COM3503/4503/6503: 3D Computer Graphics

Lecture 19: Ray tracing: Part 3

Dr. Steve Maddock, s.maddock@sheffield.ac.uk

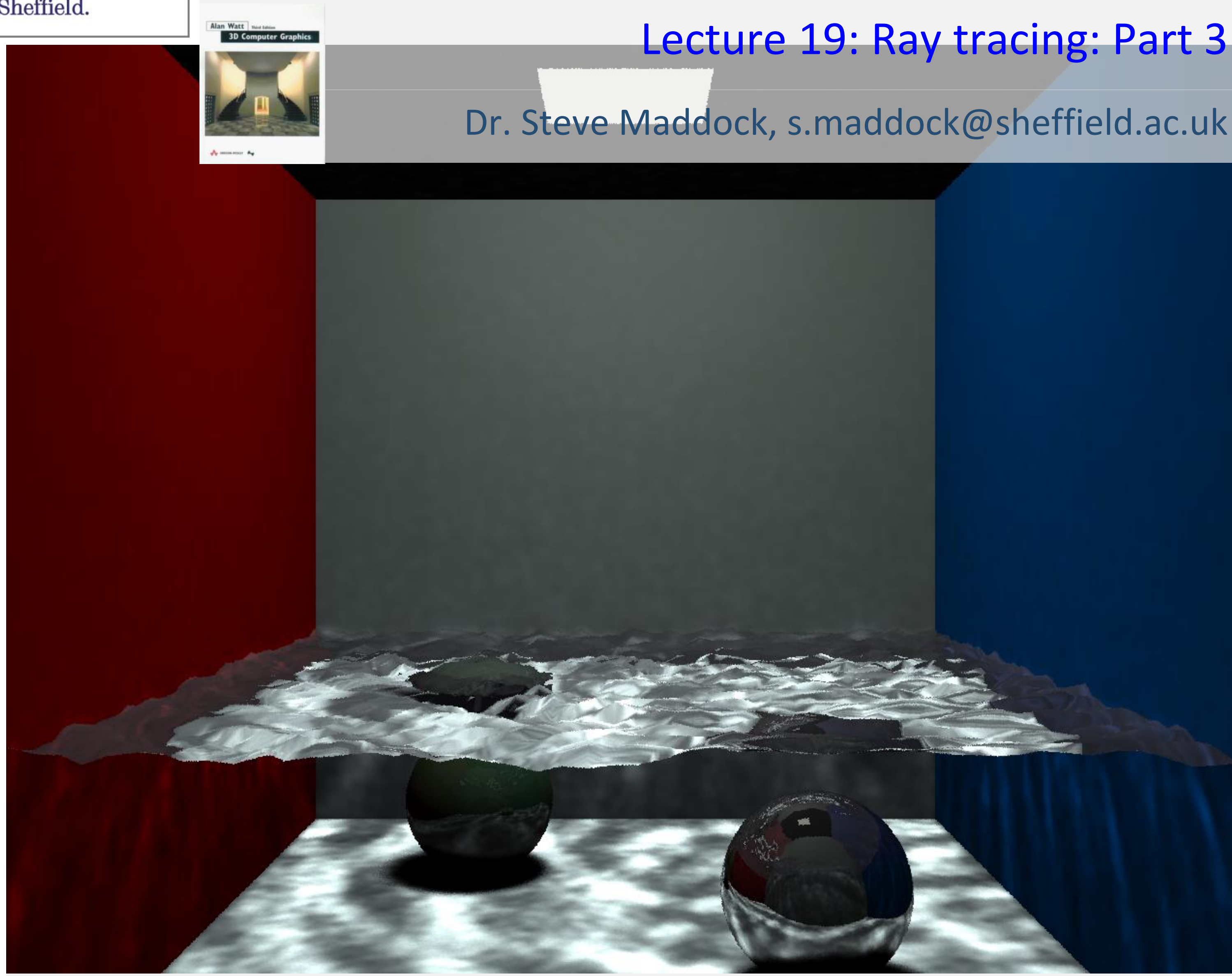
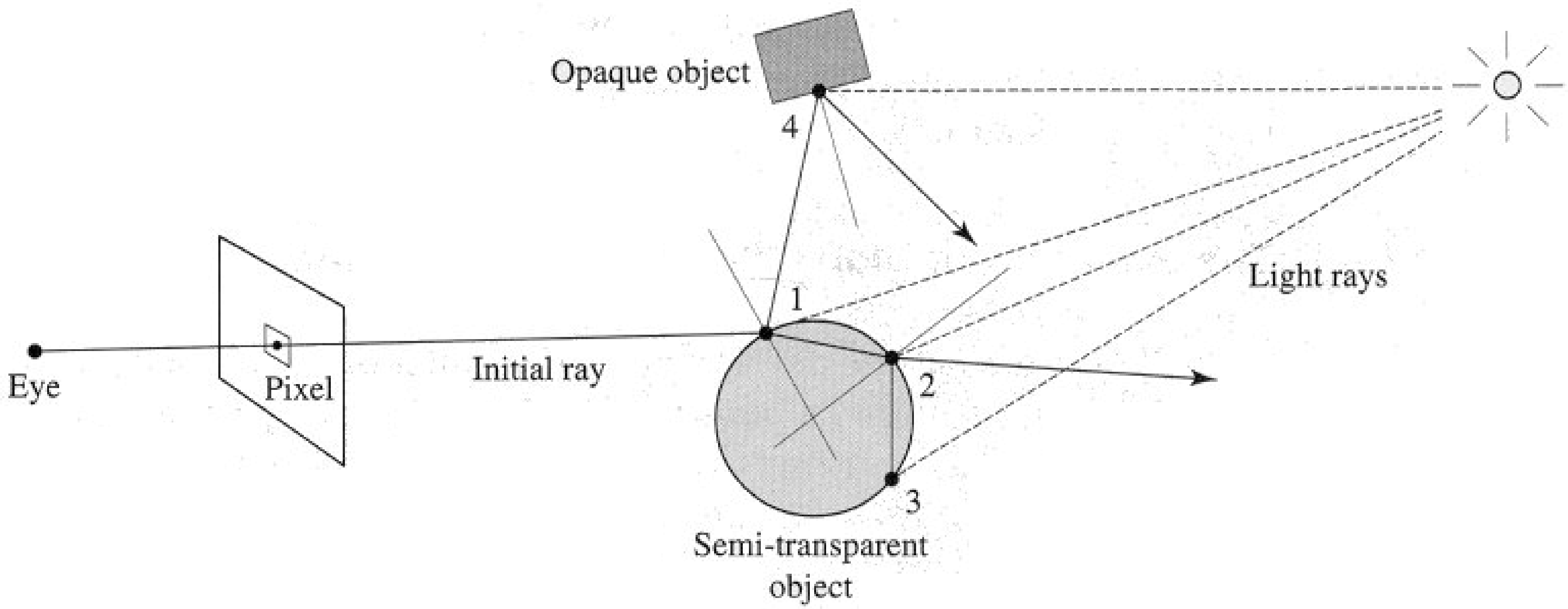


Image: Ian Pozella, “Global Lighting Effects of Caustic Formation”, Undergraduate dissertation project, Department of Computer Science, University of Sheffield, 2012

0. Summary of last lectures

- The global part of the classic (Whitted) algorithm only deals with pure specular-specular interaction
  - Mismatch: Local use of Phong gives blurred specular highlight
- Direct diffuse (but not diffuse-diffuse) is also covered in the local lighting calculation
- Intersection calculations dominate so we need speed ups



1. Part 3

Part 1

- Visible surface ray tracing
- Naïve, recursive (Whitted) ray tracing
    - HSR, Shadows, Reflection, Refraction, Recursion

Part 2

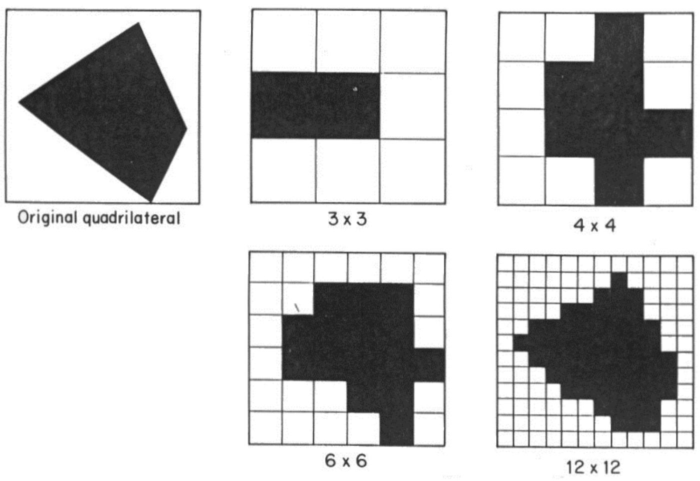
- Intersection calculations
- Speed-up techniques

Part 3

- Anti-aliasing
- Advanced techniques

2. Anti-aliasing

- A higher resolution screen produces less noticeable aliasing effects, e.g. 3x3, 4x4, etc.



- A single ray through the centre of each pixel can lead to sampling issues

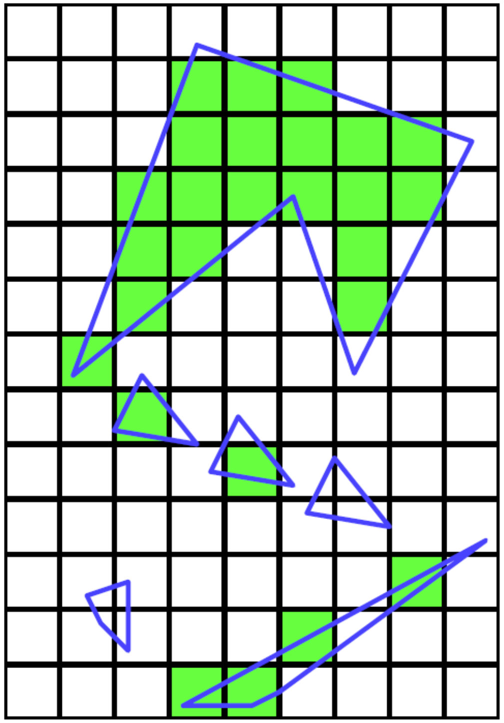
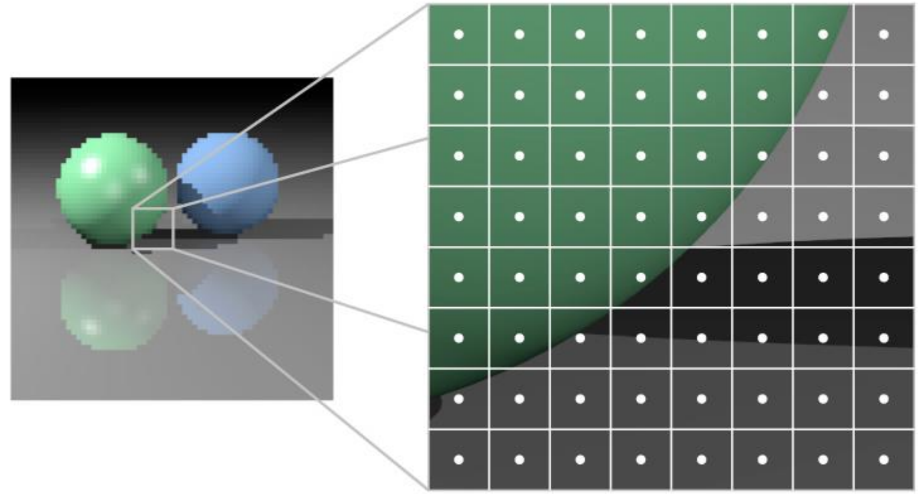


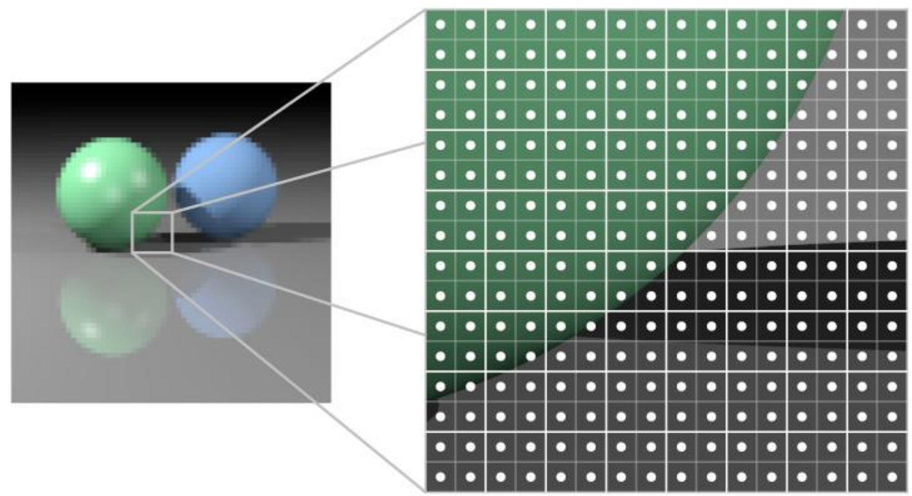
Image from: http://www.cl.cam.ac.uk/teaching/1617/Graphics/materials.html

2.1 Supersampling

- Fire extra rays per pixel, e.g. `n` instead of one
- Then average results to produce single value for the pixel
- Just reduces aliasing problems (see discussion in earlier lecture)



One sample per pixel

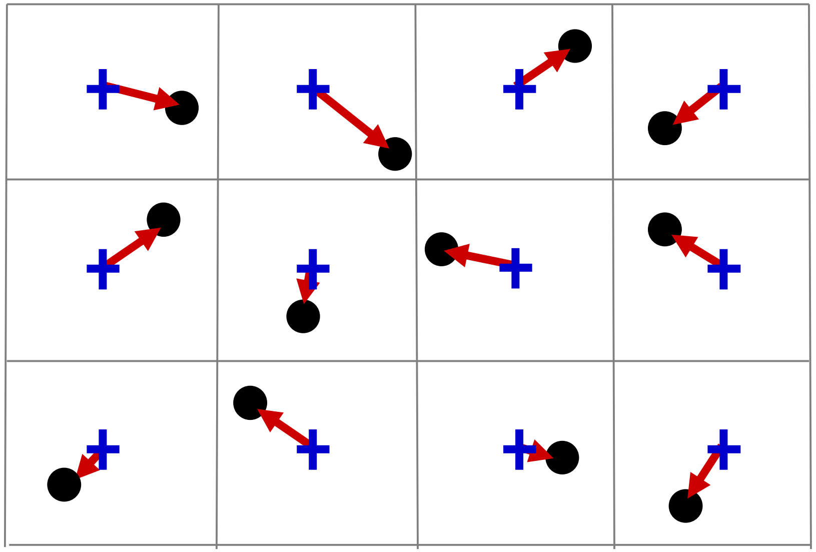


Four samples per pixel

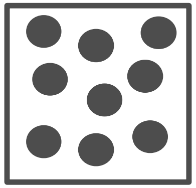
Marschner, 06

2.2 Distributed/distribution ray tracing[1] / Stochastic ray tracing

- Jittering:
  - Uniform sample + random perturbation
- Sampling is now non-uniform / stochastic
- In practice, adds noise to image
  - Trades noise for aliasing
- Use n rays per pixel and jitter each



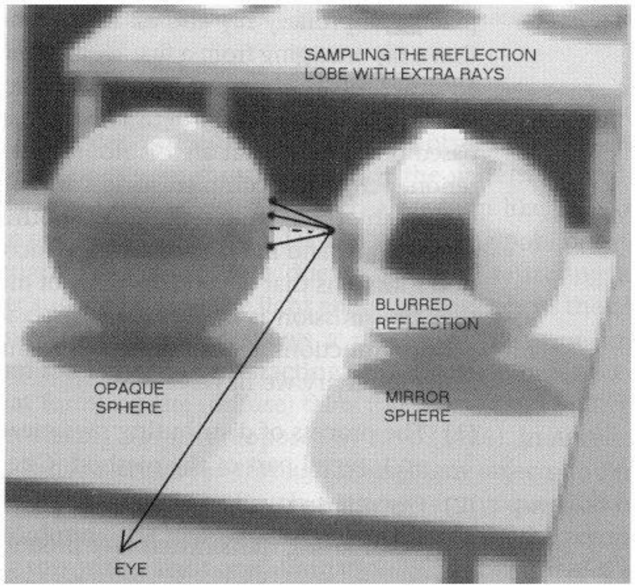
Durand, 06

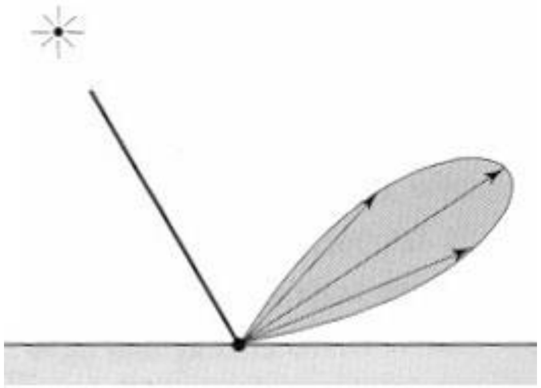


1. Originally called distributed ray tracing. This term is now more usually reserved for ray tracing on parallel processors. Many authors now refer to it as “distribution ray tracing”

2.2 Distributed/distribution ray tracing / Stochastic ray tracing

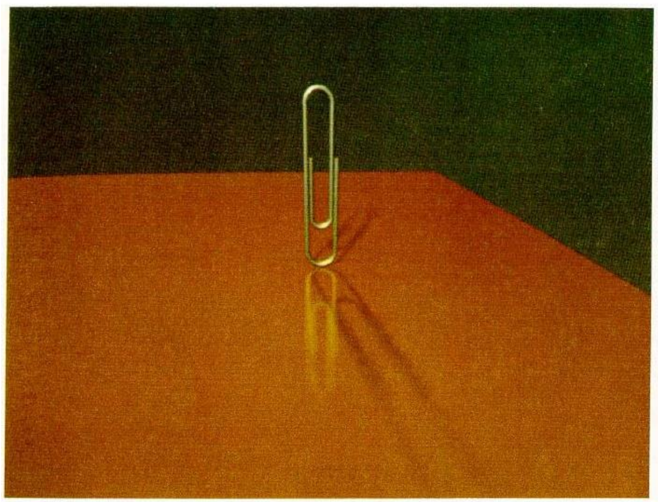
- Example: Imperfect specular interaction
  - At hit point, rather than a single perfectly reflected ray, jitter a few rays to sample the reflection lobe

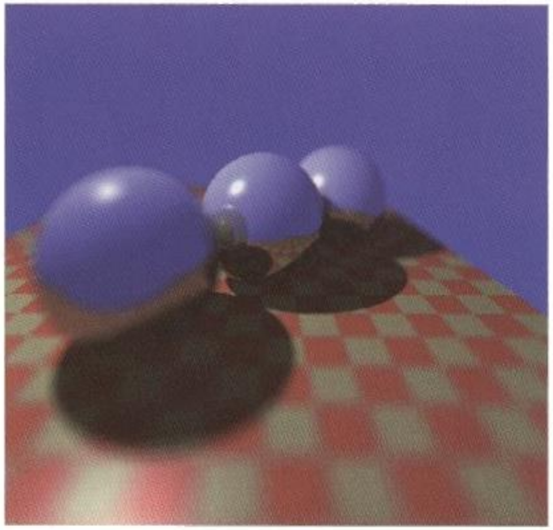




2.2 Distributed/distribution ray tracing / Stochastic ray tracing

- Blurry reflection – distribute reflected rays
- Translucency – distribute transmitted rays
- Depth of field – distribute rays over area of camera lens

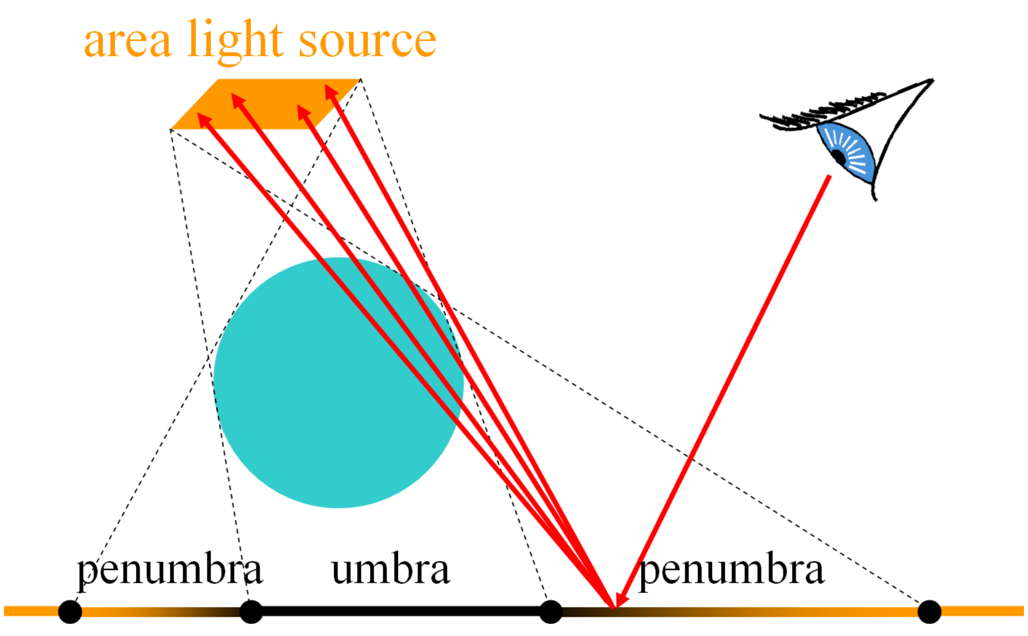




Cook, R.L., T.Porter, L.Carpenter, "Distributed ray tracing", Computer Graphics, 18(3), Jul 84, 137-145.

2.2 Distributed/distribution ray tracing / Stochastic ray tracing

- Penumbra – distribute shadow rays over solid angle subtended by extended light source as seen from the point being shaded
- Some of the rays may see the light source and some may not, thus producing penumbra



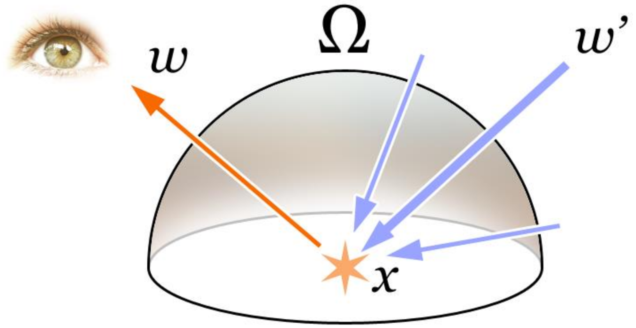
Durand, 06

2.2 Distributed/distribution ray tracing / Stochastic ray tracing

- Motion blur
  – distribute rays in time

Cook, R.L. “Stochastic sampling in computer graphics”, ACM transaction on graphics, 5(1), Jan 86, 51-72

3. Global illumination



- Essentially, all realistic rendering algorithms are an attempt to simulate *the **rendering equation*** (Jim Kajiya, 1986)
- At each point, outgoing light in one direction is the integral of incoming light in all directions multiplied by reflectance property

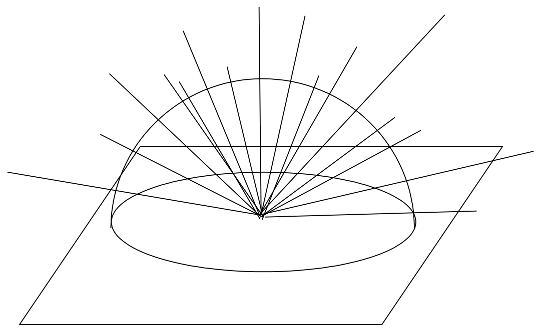
$$L_o(x,\mathbf{w},\lambda,t) = L_e(x,\mathbf{w},\lambda,t) + \int_\Omega f_r(x,\mathbf{w}',\mathbf{w},\lambda,t)L_i(x,\mathbf{w}',\lambda,t)(-\mathbf{w}'\cdot\mathbf{n})d\mathbf{w}'$$

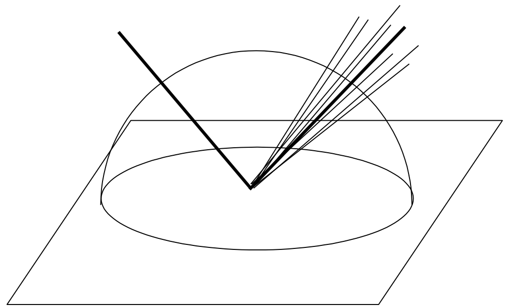
http://en.wikipedia.org/wiki/Rendering_equation

- One approach to solving the equation is based on finite element methods, leading to the radiosity algorithm:
  - Simulates diffuse-to-diffuse light transport
- Another approach using Monte Carlo methods has led to many different algorithms including path tracing and photon mapping, amongst others
  - Monte Carlo methods are a numerical tool to evaluate integrals, and are based on sampling

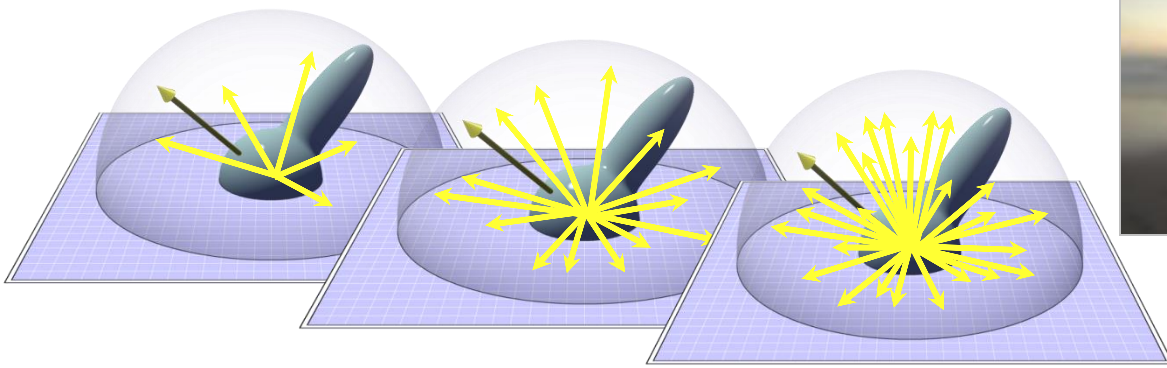
4. Advanced ray tracing

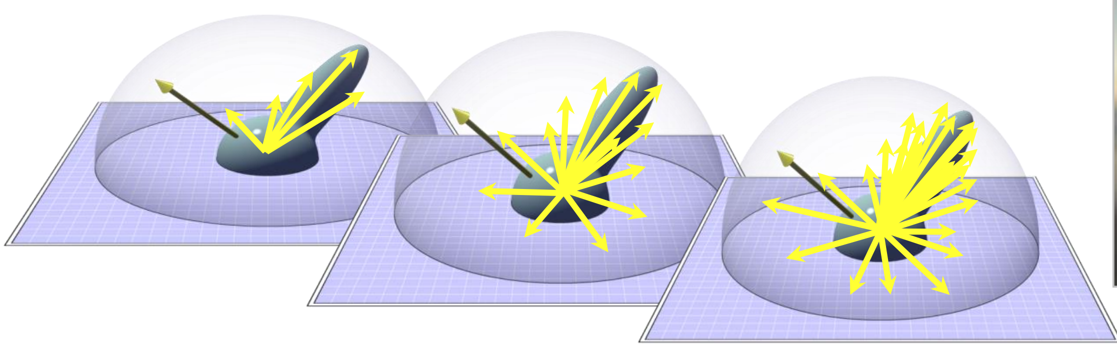
- Essentially for any intersection point we want to integrate over the hemisphere covering the point to calculate all the light reaching the point
- More compute power means a Monte Carlo approach to ray tracing is the natural choice for most realistic image synthesis problems
  - Average many ‘random’ samples over the hemisphere
  - Need many samples to avoid noise in the rendered images
  - Importance sampling – use more samples for important directions

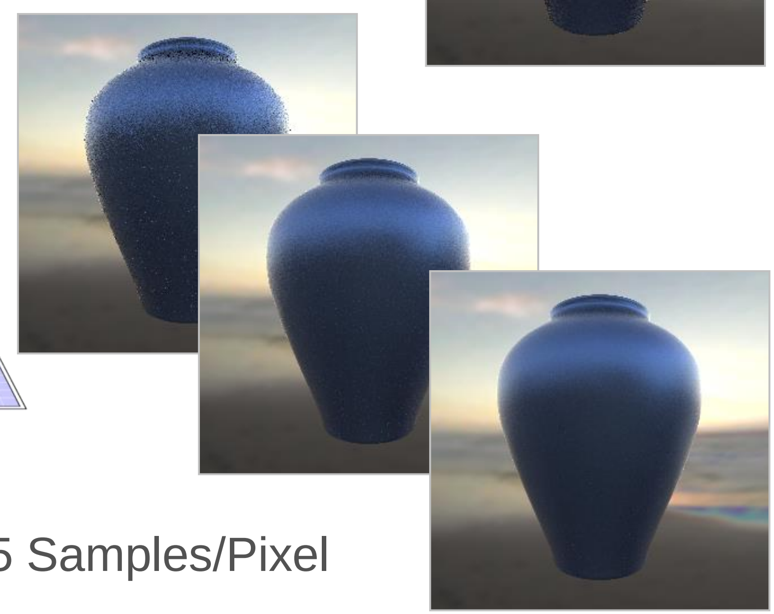




4.1 Random compared with importance sampling







5, 25, 75 Samples/Pixel

Durand, 06 , Slide courtesy of Jason Lawrence

4.2 General algorithm

- Shoot many viewing rays per pixel
  - Distribute over pixel, time, camera lens
- Shoot many indirect rays to sample the hemisphere
- Terminate recursion

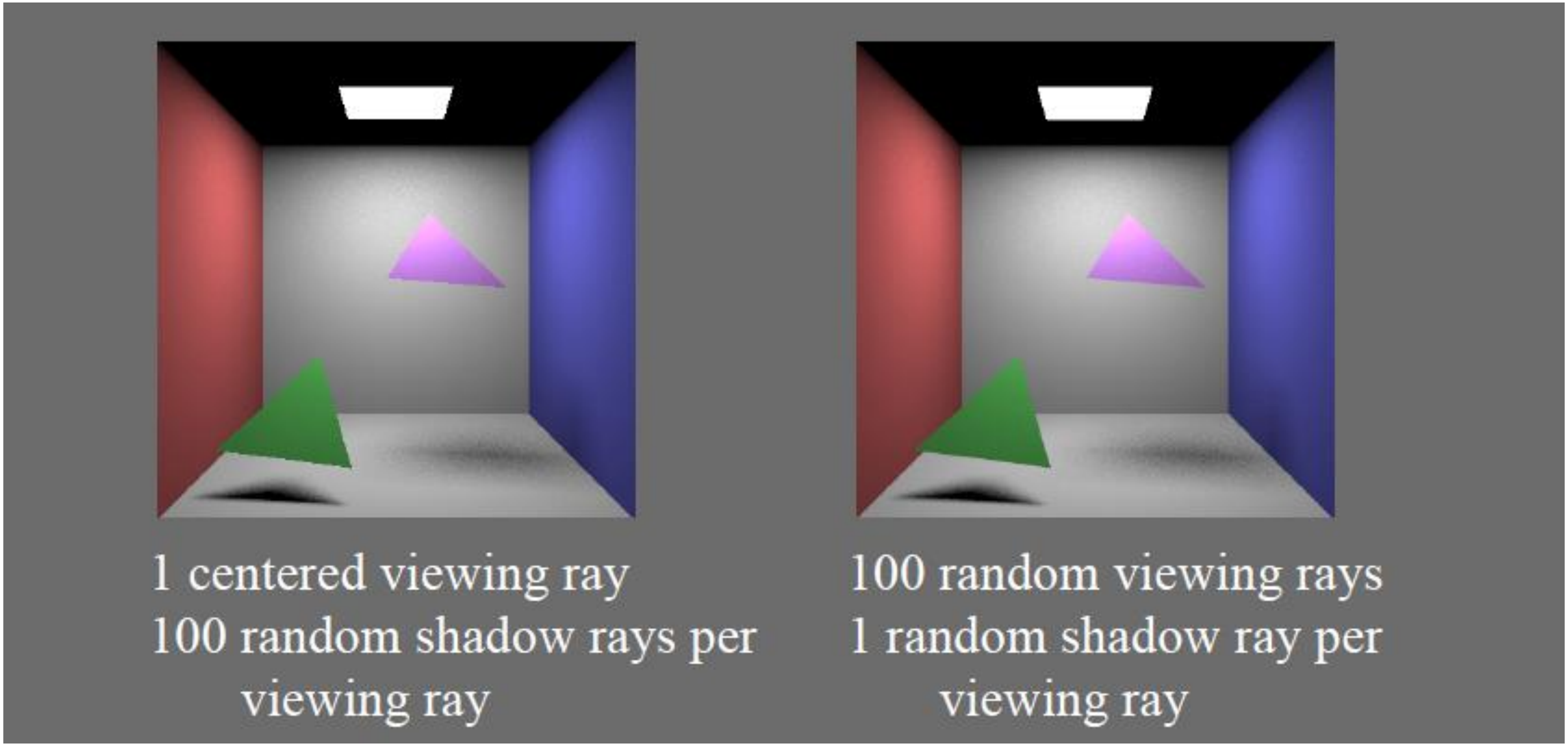
Motion blur

Depth-of-field

Kavita Bala, Cornell University, 2010

4.3 Stochastic ray tracing compared with path tracing

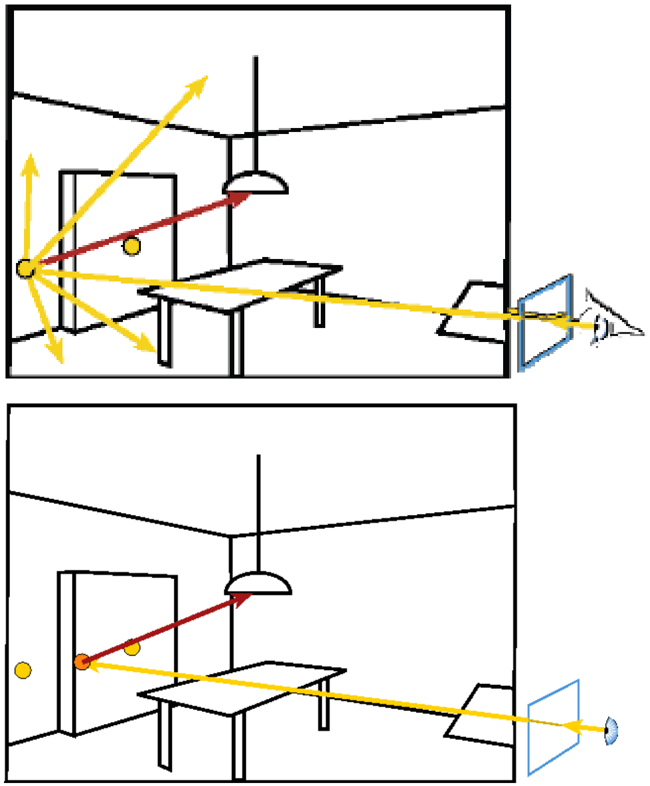
- Path tracing (Kajiya, 86) shoots many rays per pixel, but only shoots one ray at each intersection, in a random direction



Kavita Bala, Cornell University, 2010

4.4 Further advances in ray tracing

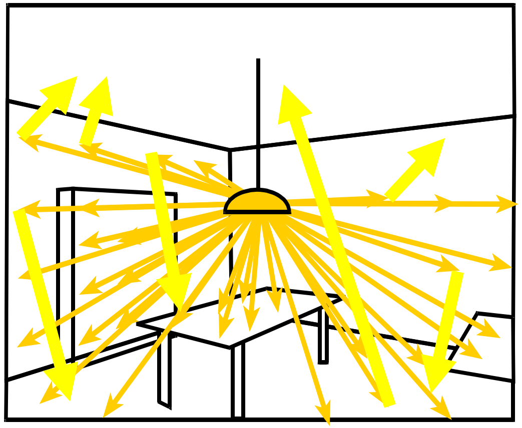
- Irradiance caching (Ward, 88)
  - Store the indirect illumination for diffuse surfaces
  - Interpolate existing cached values
  - But do full calculation for direct lighting

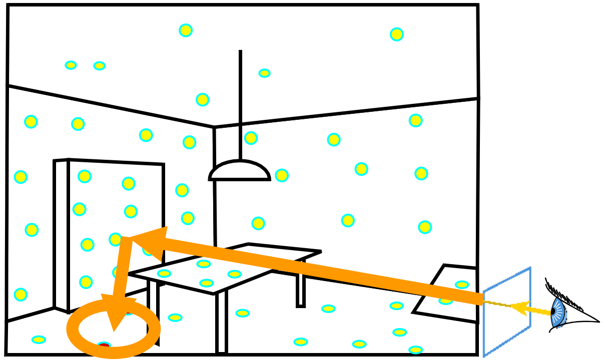


Durand, 06

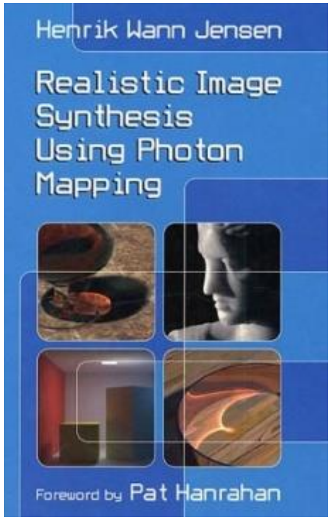
4.4 Further advances in ray tracing

- Photon Mapping (Jensen, 2001)
  - Preprocess: cast rays from light sources
  - Store photons in kd-tree
  - Cast primary rays
  - For secondary rays, reconstruct irradiance using k closest photons





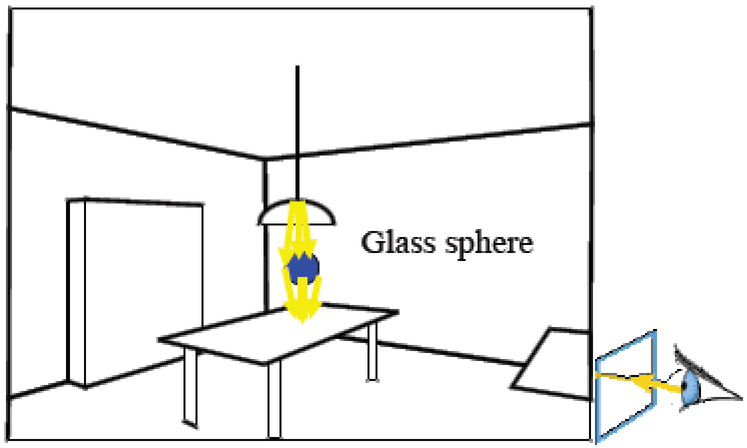
Durand, 06



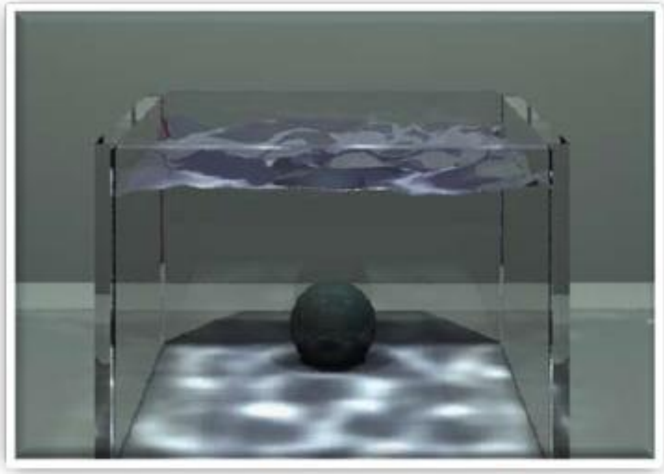
Caustics demo: [Chrome&WebGL water]

4.4 Further advances in ray tracing

- Photon Mapping (Jensen, 2001)
    - Special photon map for caustics



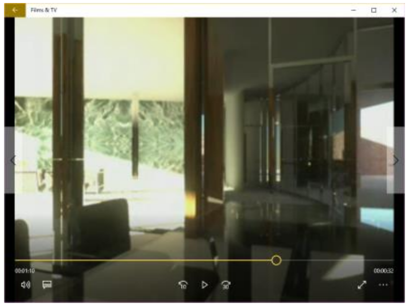
Durand, 06



- Ian Pozella, "Global Lighting Effects of Caustic Formation", Undergraduate dissertation project, Department of Computer Science, University of Sheffield, 2012

4.5 Photon mapping

- Animation by Henrik Wann Jensen (http://graphics.ucsd.edu/~henrik/)



5. Lightwork Design Sheffield

https://www.lightworkdesign.com/gallery_inner.php?gallery_cat=6

6. **NVIDIA® OptiX™ ray tracing engine**

- "The NVIDIA® OptiX™ Ray Tracing Engine is a programmable ray tracing framework helping software developers build ray tracing applications in a fraction of the time of conventional methods, that then run exceedingly fast on NVIDIA GPUs" (https://developer.nvidia.com/gameworks-optix-overview)

OptiX™: A General Purpose Ray Tracing Engine

Steven G. Parker[1*] James Bigler[1] Andreas Dietrich[1] Heiko Friedrich[1] Jared Hoberock[1] David Luebke[1]
David McAllister[1] Morgan McGuire[1,2] Keith Morley[1] Austin Robison[1] Martin Stich[1]
NVIDIA[1] Williams College[2]

Figure 1: *Images from various applications built with OptiX. Top: Physically based light transport through path tracing. Bottom: Ray tracing of a procedural Julia set, photon mapping, large-scale line of sight and collision detection, Whitted-style ray tracing of dynamic geometry, and ray traced ambient occlusion. All applications are interactive.*

ACM Transactions on Graphics (SIGGRAPH 2010)

7. Summary

- We trace rays from the eye into the scene
- The global part of the classic (Whitted) algorithm only deals with pure specular-specular interaction
- Direct diffuse (but not diffuse-diffuse) is also covered by the local light calculation
- Intersection calculations dominate and we need speed-up techniques
- In advanced approaches to global illumination we use better approximations to the ‘rendering equation’
- Advanced ray tracing approaches use a Monte Carlo approach