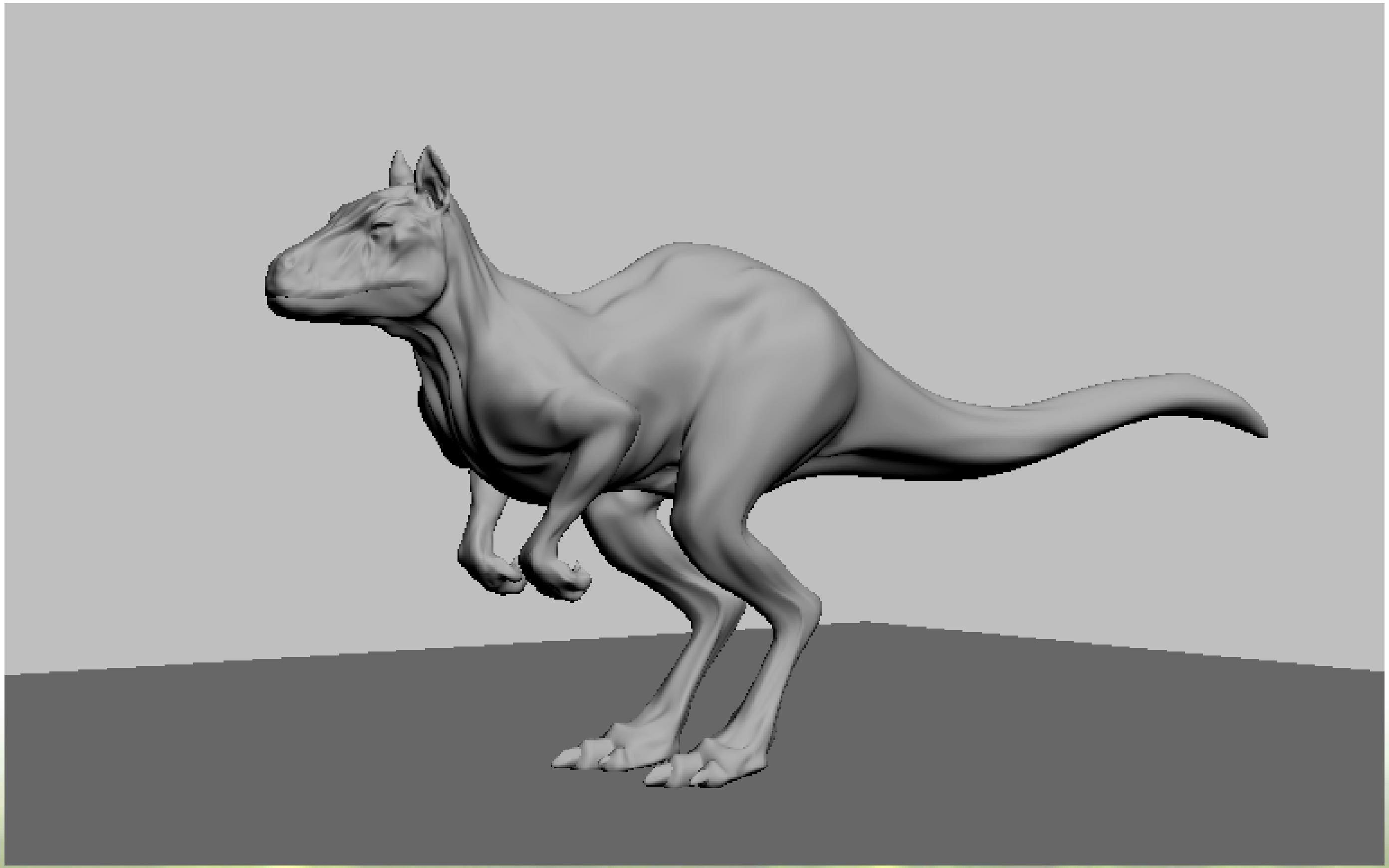
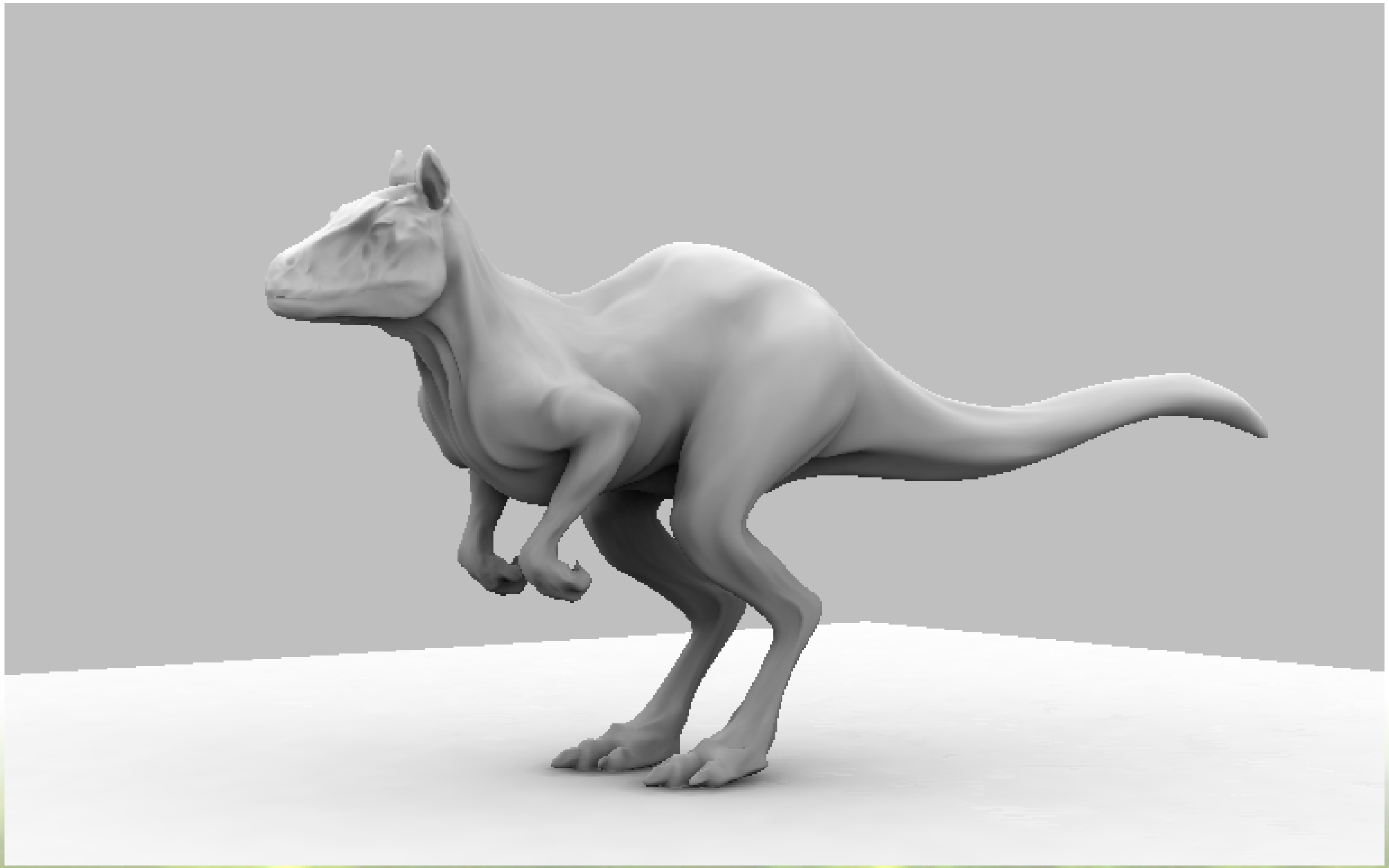
Ambient Occlusion

Matt Pharr
NVIDIA

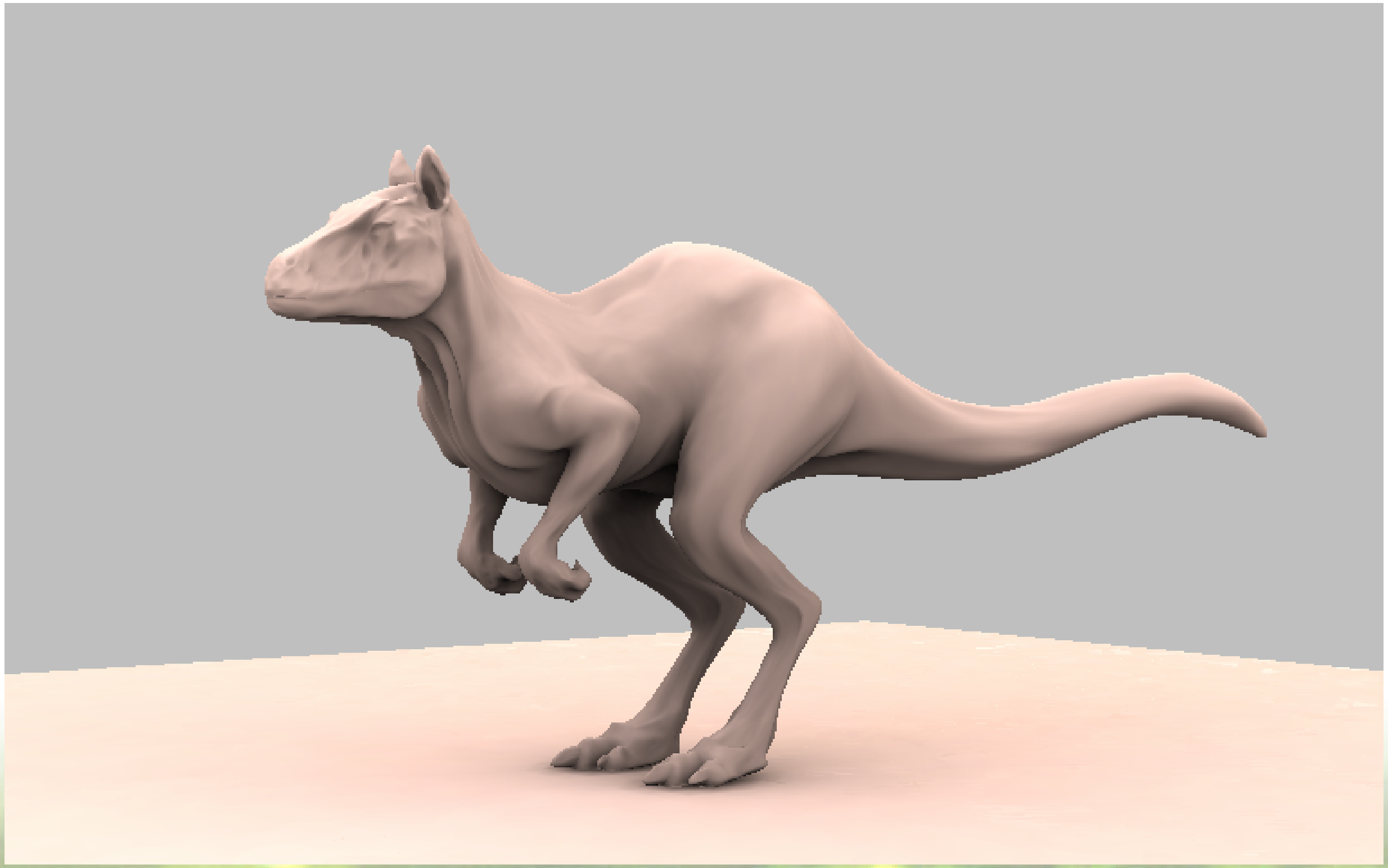
Diffuse Shading



Diffuse + Ambient Occlusion



Diffuse + Ambient Occlusion + Environment Lighting



Overview

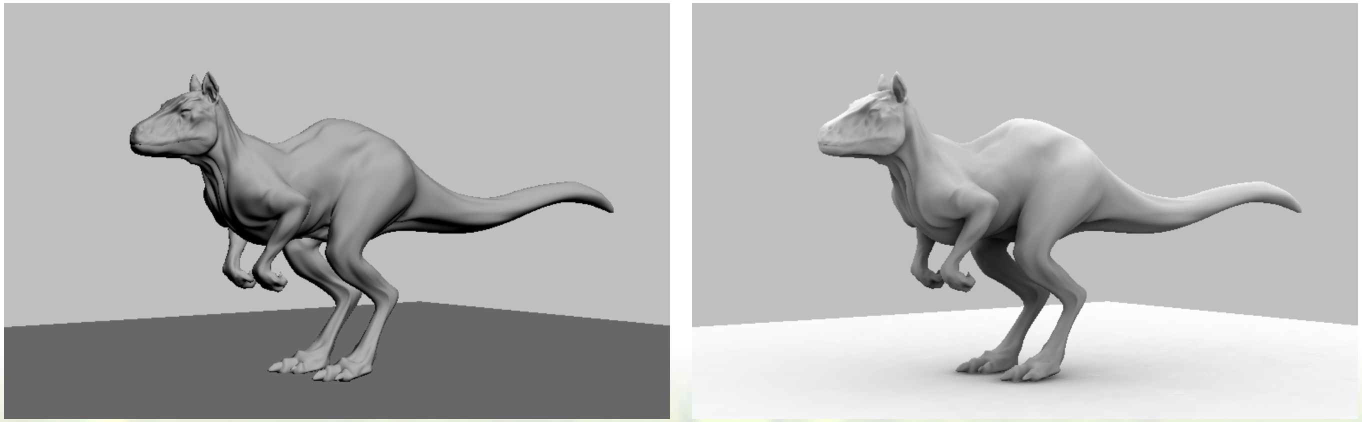
- Light scattering from surfaces
- Main Idea:
  - Use occlusion to modulate lighting
- Implementation details
- Demo
- Further resources

What do point lights and reflection maps have in common?

- Point or distant lights
  - Sum over light from single directions
- Environment maps
  - Add light from single reflected/transmitted direction
- Both are discrete: ignore continuous directional variation of illumination

Problems with These Simplifications

- Hard shadows
- Perfect, unblurred, reflections
- “Stark” lighting

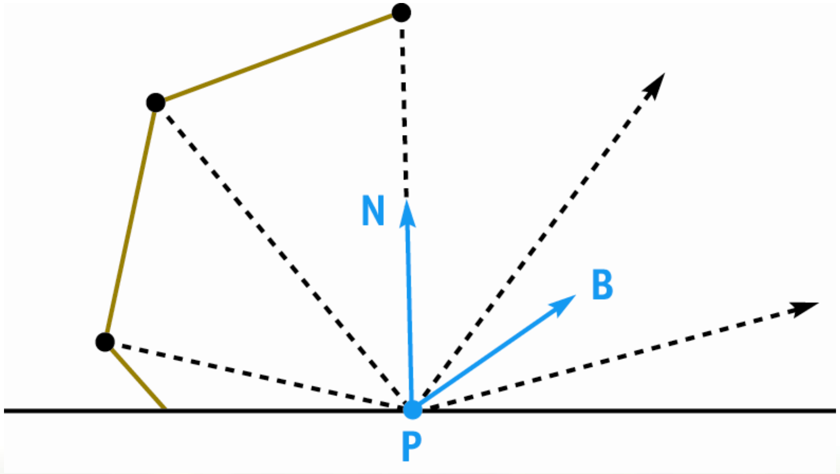


Introducing Directional Variation

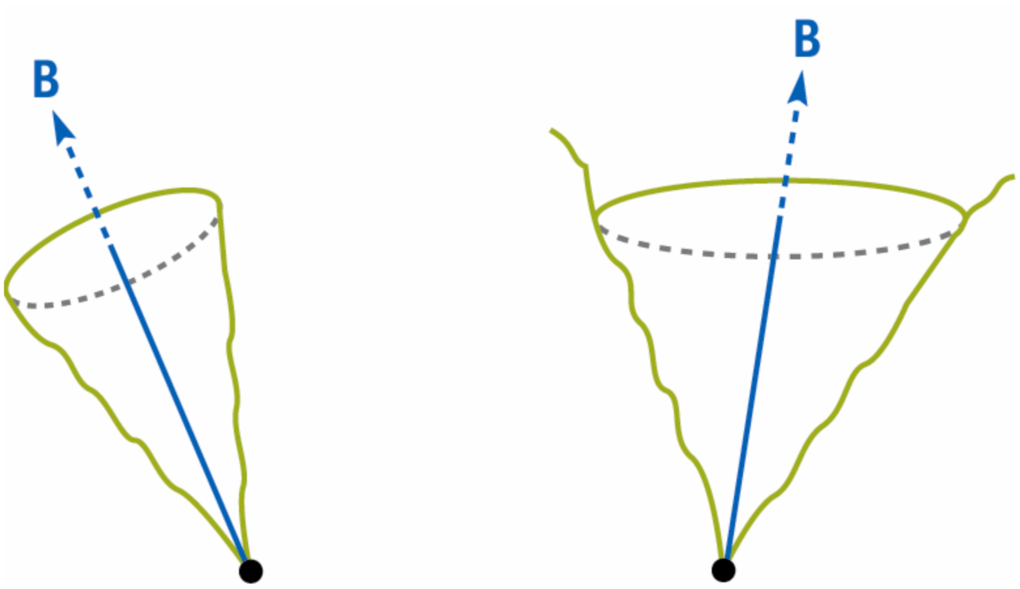
- Ambient occlusion
- Irradiance maps
- Blurred environment maps
- Soft shadows (smoothies, etc.)
- Spherical harmonic lighting

The Basic Idea

Compute fraction of visible hemisphere and average unoccluded direction at P



Cone of Unoccluded Directions Approximation



Computing Occlusion Values

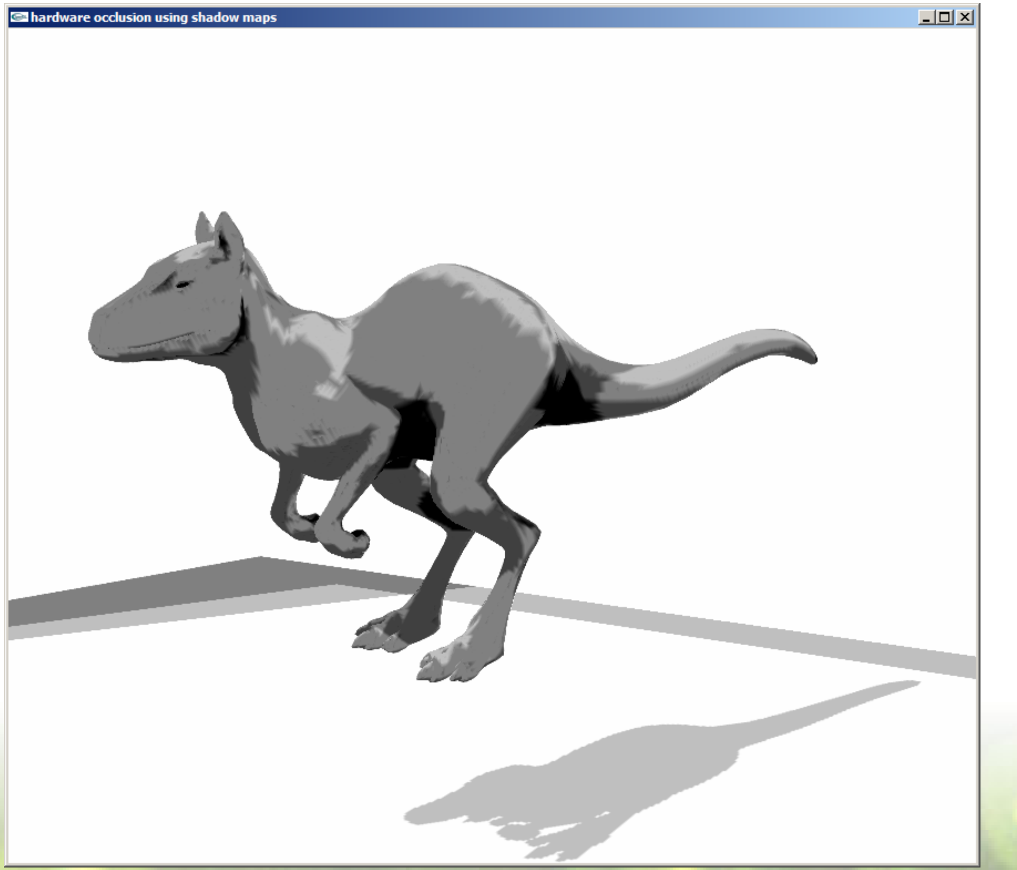
Ray-tracing pre-process

```
generate rays over hemisphere
occlusion = 0
avgDir = (0,0,0)
foreach ray {
      if ray doesn't intersect model
           avgDir += ray.dir
      else ++occlusion;
}
occlusion /= nRays
normalize(avgDir)
```

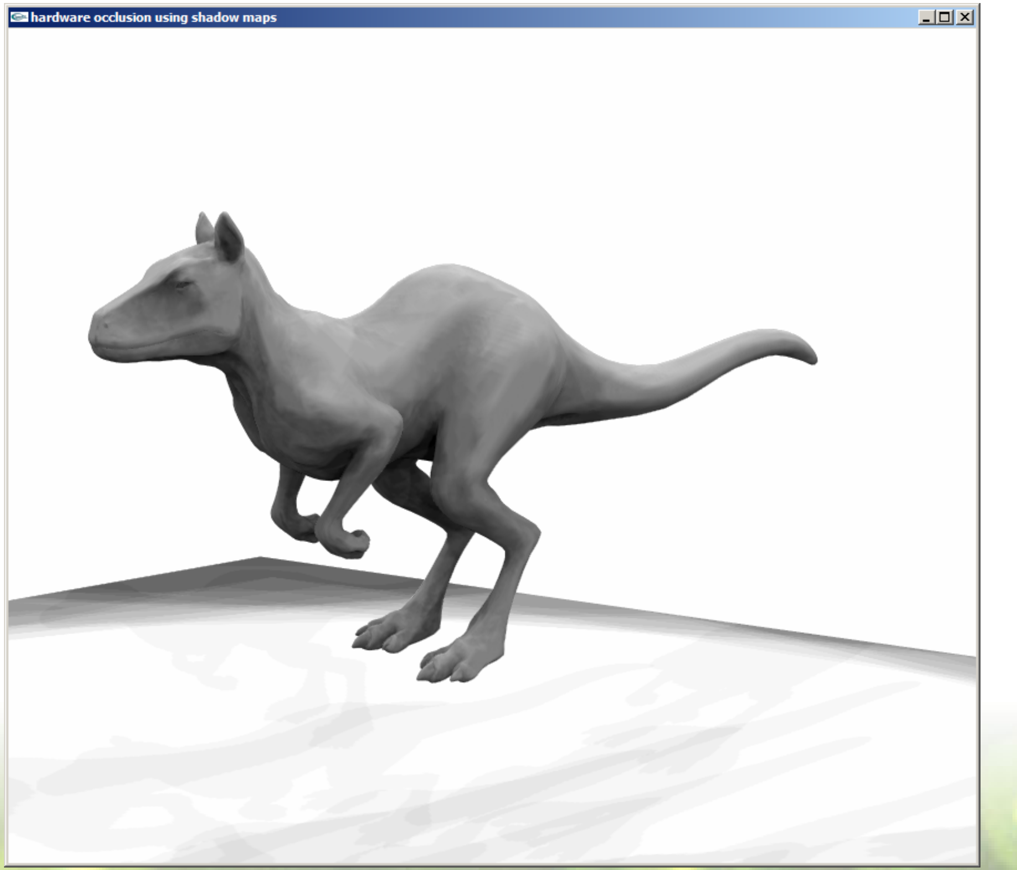
Computing Occlusion Values

- Multi-pass rendering in hardware with shadow maps
  - See article for details

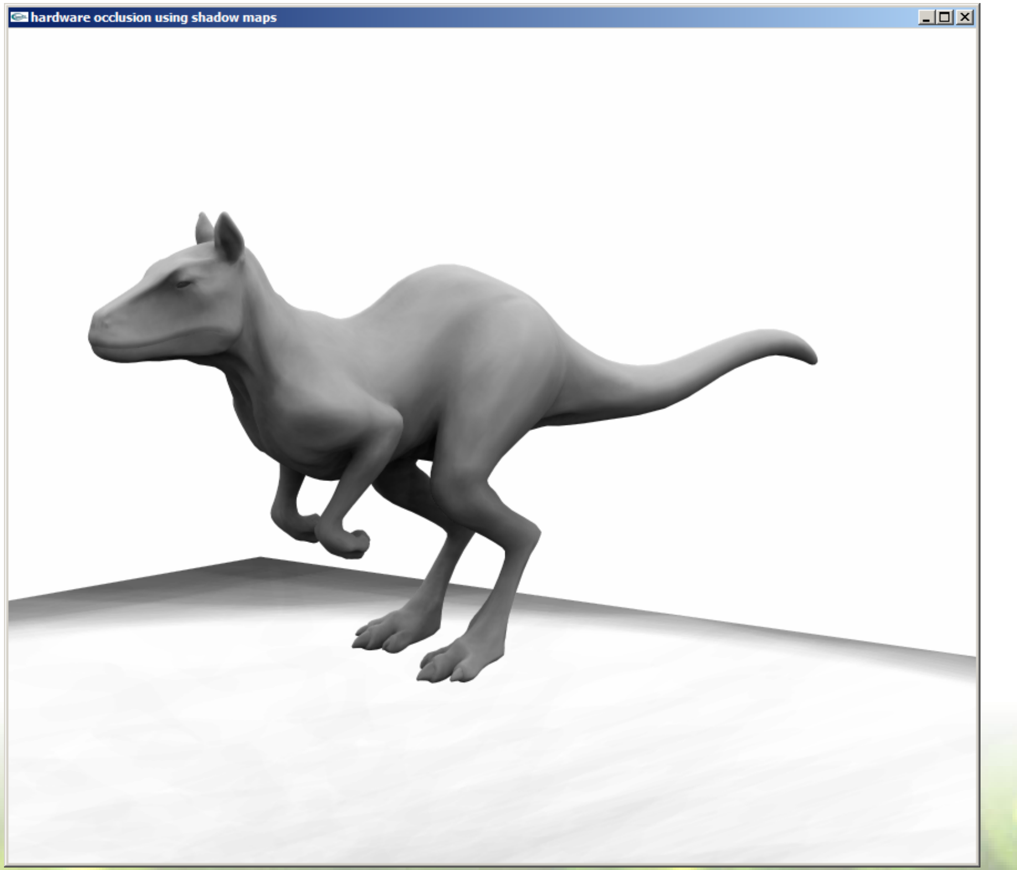
GPU-Generated Occlusion Maps (4 samples)



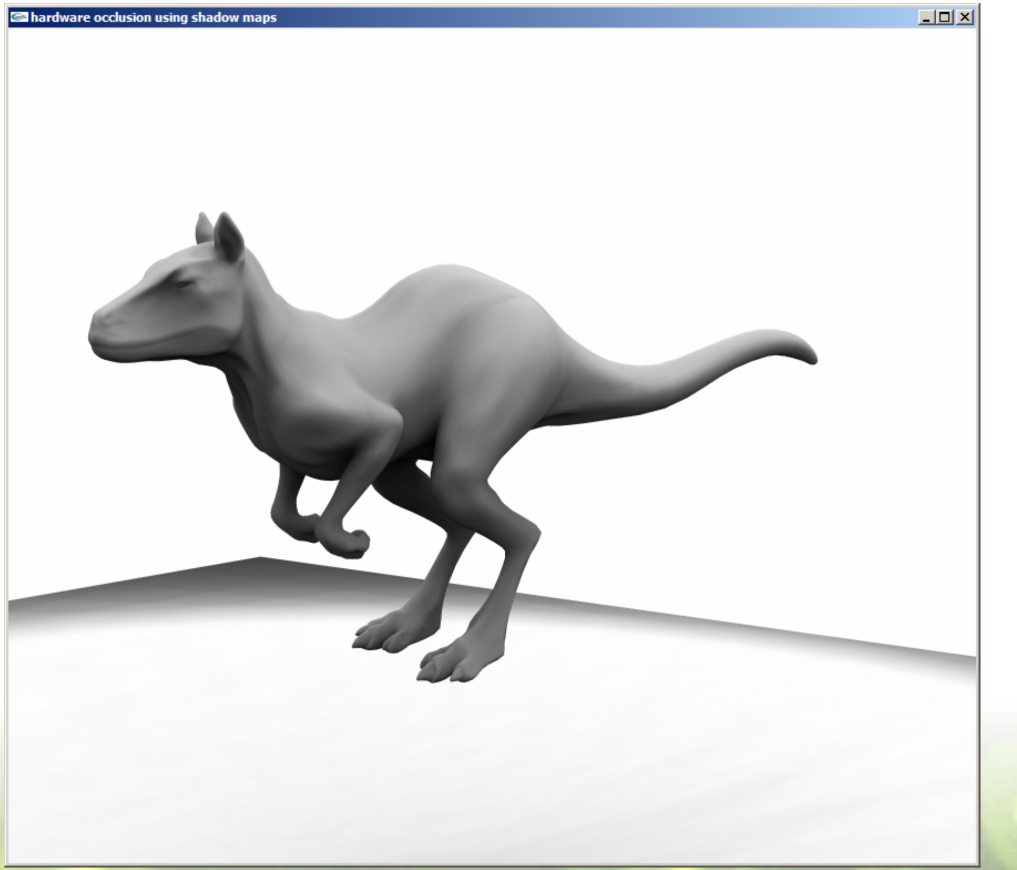
GPU-Generated Occlusion Maps (32 samples)



GPU-Generated Occlusion Maps (128 samples)



GPU-Generated Occlusion Maps (512 samples), ~4 seconds



Using Occlusion Values I

- Simple blurred env map lookup

```
half4 main(half3 avgDir,
           half occlusion,
           half3 Kd) : COLOR {
  half blur = 1 – occlusion; //ad-hoc
  return Kd * (1-occlusion) *
    texCube(envMap, avgDir, blur);
}
```

Using Occlusion Values II

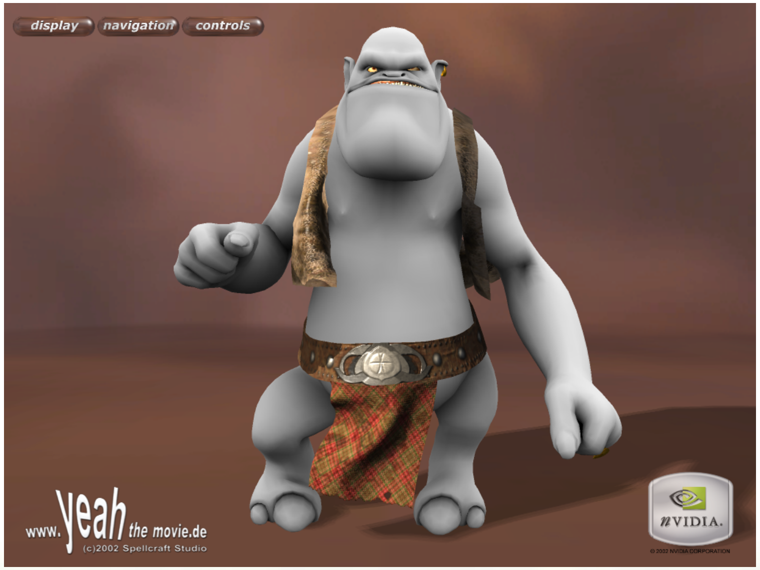
- Modulate irradiance map lookup

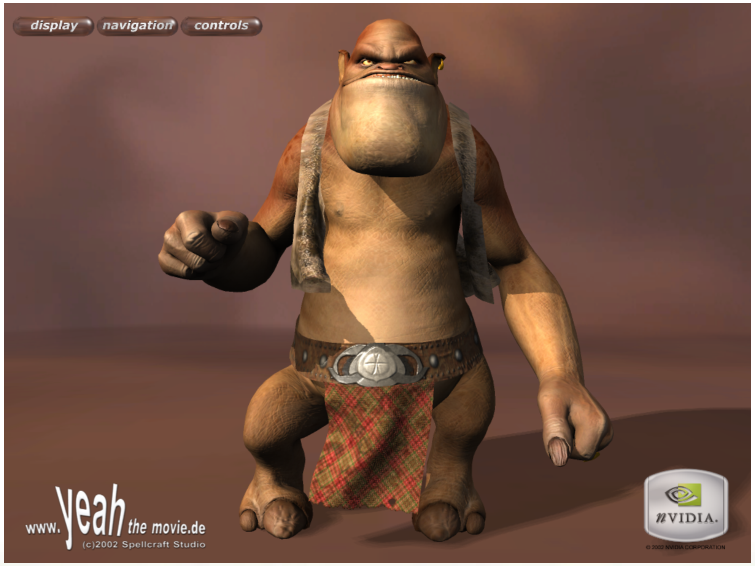
```
half4 main(samplerCUBE irradMap,
           half3 Kd, half occlusion,
           half3 N) : COLOR {
     return Kd * (1-occlusion) *
         texCUBE(irradMap, N);
}
```

Demo

Animated Occlusion

Ogre Demo





Summary

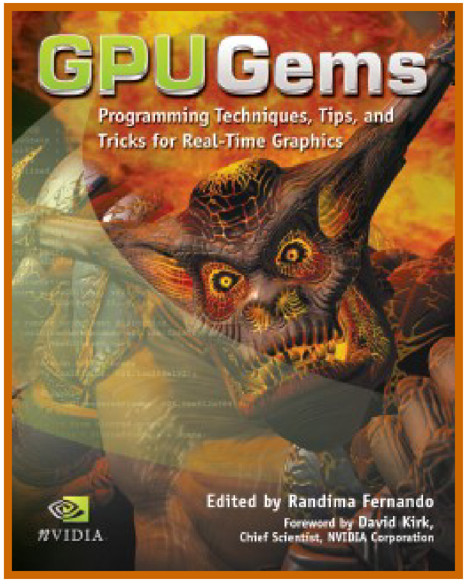
- AO helps reduce the stark “CG” look
  - Medium expensive precomputation
  - Fast rendering
- Special case of some spherical harmonic approaches

Further Reading

- S. Zhukov, A. Iones, G. Kronin, *An Ambient Light Illumination Model*, proc. Eurographics Rendering Workshop '98
- Hayden Landis, *Production-Ready Global Illumination*, "*RenderMan in Production*" SIGGRAPH Course Notes, 2002

Further Resources

- GPU Gems
- FX Composer
  - – *developer.nvidia.com/fxcomposer*
- NVIDIA SDK Version 7
  - – Ambient occlusion demo


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

Questions?

- mpharr@nvidia.com