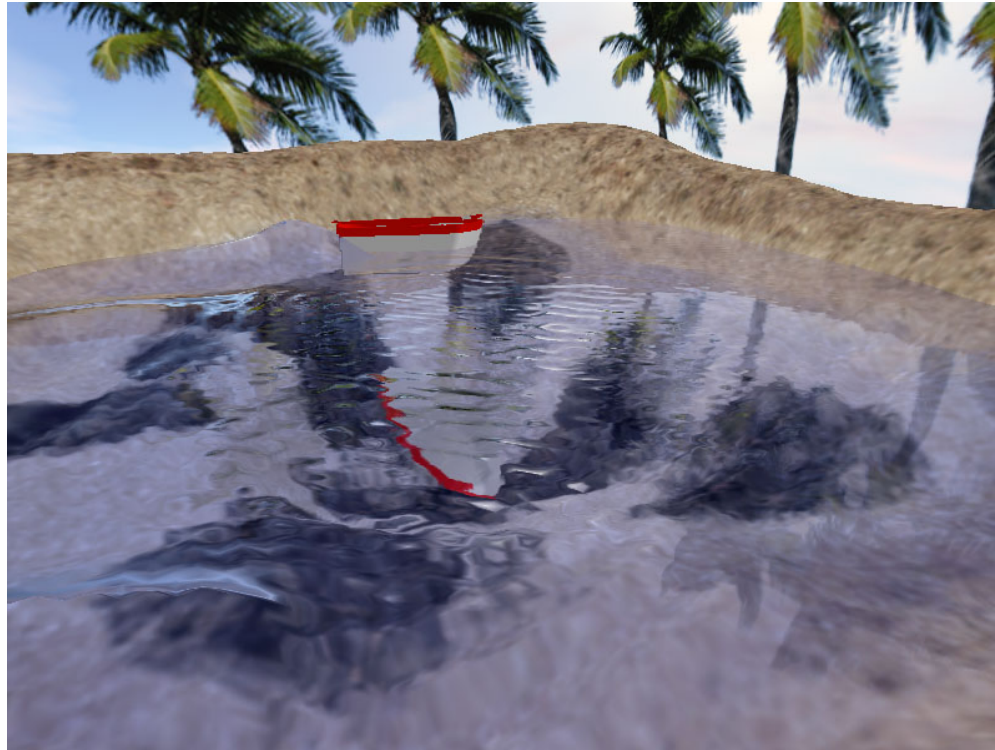


NVIDIA®

GPU Water Simulation

Jeremy Zelsnack

Demo



NVIDIA.

Algorithm Overview

- **Perform water simulation in pixel shader**
 - **Render to texture (D3DFMT_A16B16G16R16F)**
- **Render refraction and reflection maps**
- **Render water surface**
 - **Use simulation results via VS3.0 vertex texture fetch**
 - **Compute perturbed texture coordinates**
 - **Combine refraction and reflection using Fresnel term**



Simulation

- **Solve 2D wave equation**

- **Verlet Integration**

- **Good stability**
 - **Uses previous 2 results**
 - **No velocity stored**

$$\frac{\partial^2 y}{\partial t^2} = c^2 \left(\frac{\partial^2 y}{\partial x^2} + \frac{\partial^2 y}{\partial z^2} \right)$$

$$\left(\frac{\partial^2 y}{\partial x^2} + \frac{\partial^2 y}{\partial z^2} \right) \approx \left[\sum_{4\text{-neighbors}} y_{i,j} \right] - 4 * y_{0,0}$$

- **Influence simulation**

- **Dampening maps**

- **Control simulation**

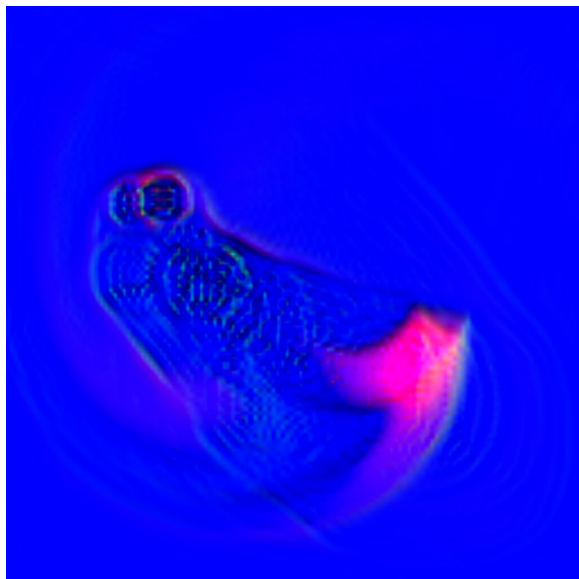
- **Render alpha-blended geometry into simulation**



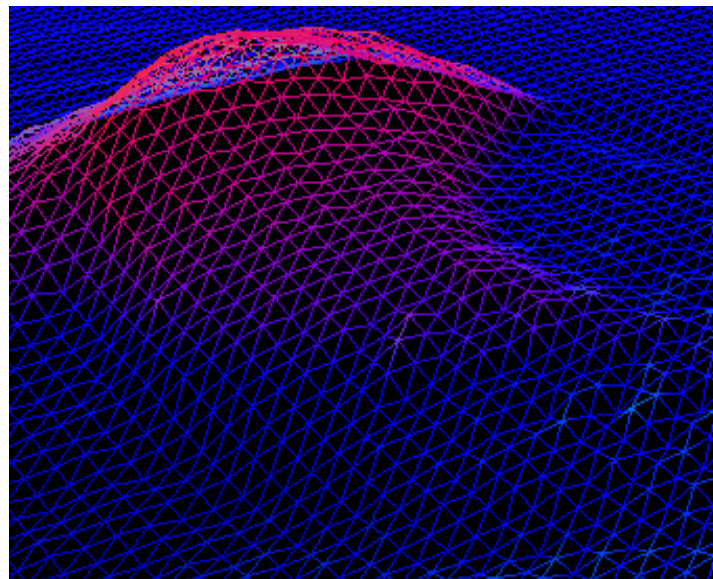
Vertex Texture Fetch (VS3.0)

- Vertex shader reads simulation result with vertex texture fetch

Simulation Texture



Height Map



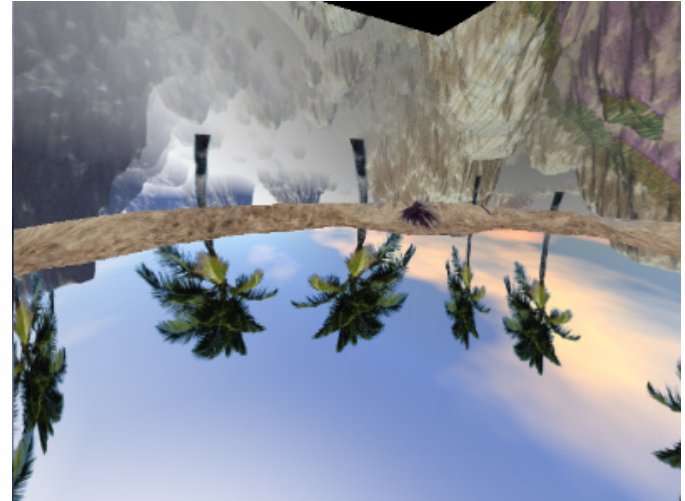
Refraction Map

- Render scene from camera viewpoint
- If camera is above water
 - Render underwater geometry
- If camera is underwater
 - Render above water geometry



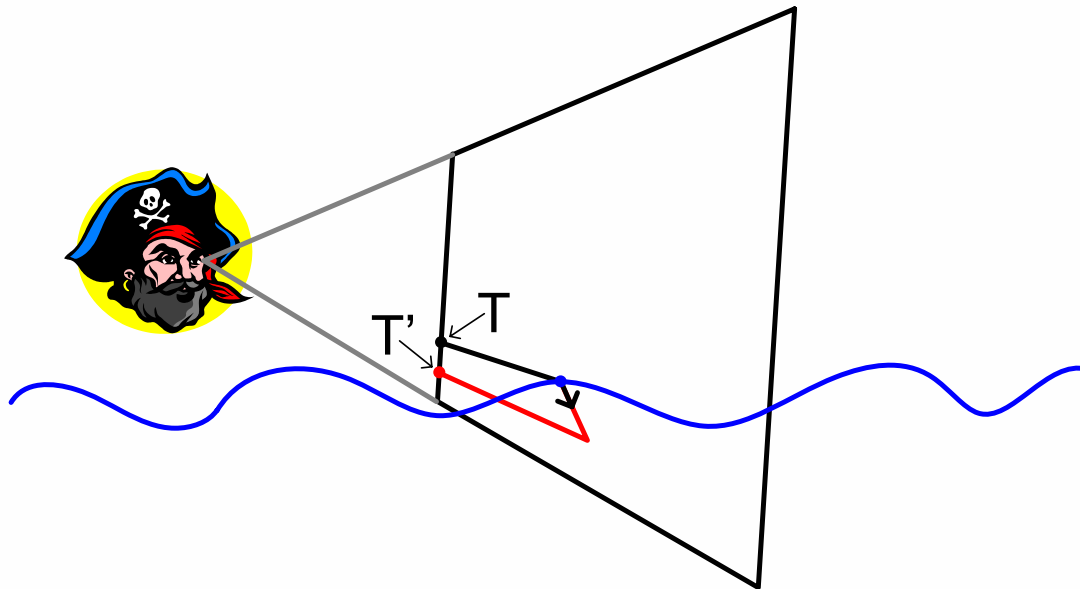
Reflection Map

- Render scene from reflected camera viewpoint
 - Reflect view transform about water plane
- If camera is underwater
 - Render underwater geometry
- If camera is above water
 - Render above water geometry



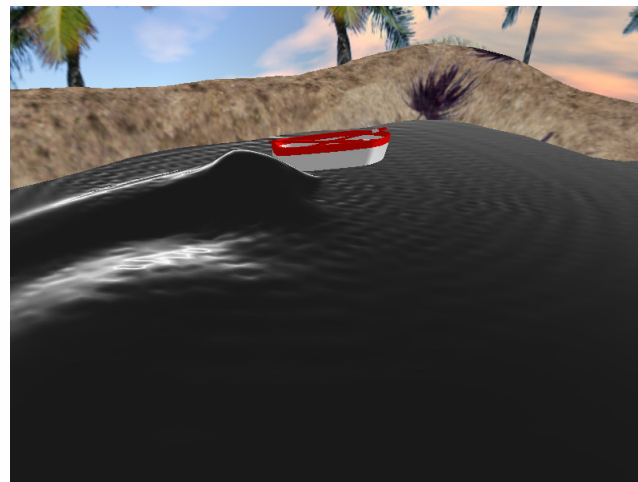
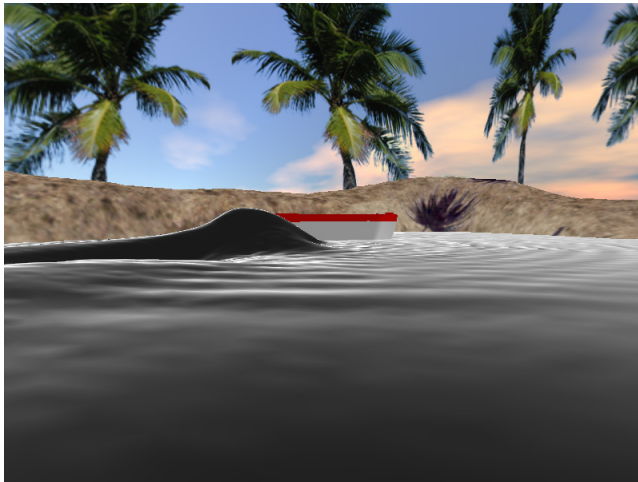
Perturbed Texture Coordinates

- Start at water position
- Move along refraction or reflection vector
- Project into screen space

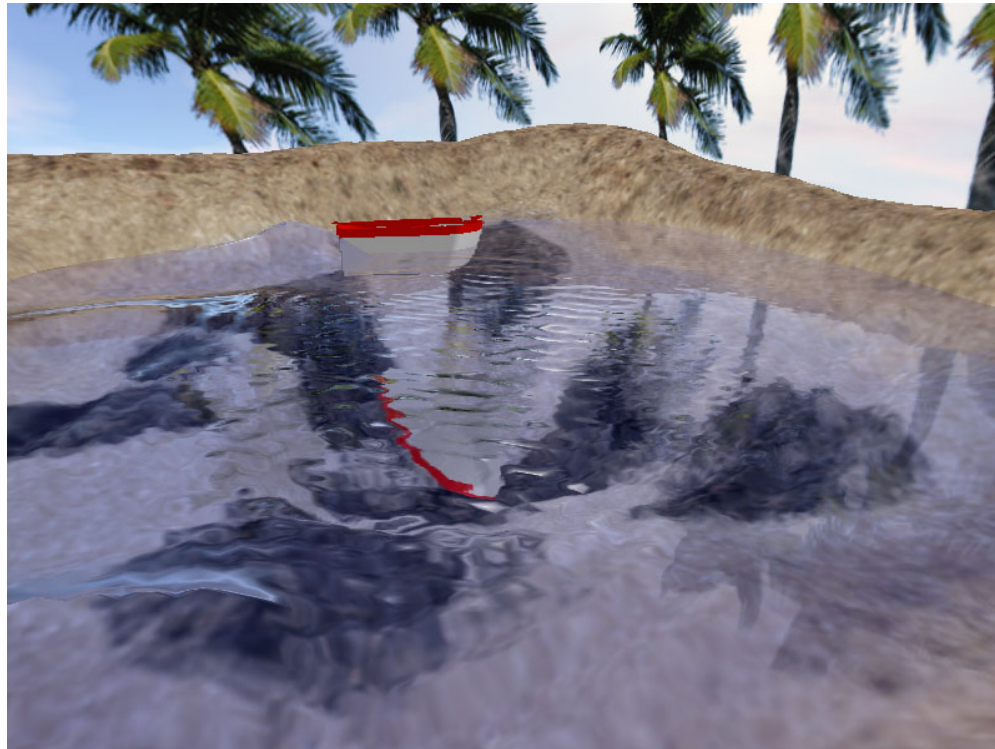


Fresnel Reflection Term

- Determines amount of reflection / refraction
- Roughly $\text{pow}((1 - \text{dot}(\text{eye}, \text{normal})), p)$
 - Fresnel term = 0 \Rightarrow all refraction
 - Fresnel term = 1 \Rightarrow all reflection



Demo



NVIDIA.

Questions or Comments?

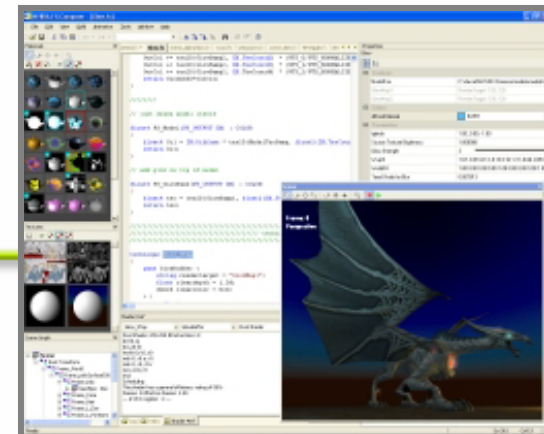
- jzelsnack@nvidia.com
- <http://developer.nvidia.com>



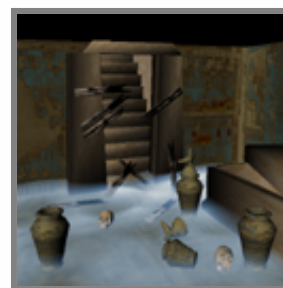
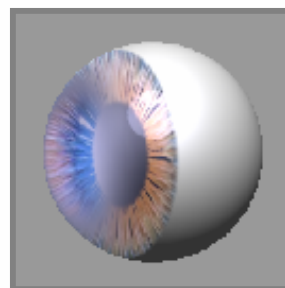
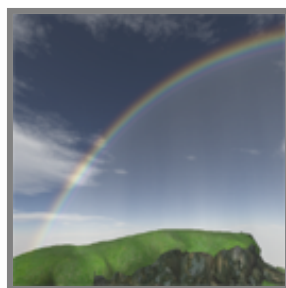
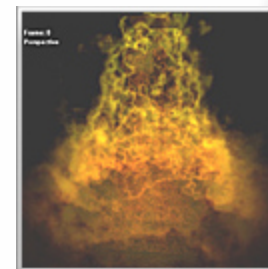
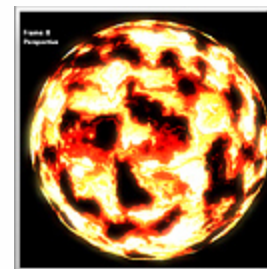
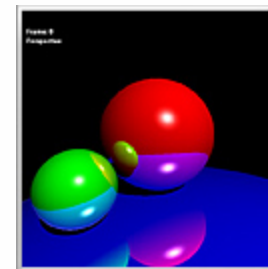
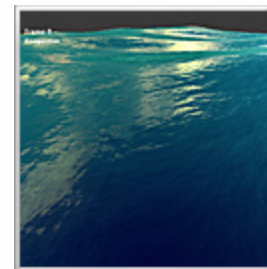
developer.nvidia.com

The Source for GPU Programming

- Latest documentation
- SDKs
- Cutting-edge tools
 - Performance analysis tools
 - Content creation tools
- Hundreds of effects
- Video presentations and tutorials
- Libraries and utilities
- News and newsletter archives



EverQuest® content courtesy Sony Online Entertainment Inc.



NVIDIA.

GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Graphics

- Practical real-time graphics techniques from experts at leading corporations and universities
- Great value:
 - Contributions from industry experts
 - Full color (300+ diagrams and screenshots)
 - Hard cover
 - 816 pages
 - Available at GDC 2004

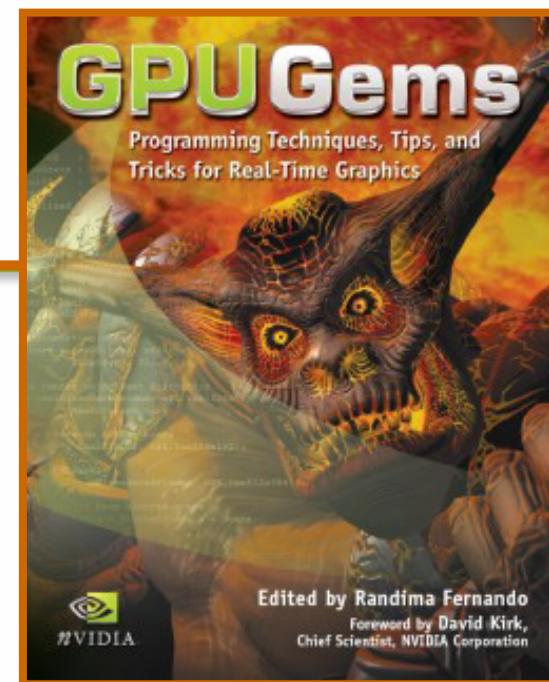
For more, visit:

<http://developer.nvidia.com/GPUGems>

“*GPU Gems* is a cool toolbox of advanced graphics techniques. Novice programmers and graphics gurus alike will find the gems practical, intriguing, and useful.”

Tim Sweeney

Lead programmer of *Unreal* at Epic Games



“This collection of articles is particularly impressive for its depth and breadth. The book includes product-oriented case studies, previously unpublished state-of-the-art research, comprehensive tutorials, and extensive code samples and demos throughout.”

Eric Haines

Author of *Real-Time Rendering*