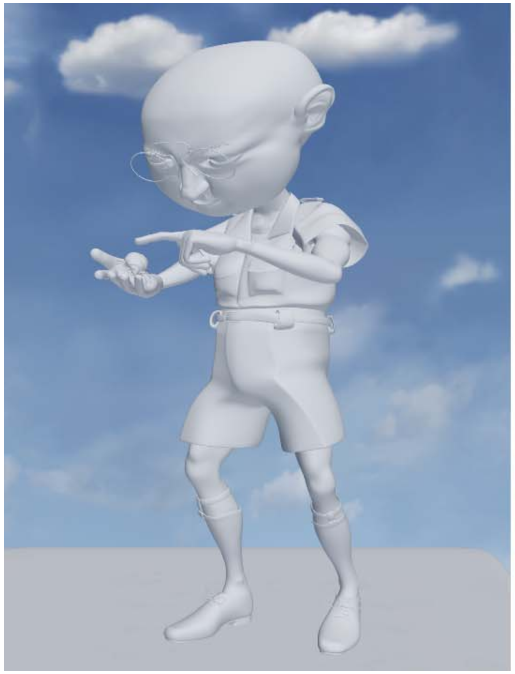
Dynamic Ambient Occlusion and Indirect Lighting

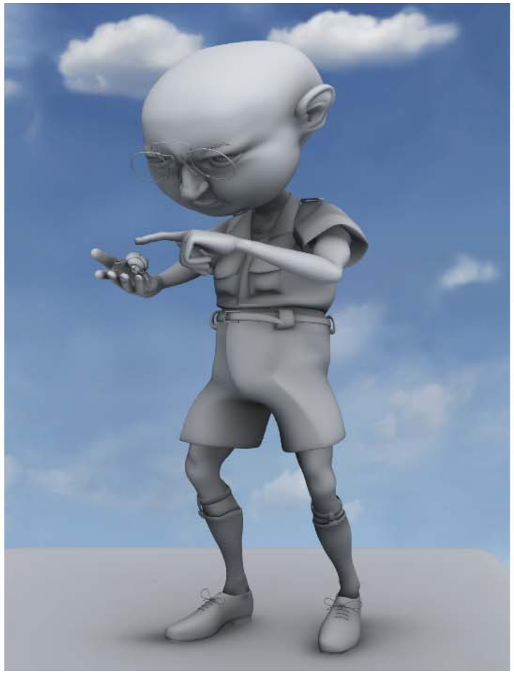
Michael Bunnell

NVIDIA Corporation

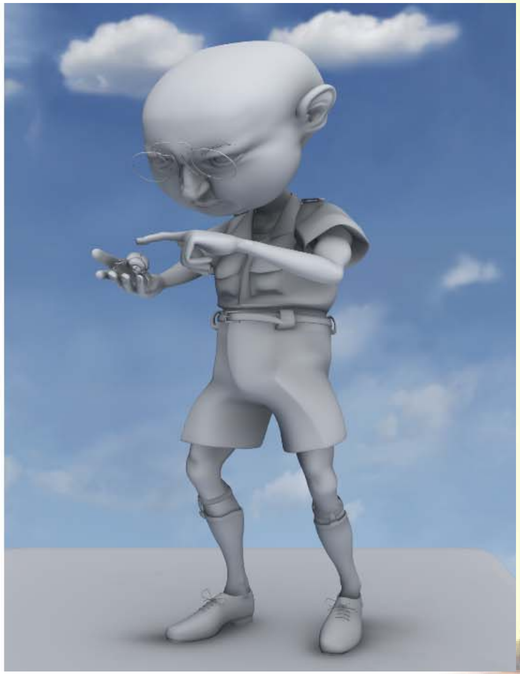
Environment Lighting



Environment Map



+ Ambient Occlusion

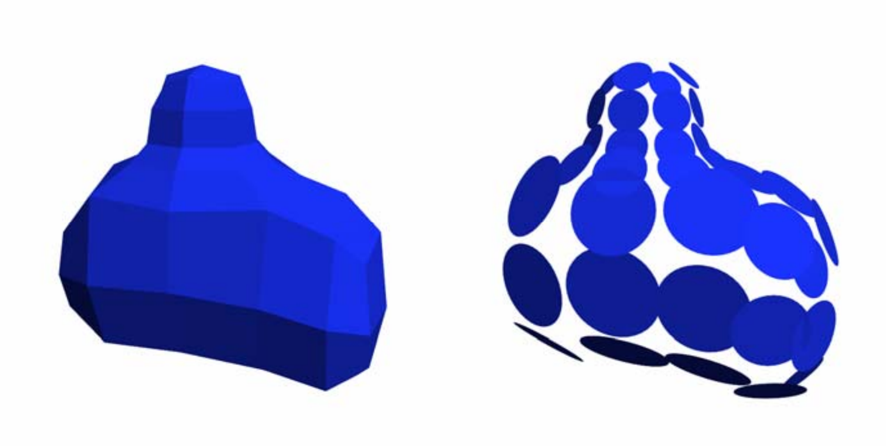


+ Indirect Lighting

New Radiance Transfer Algorithm

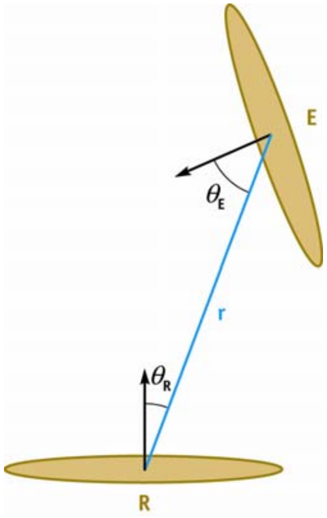
- Useful for calculating Ambient Occlusion and Indirect Lighting
- Efficient and parallelizable
- Implementation is real-time on GPU
- Ideal for non-rigid bodies and dynamic environments

Dynamic Ambient Occlusion



- **Define polygon meshes as disk-shaped elements**
  - **one element created for each vertex**
  - **elements defined by position, normal, and area**
  - **simplifies form factor calculation**

Form Factor



Emitter element E occludes receiver element R based on distance r and angles θ_E and θ_R

- **Percentage of the hemisphere above a point occluded by an element (Solid Angle)**
- **Like radiosity form factor with 100% visibility**

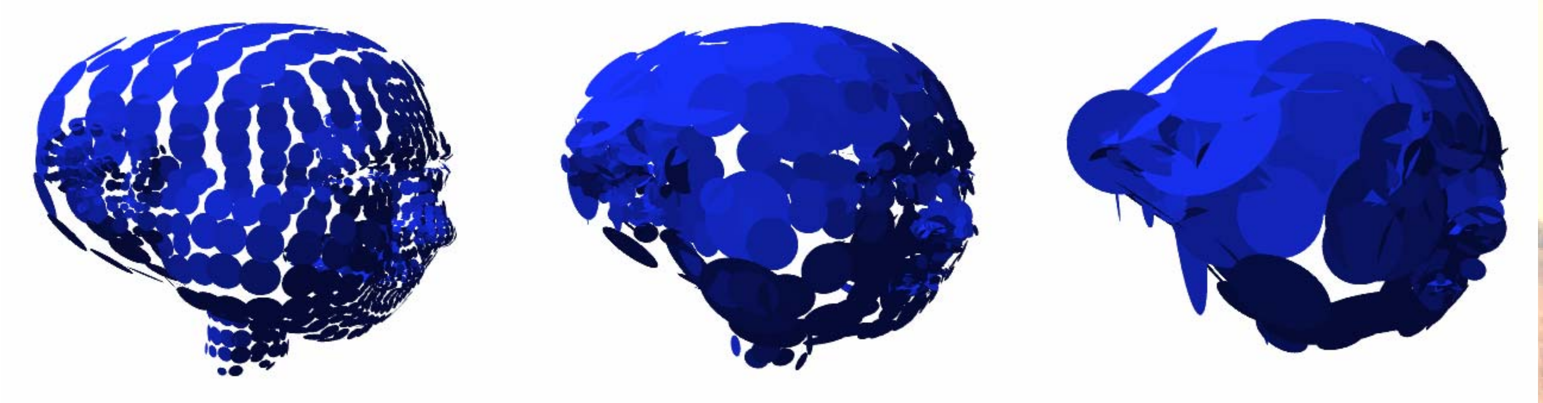
Calculating Occlusion

- Calculate occlusion at a receiver element by summing form factors:

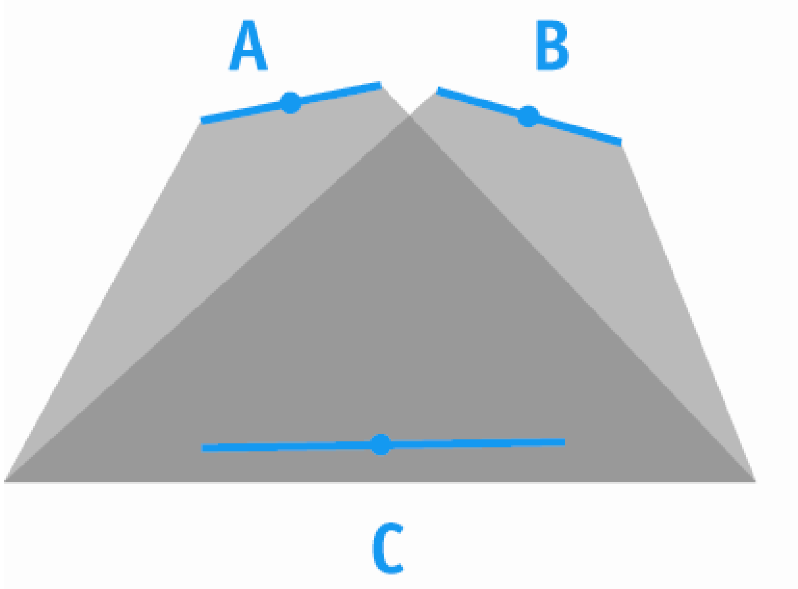
```
occlusion = 0;
for each element E
    occlusion += form factor of E;
```

Element Hierarchy

- We do not need to consider so many elements to get an accurate answer
  - A detailed head or simple ball will shadow distant objects the same
- Group elements together, forming larger elements
- Only traverse children when close to parent
- Easy to generate automatically since we don’t need actual geometry

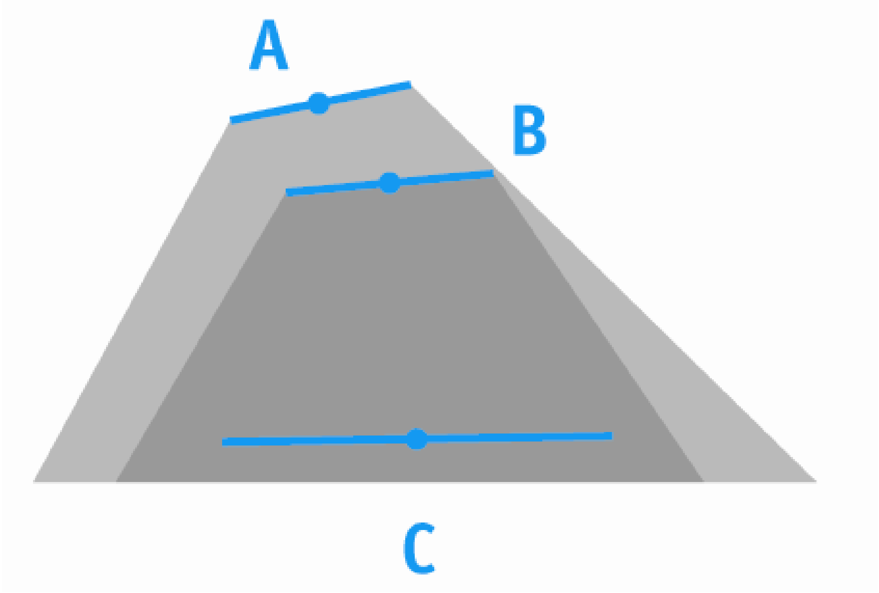


Double Shadowing



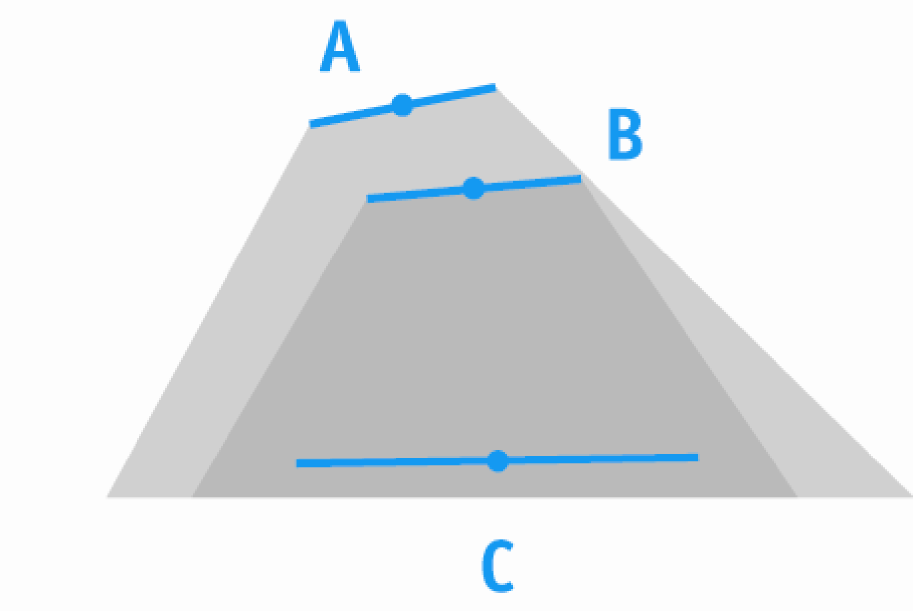
- A and B both shadow C
- C shadowed properly
- No double shadowing

Double Shadowing



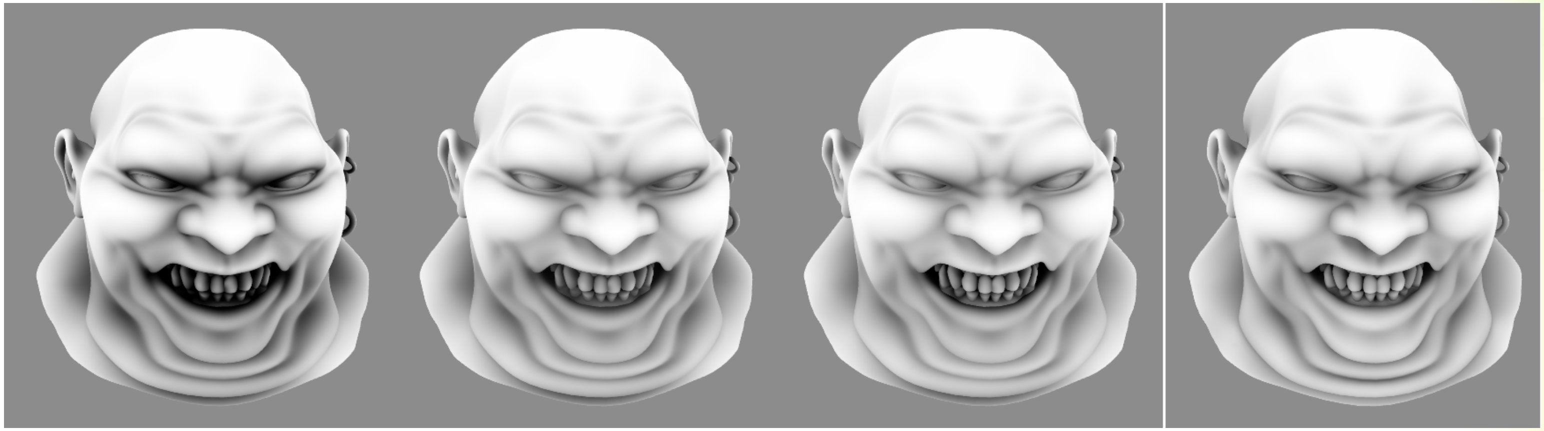
- A and B both shadow C
- C is shadowed too much
- Double shadowing after first pass

Double Shadowing



- **Lighten B's shadow in second pass since it is shadowed**
- **Double shadowing eliminated**

Eliminating Double Shadowing



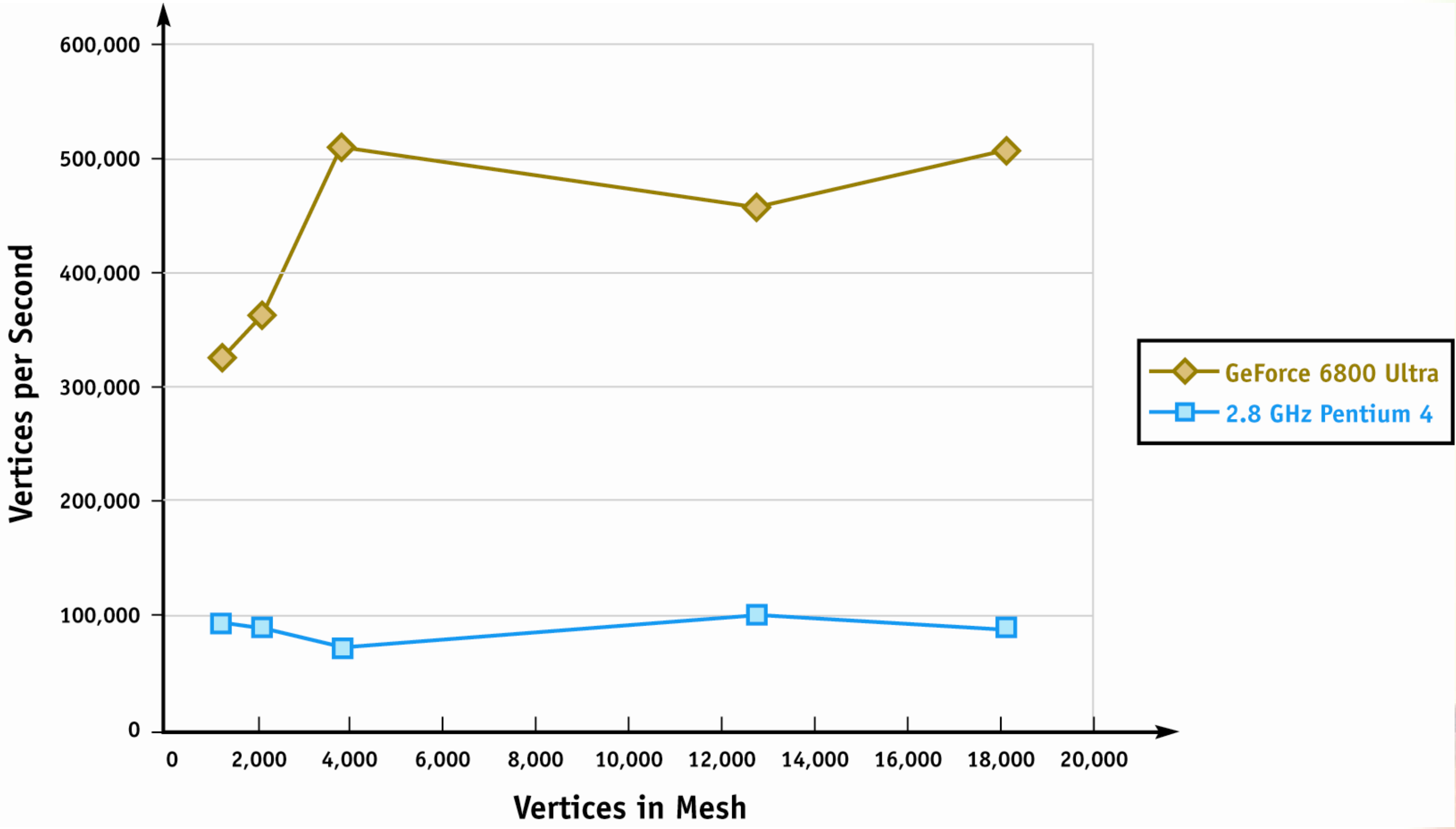
1 pass 2 passes 3 passes Ray traced

- **Multiply form factor by 1 - occlusion calculated in the previous pass**
- **Converges to correct shadowing quickly (2 passes are often enough)**
- **Results compare favorably with ray tracing**

GPU Implementation

- **Element data and index coordinates stored in a texture maps**
  - **Position, normal and area* are dynamic**
  - **index coordinates are pre-computed**
- **Shader traverses elements in a loop using** *next* **or** *child* **index coordinates**
- **Render a single quad (2 triangles) to complete a pass, 1 pixel per element**
- **Results are rendered to a texture for use in subsequent passes**

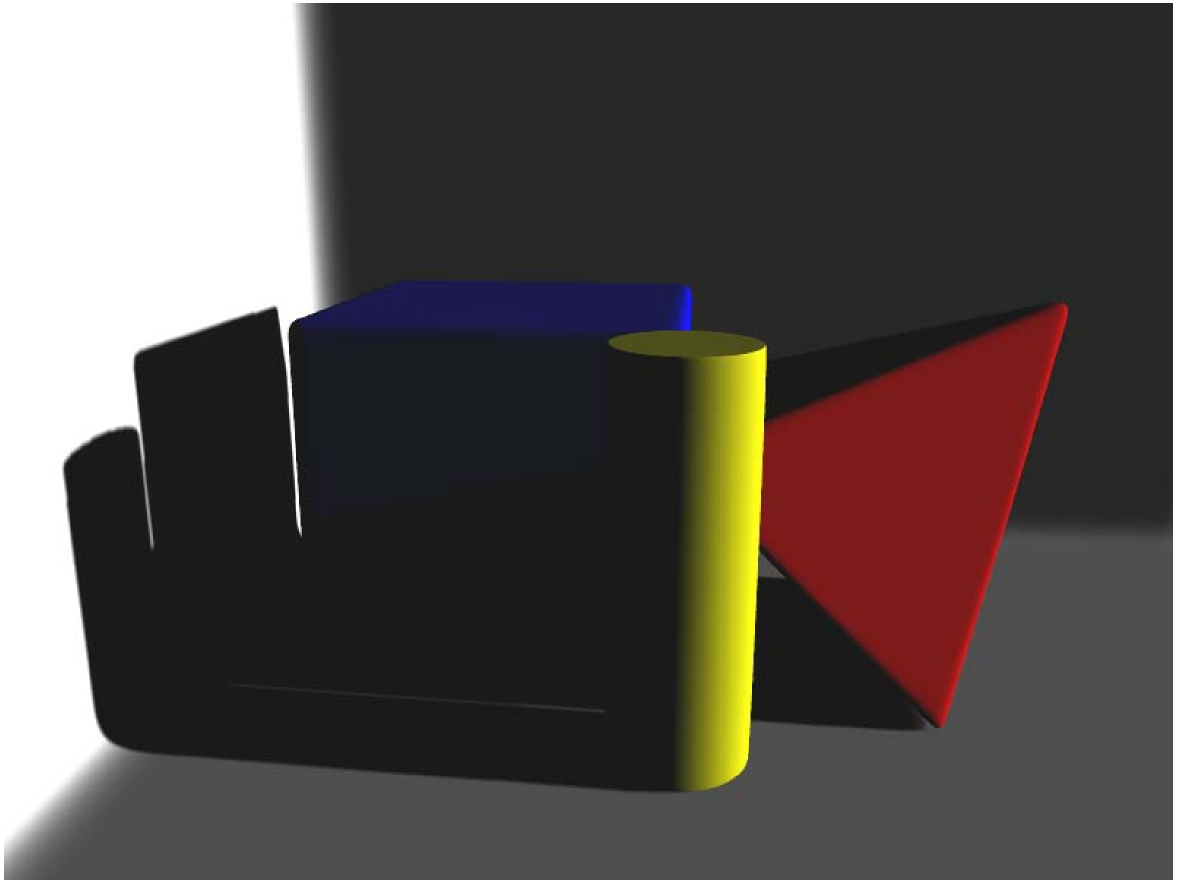
Performance



Indirect Lighting

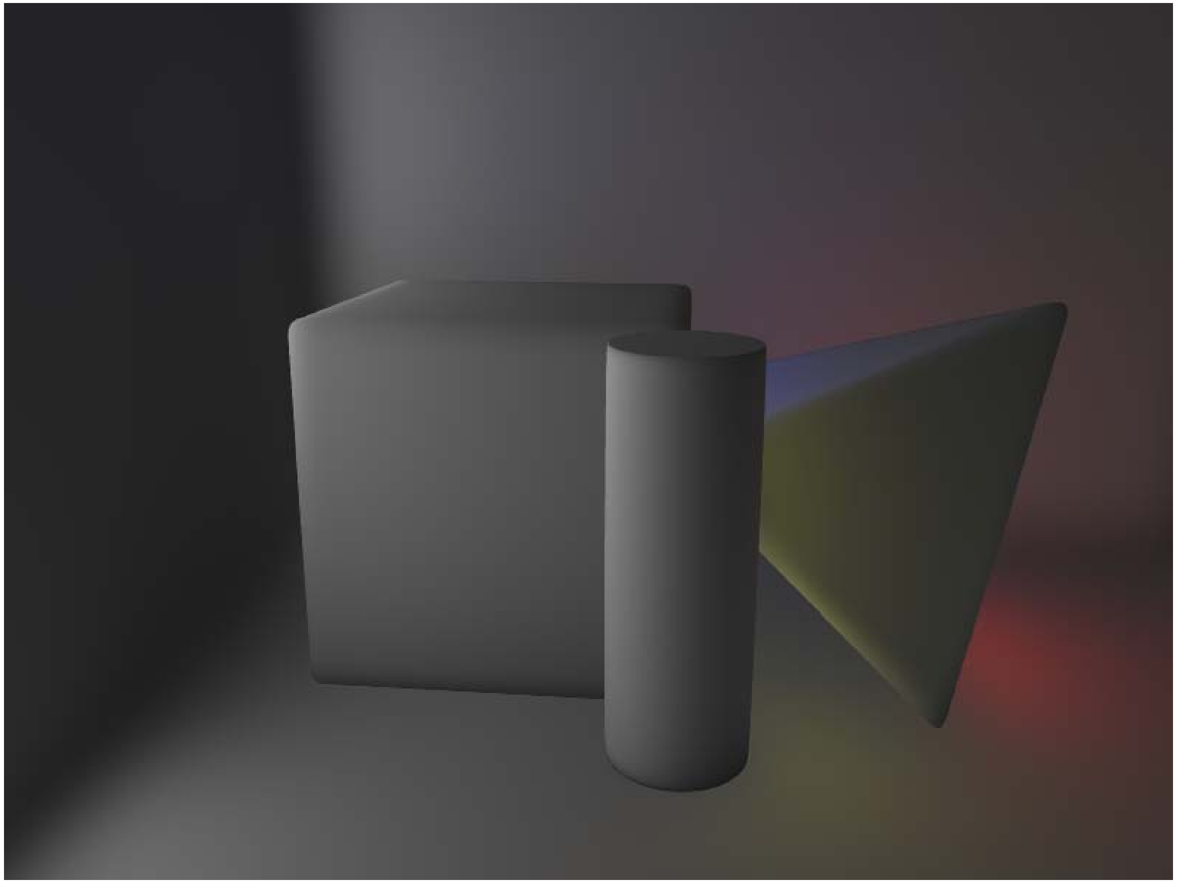
- Light reflecting off diffuse surfaces
- Used effectively in Shrek 2
- Adds an extra level of realism
- Can be used with traditional and environment lighting

Direct Lighting



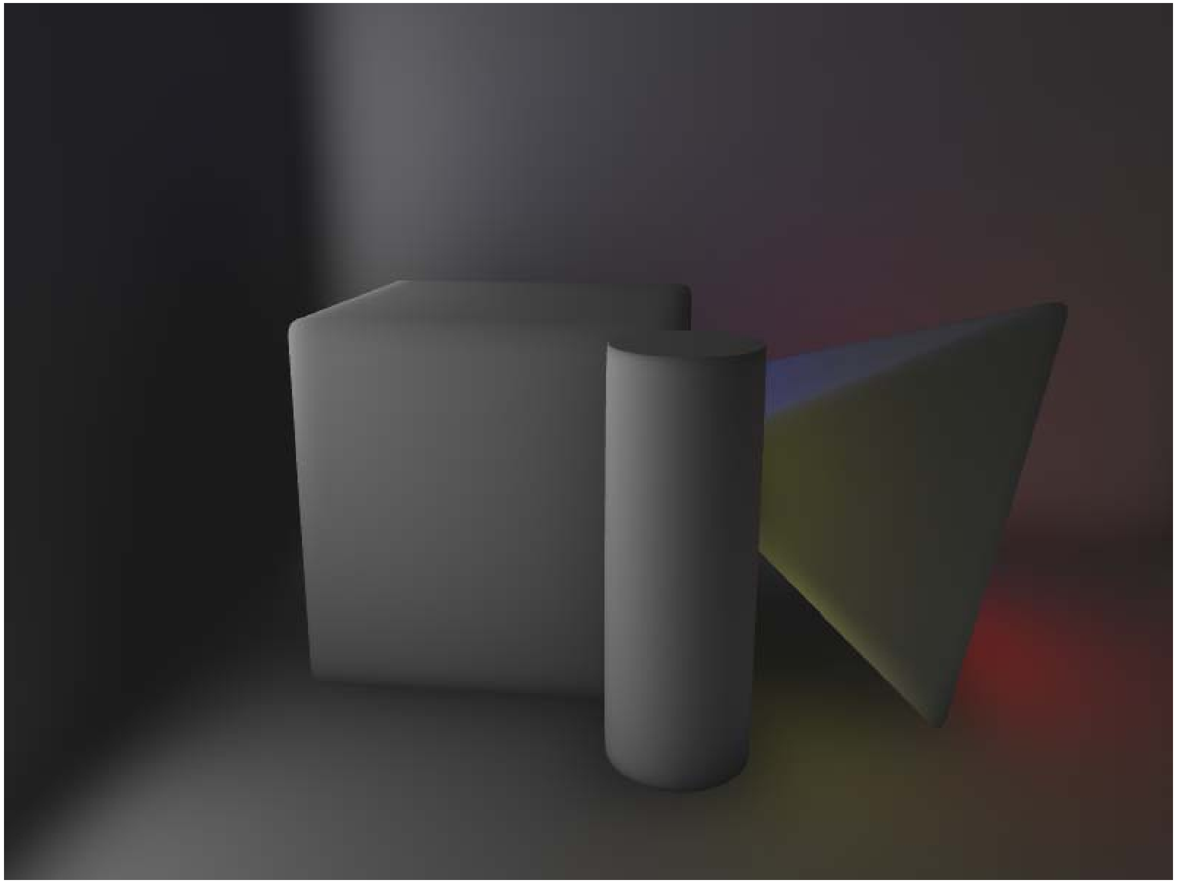
Scene lit with shadow mapped point light source

Indirect Light Pass 1



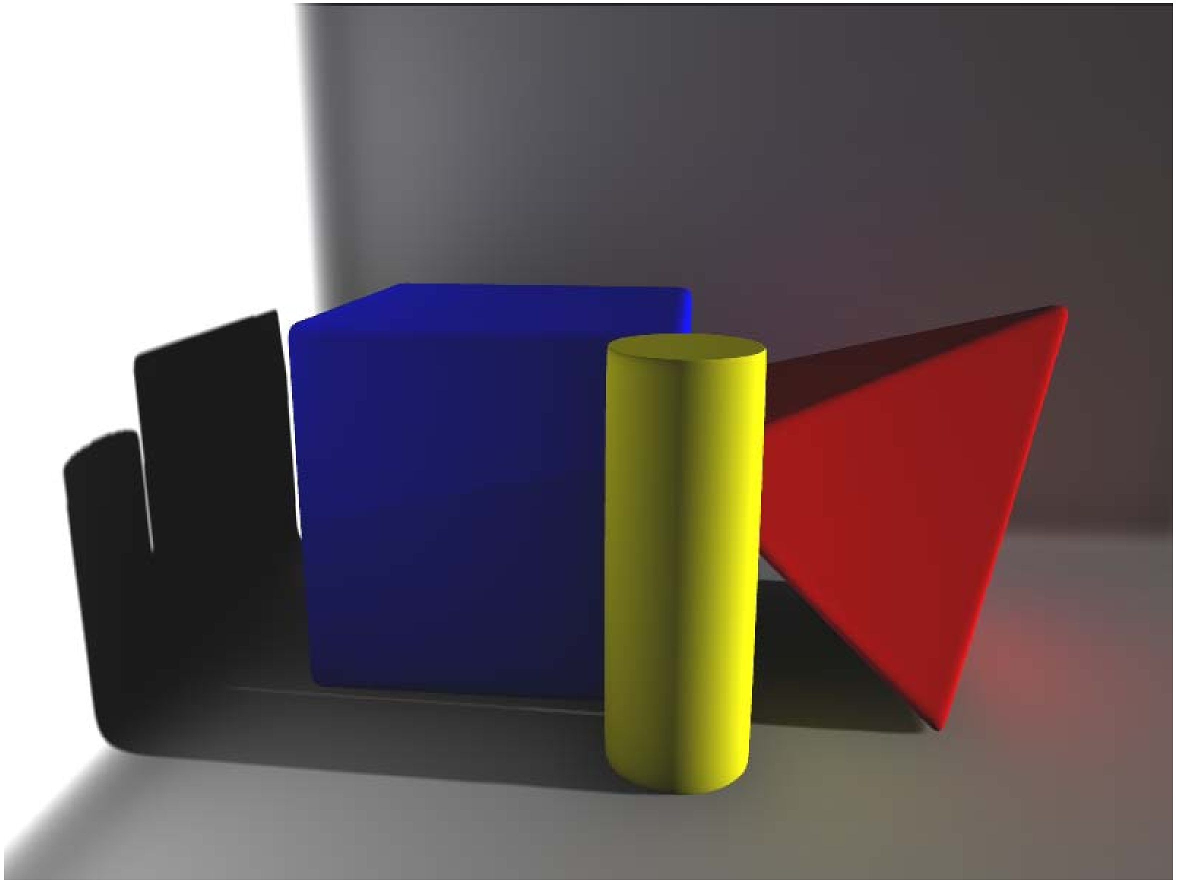
Distribute indirect light in first pass

Indirect Light Pass 2



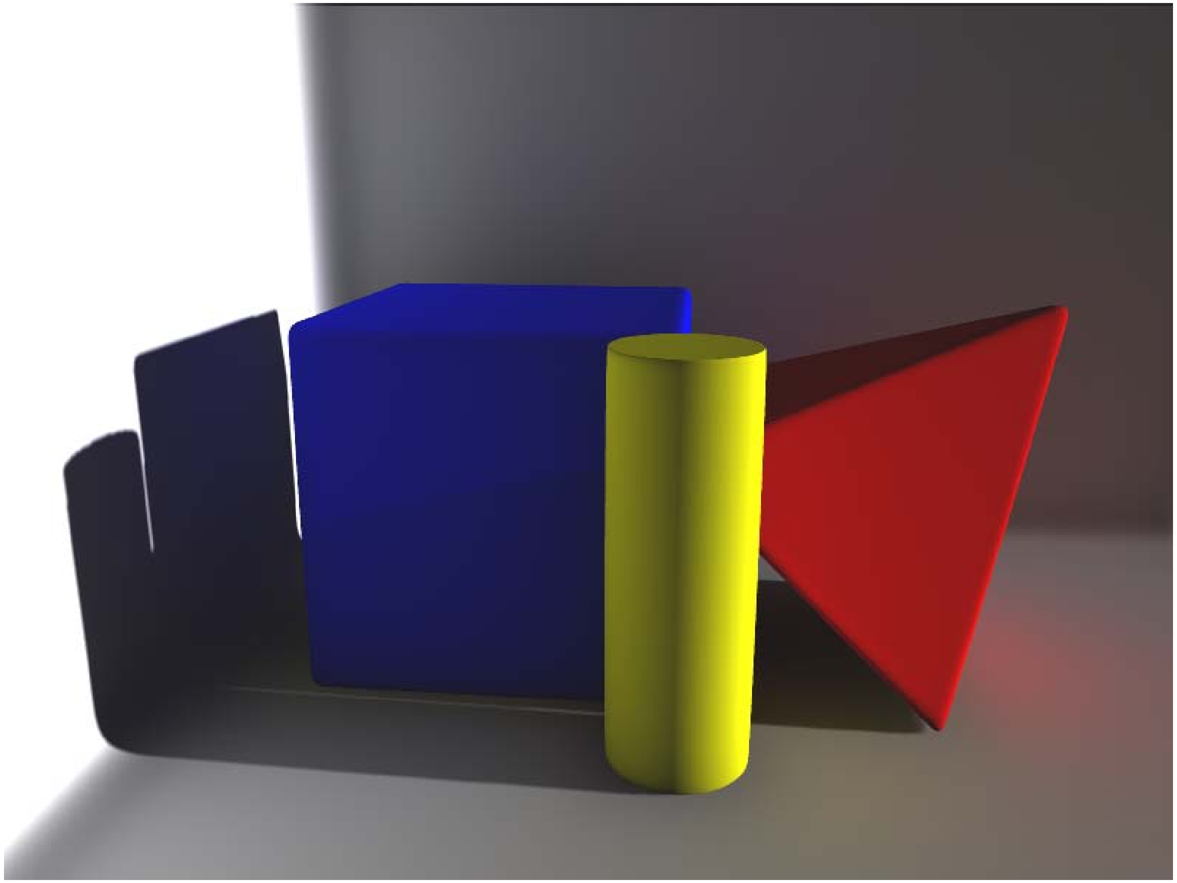
Shadow indirect light in second pass

Direct Light + 1 Bounce Indirect Light



Indirect light * surface color + direct light

Direct Light + 2 Bounces Indirect Light



Second bounce of indirect light takes 2 more passes

Indirect Lighting Shader

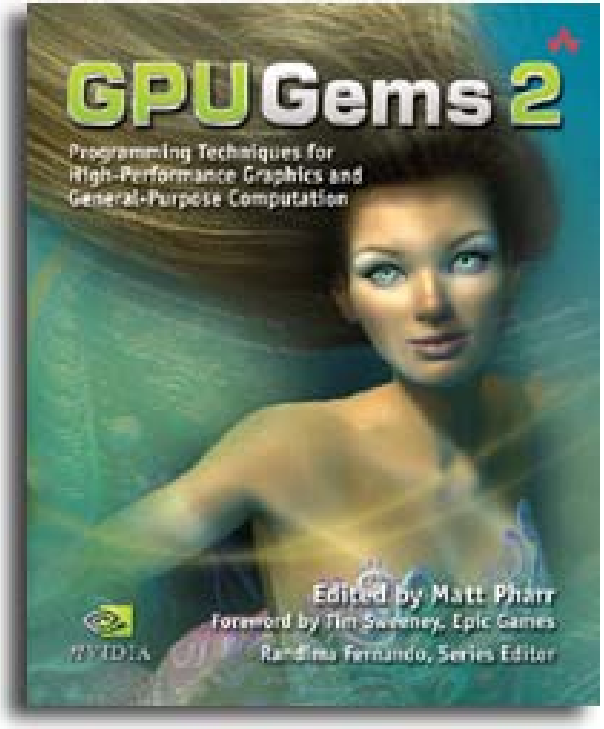
- Use the same basic shader as ambient occlusion
- Uses standard radiosity disk to disk radiance transfer approximation
- First pass distributes 3-component light values
- One or more subsequent passes shadow that light, subtracting from it
- Area lights can use the same shader

Applications

- Shadow environment lighting of non-rigid objects
- Indirect lighting
- Area lights
- Subsurface scattering*
- Accelerate generation of
  - pre-computed radiance transfer data
  - light maps
  - ambient occlusion data

GPU Gems 2

Programming Techniques for High-Performance Graphi and General-Purpose Computation


GPU Gems 2
Programming Techniques for High-Performance Graphics and General-Purpose Computation
Edited by Matt Pharr
Foreword by Tim Sweeney, Epic Games
Randima Fernando, Series Editor
NVIDIA

- **880 full-color pages, 330 figures, hard cover**
- **$59.99**
- **Experts from universities and industry**

“The topics covered in *GPU Gems 2* are critical to the next generation of game engines.”

— Gary McTaggart, Software Engineer at Valve, Creators of Half-Life and Counter-Strike

“*GPU Gems 2* isn’t meant to simply adorn your bookshelf—it’s required reading for anyone trying to keep pace with the rapid evolution of programmable graphics. If you’re serious about graphics, this book will take you to the edge of what the GPU can do.”

—Rémi Arnaud, Graphics Architect at Sony Computer Entertainment