How to Render a Real Rainbow

Clint Brewer

NVIDIA Developer Technology Group

Overview

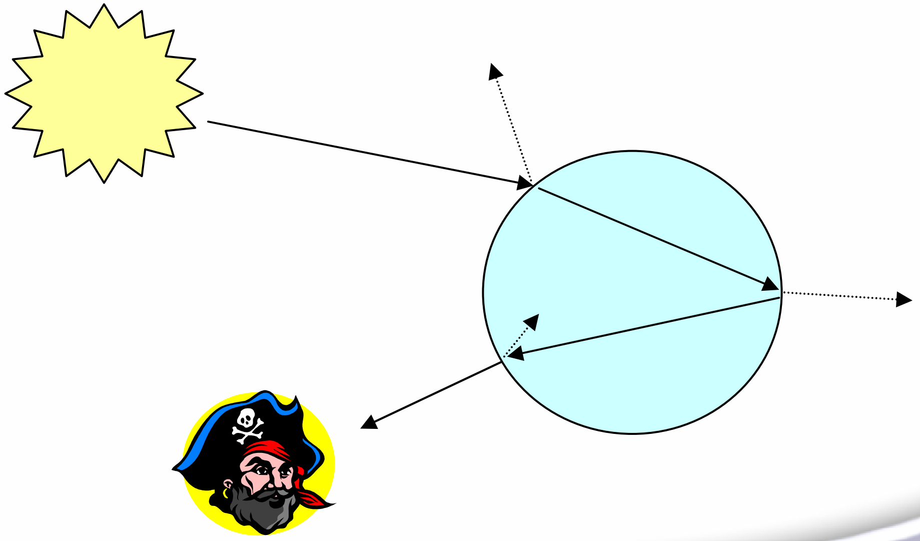
- Demo
- What causes a rainbow
- Different ways to render the rainbow
- How to combine it with the rest of the scene
- Demo
- Questions

Demo

What Causes a Rainbow

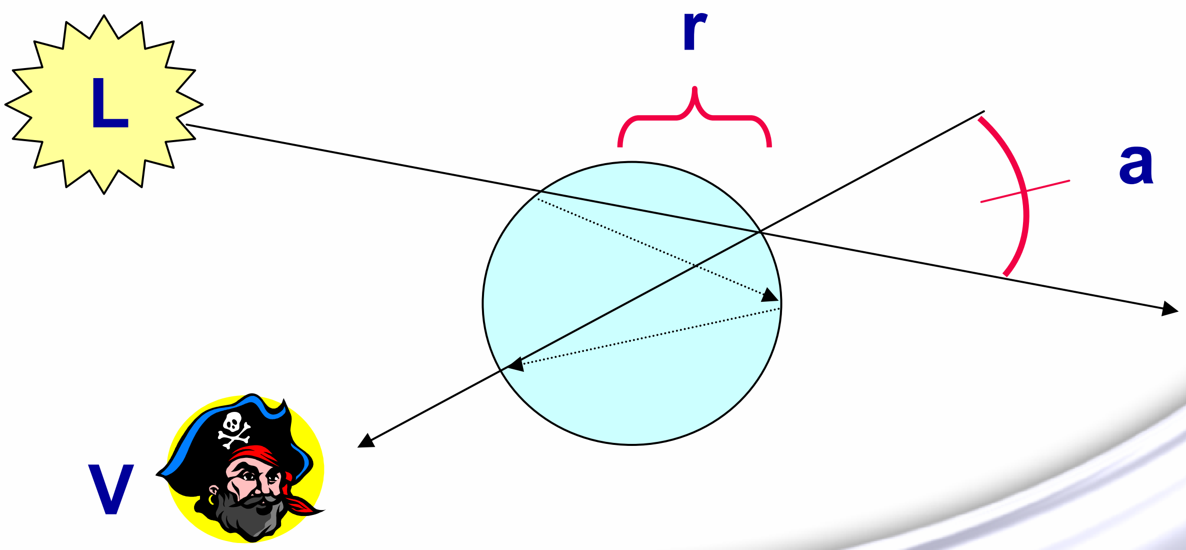
- Scattering of sunlight

- Refraction and reflection by spherical water drops



What Causes a Rainbow

- Color of a rainbow at your eye depends on
  - Angle of deviation (a)
    - Between view (V) and Sun light (L)
  - Radius of water droplet (r)

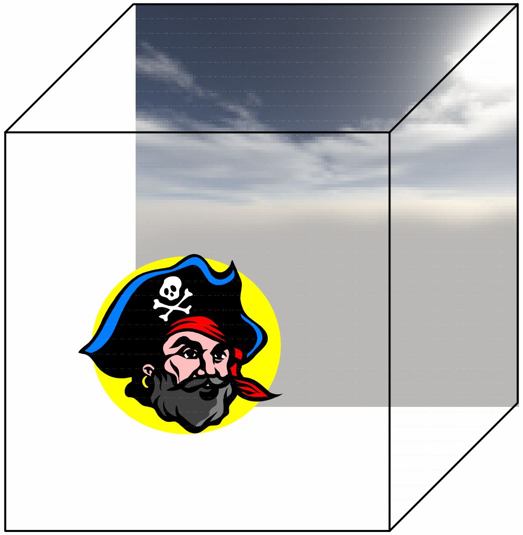


How Can We Render This?

- Pre-rendered skybox
- Screen-aligned quad + pixel shader
    - Sky dome + texture slices

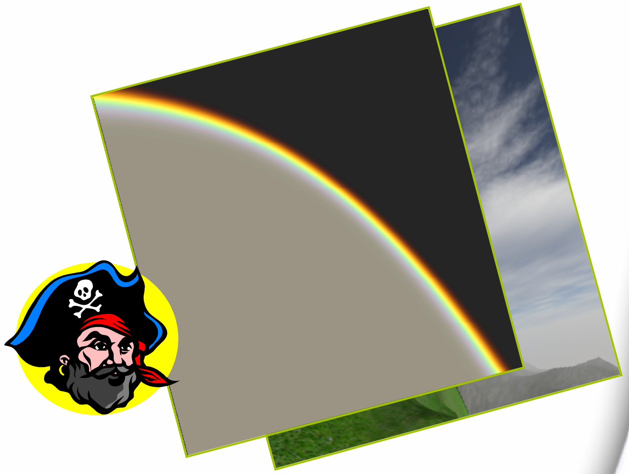
Pre-Rendered Rainbow Skybox

- Just bake the rainbow into your skybox
- The good
  - Very cheap!
- The not so good
  - Static
  - Requires extra texture storage



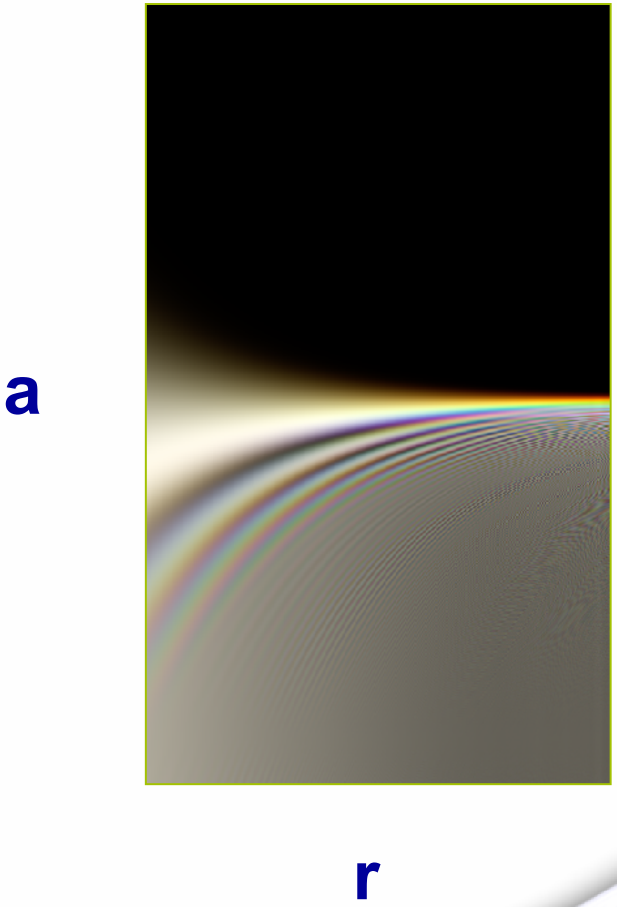
How About a Screen-Aligned Quad

- Screen-aligned quad + pixel shader
  - The good
    - Complete control over pixel color
    - Dynamic time of day
    - No extra texture storage
  - The not so good
    - Added computation at every pixel



Color on the Quad

- Lee Diagram
  - Angle of deviation
  - Droplet radius
- MiePlot (Philip Laven)
  - Can generate Lee Diagrams
  - lets you change the scattering model
    - Mie, Airy, Descartes

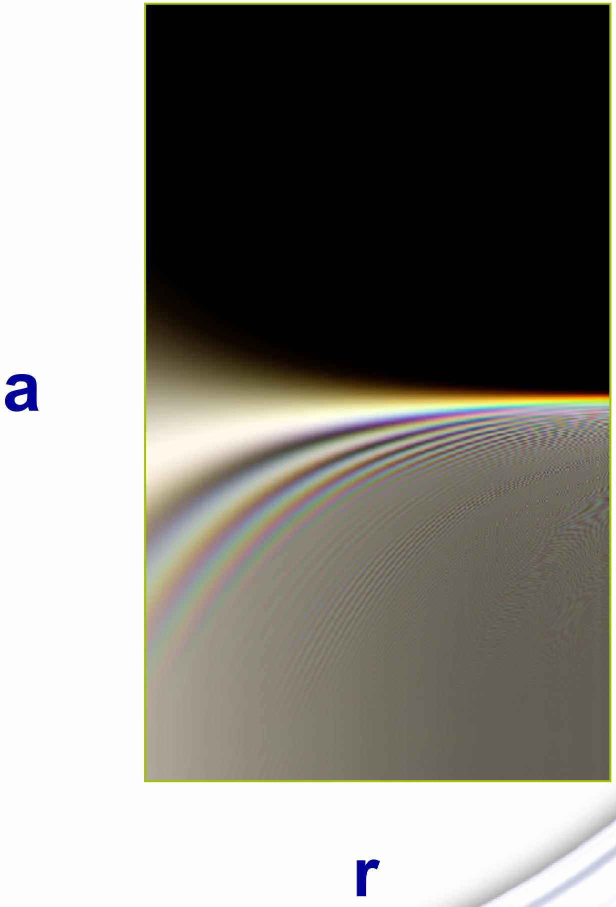


How to Use the Lee Diagram

- Math
  - Calculate **a** at each pixel on the screen in a pixel shader
  - Then actually compute the color with scattering math (Mie, Airy, etc)
- Or use a dependent texture read

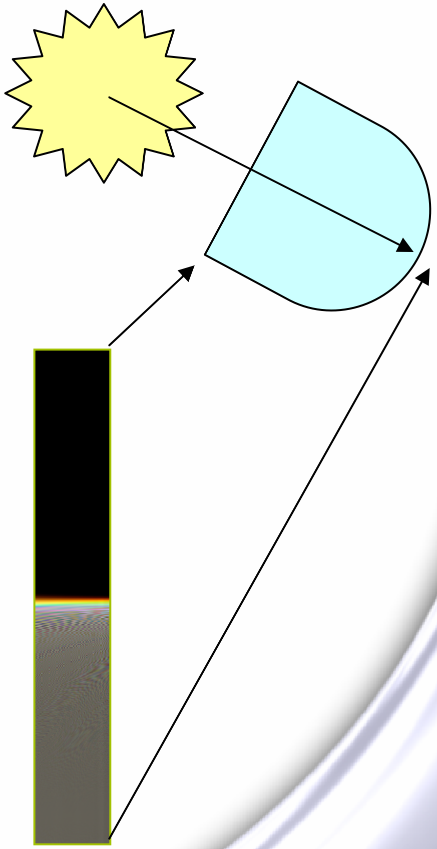
$$\mathbf{a} = 1 - \text{Light} \bullet \text{View}$$

$$\text{color} = \text{tex2D}(r, \mathbf{a})$$



An Optimization

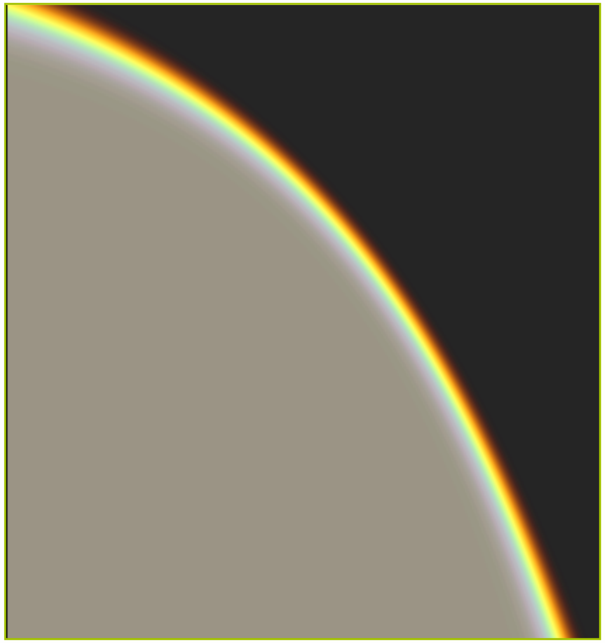
- Sun aligned sky dome
  - Map 1d slices of a Lee Diagram
  - The good
    - Cheap on the pixels
    - Supports changing sun position
    - No extra texture storage
  - The not so good
    - Added vertex processing
      - (Not that big a problem really)



A rendered rainbow is ...

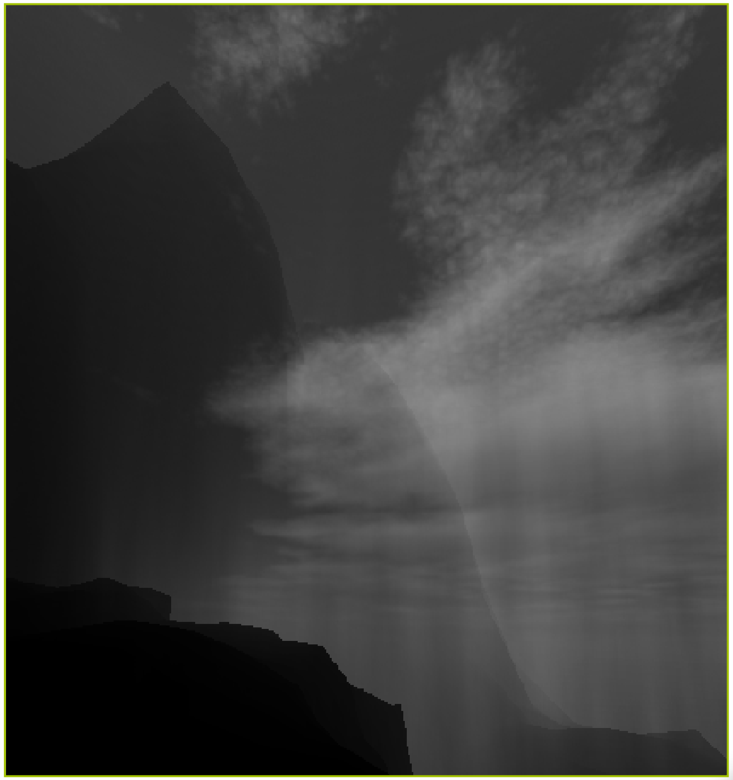
- Only as nice as your scene

- Things that can help
  - Fog
  - Atmospheric scattering
  - Rain
  - Clouds
  - Skybox

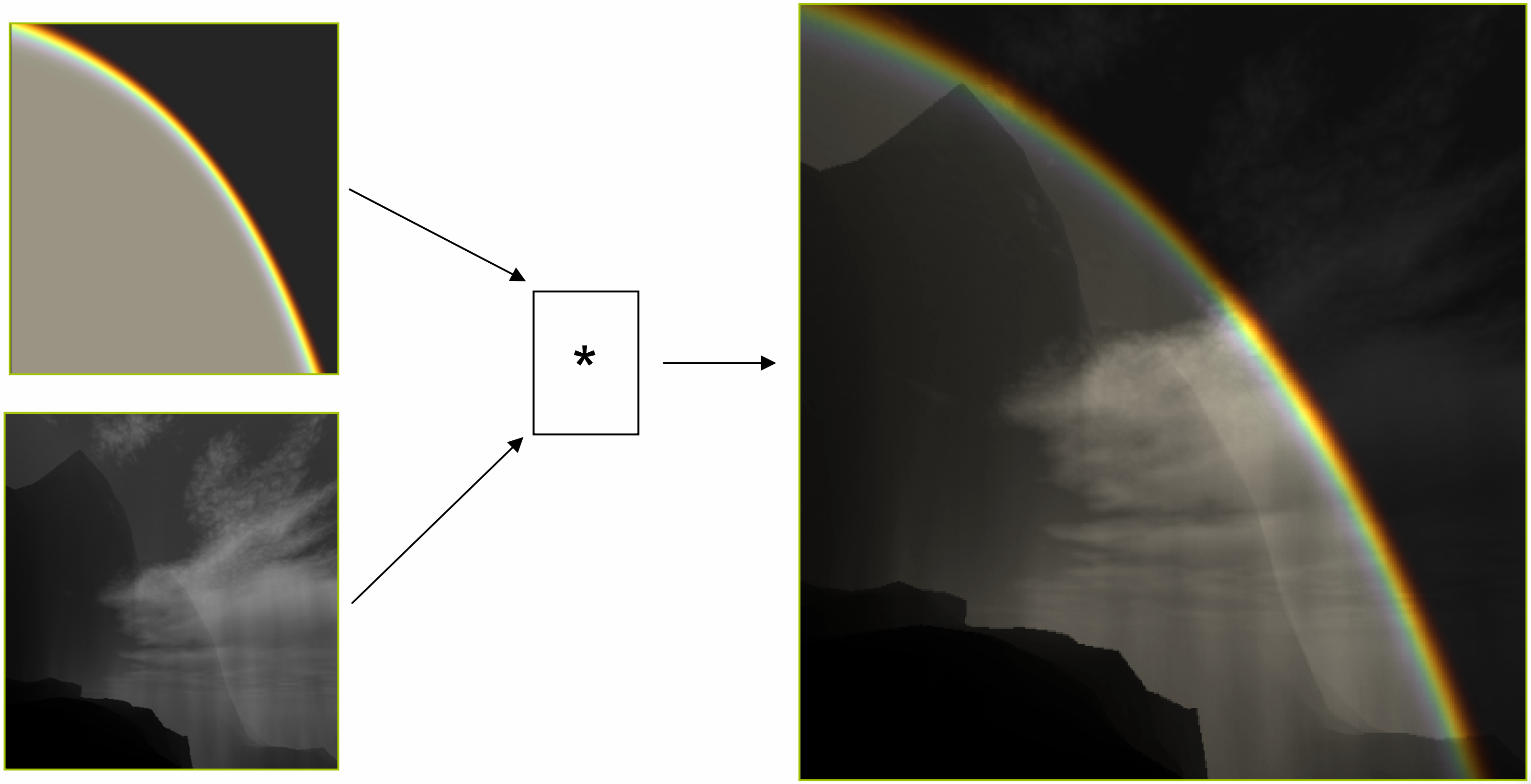


Moisture Texture

- Render moisture amount to a texture:
  - Fog
  - Rain
  - Clouds
  - Etc.

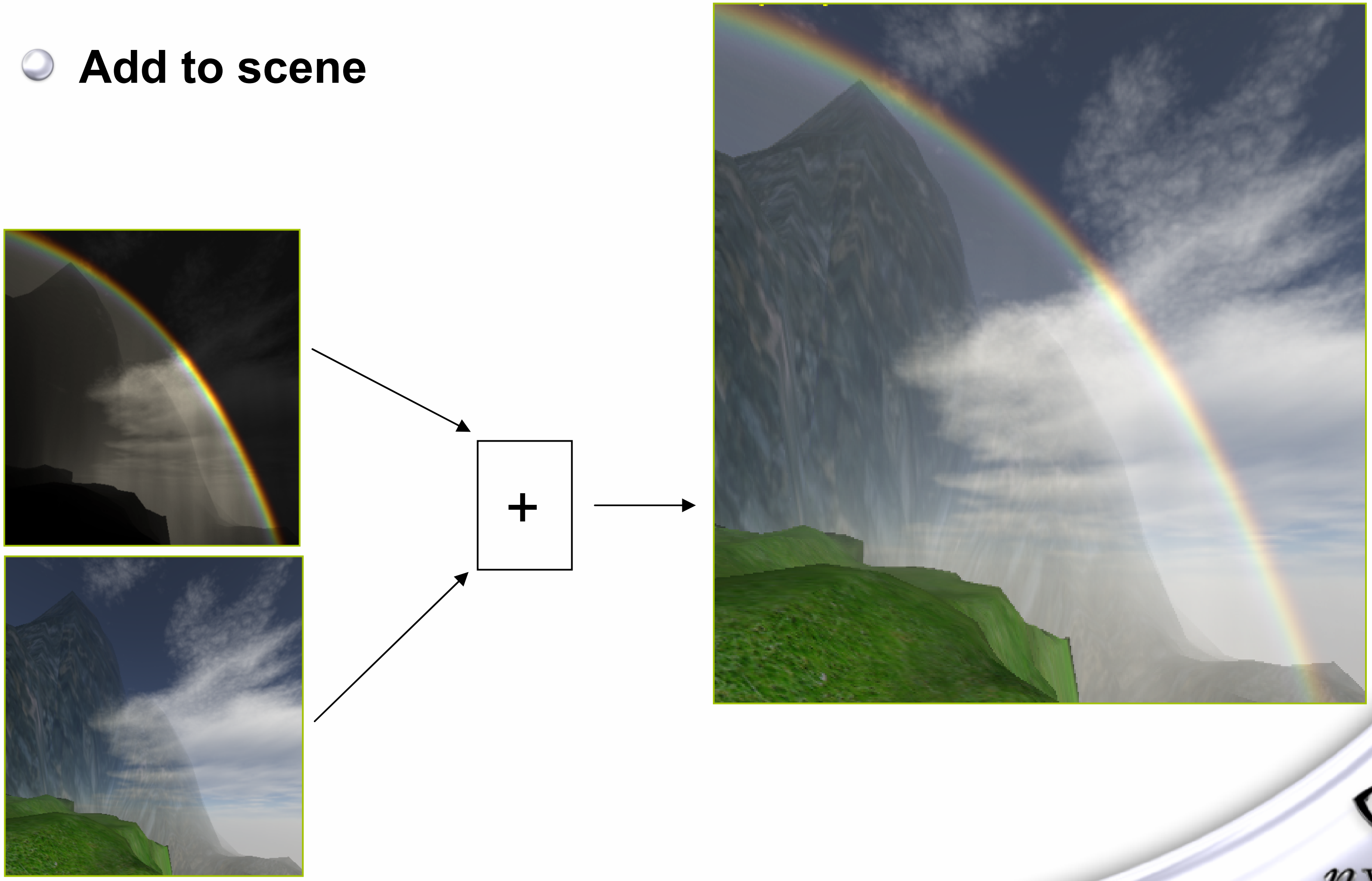


More moisture means more rainbow



Combining it with the Scene

- **Add to scene**



Demo

Questions?

- cbrewer@nvidia.com
- http://developer.nvidia.com

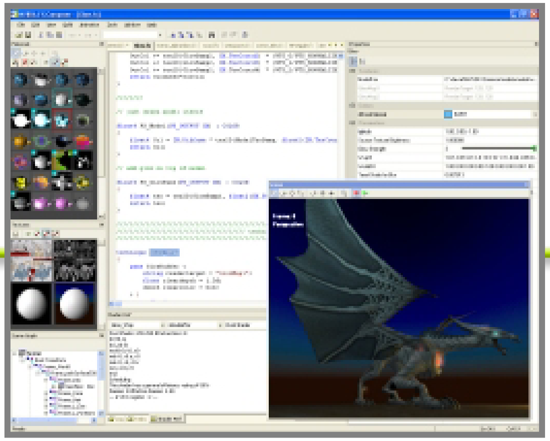
- slides are going online daily...
  - http://developer.nvidia.com/object/gdc_2004_presentations.html

References

- MiePlot, http://www.philiplaven.com/
- Atmospheric Optics, http://www.sundog.clara.co.uk/resource/intro.htm
- Lee, Raymond L, *"Mie Theory Airy Theory and the Natural Rainbow"*, Applied Optics, Vol. 37, No. 9, 1998

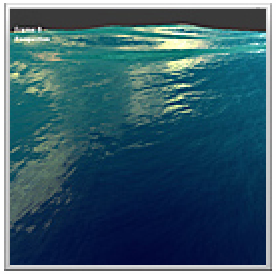
developer.nvidia.com

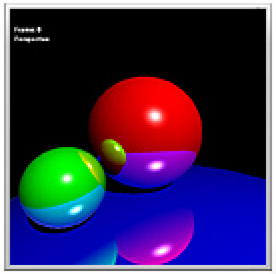
The Source for GPU Programming

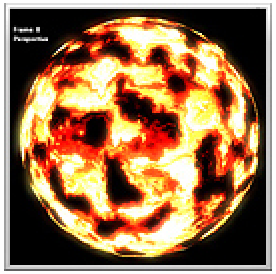


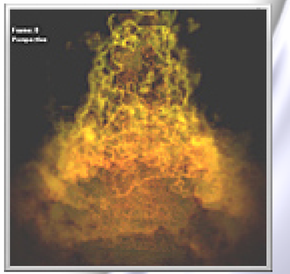
EverQuest® content courtesy Sony Online Entertainment Inc.

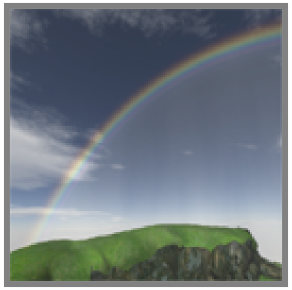
- Latest documentation
- SDKs
- Cutting-edge tools
  - Performance analysis tools
  - Content creation tools
- Hundreds of effects
- Video presentations and tutorials
- Libraries and utilities
- News and newsletter archives

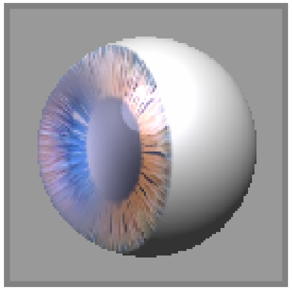




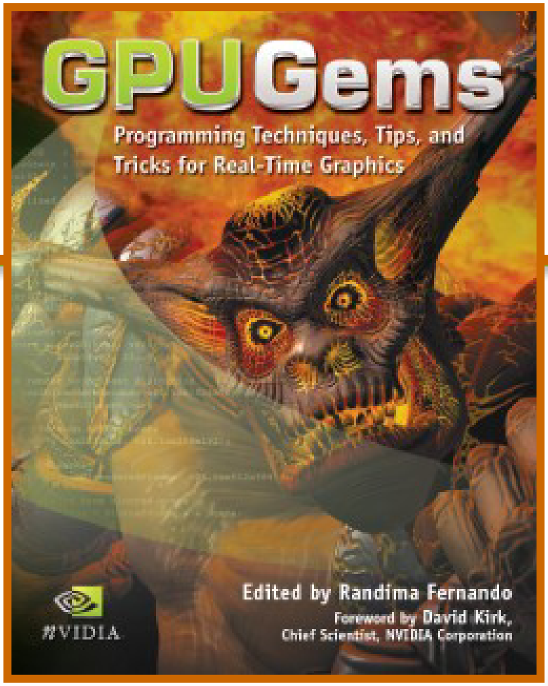








GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Graphics


GPUGems
Programming Techniques, Tips, and Tricks for Real-Time Graphics
Edited by Randima Fernando
Foreword by David Kirk, Chief Scientist, NVIDIA Corporation
NVIDIA

- 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*