# CS-184: Computer Graphics

Lecture #10: Raytracing

Prof. James O'Brien
University of California, Berkeley

V2005F-10-1.0

# Today

- Raytracing
  - Shadows and direct lighting
  - Reflection and refraction
  - Antialiasing, motion blur, soft shadows, and depth of field
- Intersection Tests
  - Ray-primitive
  - Sub-linear tests

# Light in an Environment

**Lady writing a Letter with her Maid**
National Gallery of Ireland, Dublin
Johannes Vermeer, 1670



# Global Illumination Effects

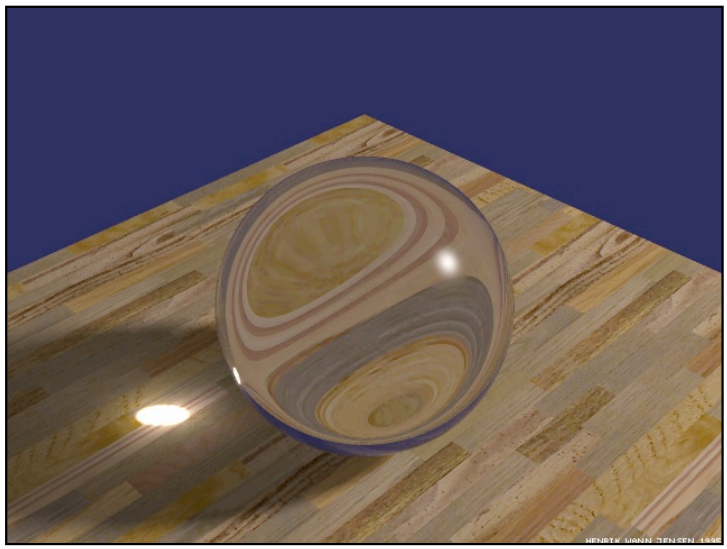
**PCKTWTCH**
Kevin Odhner
POV-Ray

# Global Illumination Effects


**A Philco 6Z4 Vacuum Tube**
Steve Anger
POV-Ray

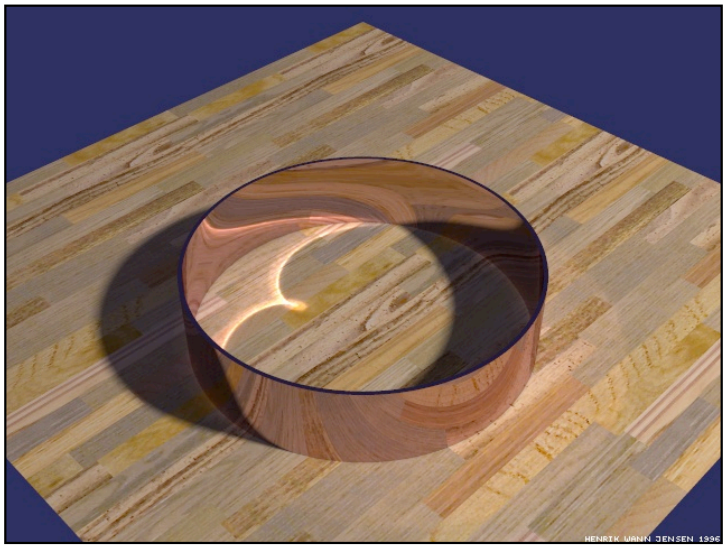


# Global Illumination Effects



**Caustic Sphere**
Henrik Jensen
(refraction caustic)

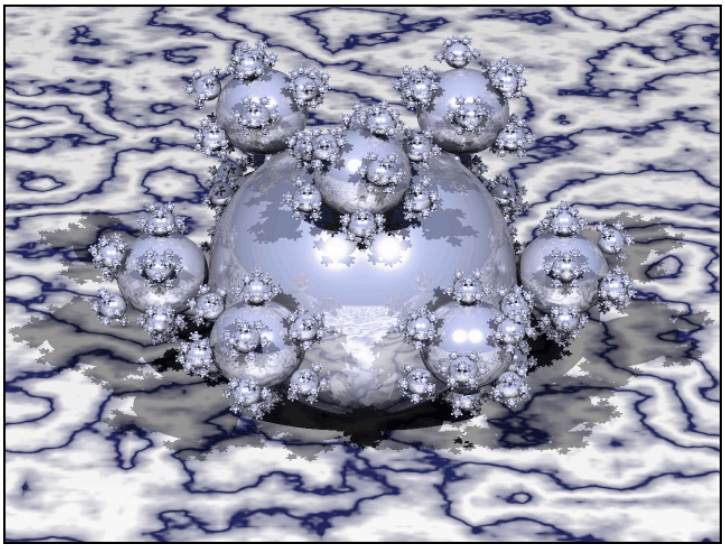
# Global Illumination Effects




**Caustic Ring**
Henrik Jensen
(reflection caustic)

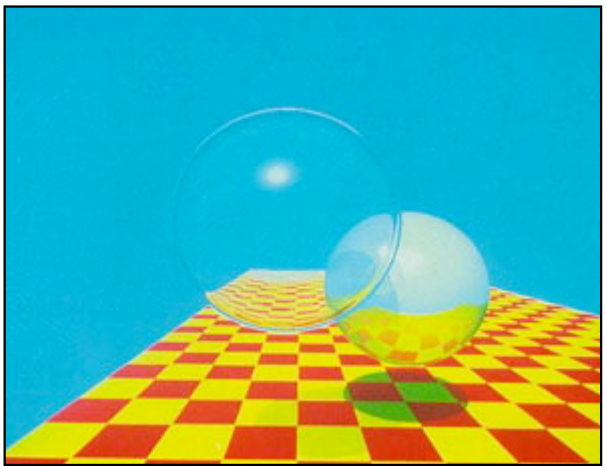


# Global Illumination Effects



**Sphere Flake**
Henrik Jensen

# Early Raytracing



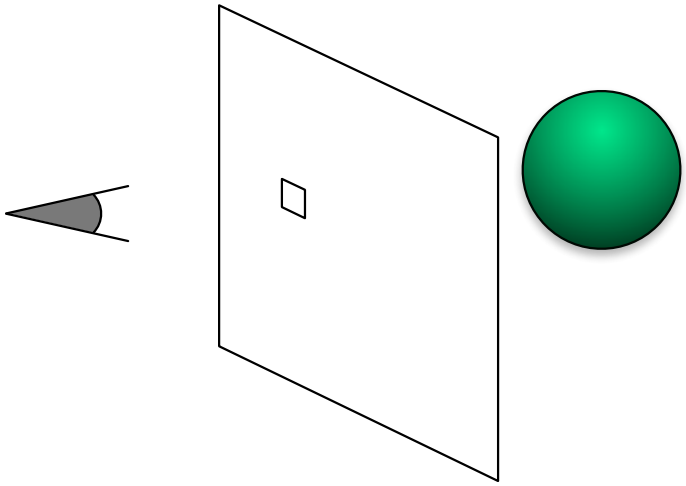
Turner Whitted



# Raytracing

- Scan conversion
  - 3D → 2D → Image
  - Based on transforming geometry
- Raytracing
  - 3D → Image
  - Geometric reasoning about light rays

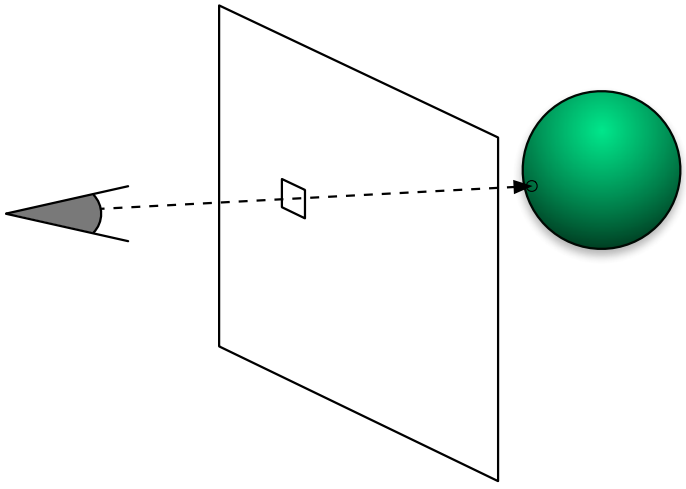
# Raytracing



Eye, view plane section, and scene

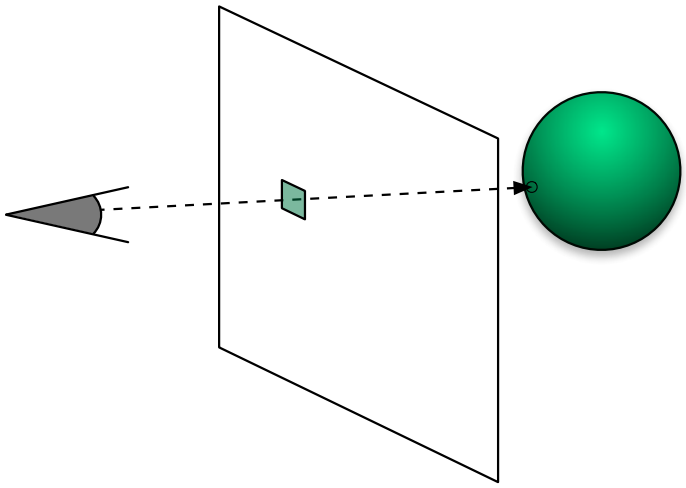


# Raytracing



Launch ray from eye through pixel, see what it hits

# Raytracing



Compute color and fill-in the pixel

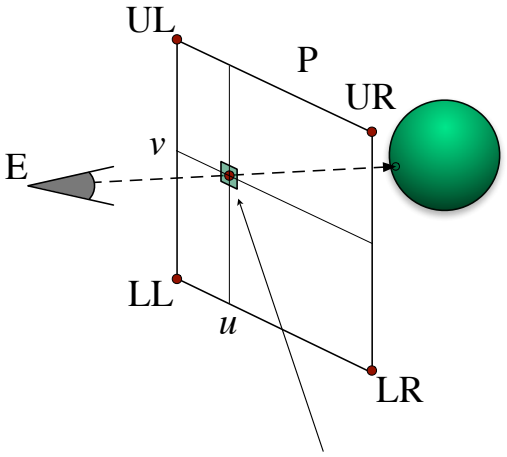


# Raytracing

- Basic tasks
  - Build a ray
  - Figure out what a ray hits
  - Compute shading

# Building Eye Rays

- Rectilinear image plane build from four points

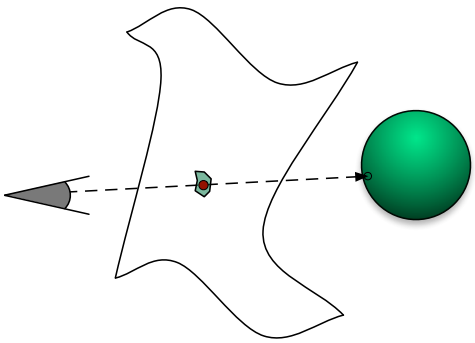



$$P = u\,(vLL + (1-v)UL) + (1-u)(vLR + (1-v)UR)$$

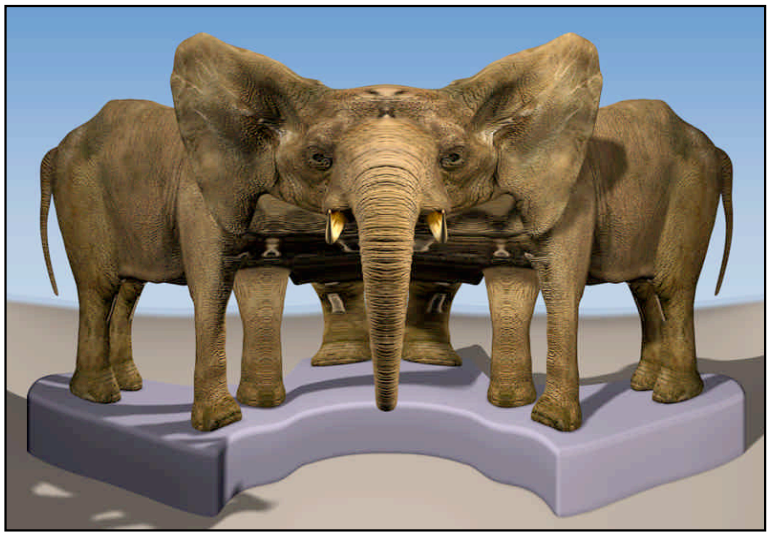


# Building Eye Rays

- Nonlinear projections
  - Non-planar projection surface
  - Variable eye location



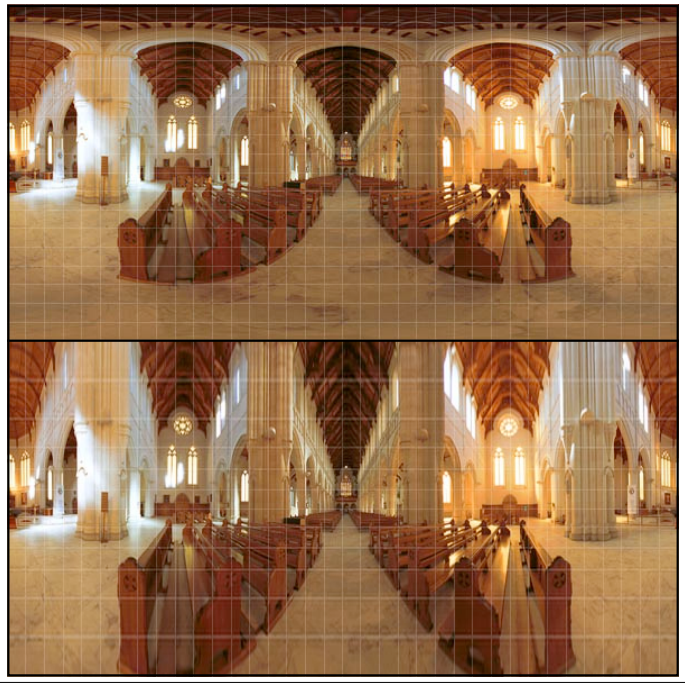
# Examples



***Multiple-Center-of-Projection Images***
P. Rademacher and G. Bishop
SIGGRAPH 1998



# Examples



***Spherical and Cylindrical Projections***
Ben Kreunen
From *Big Ben's Panorama Tutorials*

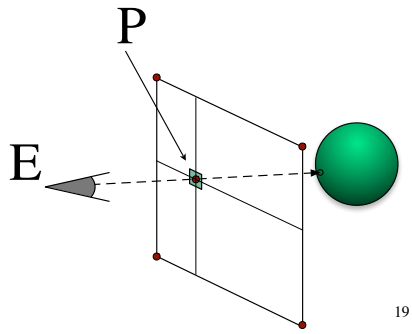
# Building Eye Rays

- Ray equation

$$\mathrm{R}(t) = \mathrm{E} + t(\mathrm{P} - \mathrm{E})$$

$$t \in [1 \ldots + \infty]$$

  - Through eye at $t = 0$
  - At pixel center at $t = 1$



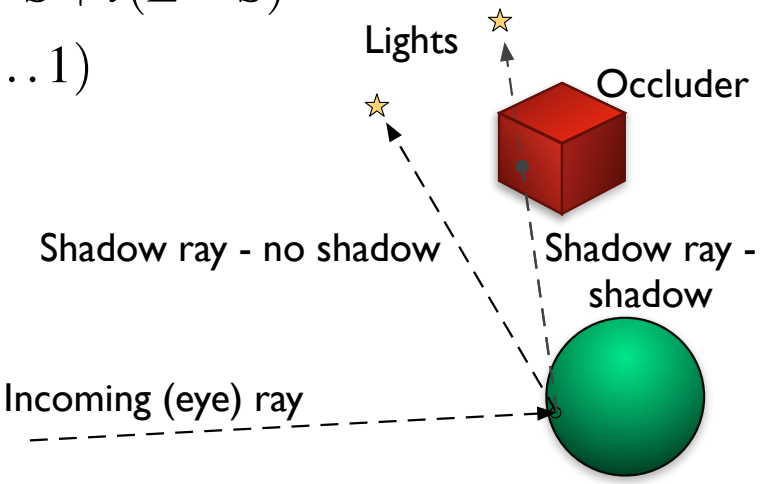



# Shadow Rays

- Detect shadow by rays to light source

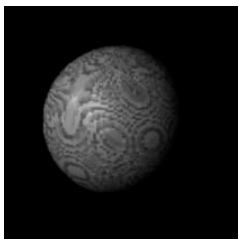
$$\mathrm{R}(t) = \mathrm{S} + t(\mathrm{L} - \mathrm{S})$$

$$t \in [\varepsilon \ldots 1)$$

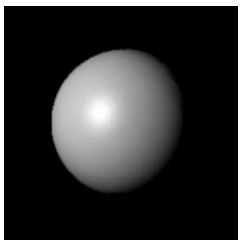


# Shadow Rays

- Test for occluder
  - No occluder, shade normally ( e.g. Phong model )
  - Yes occluder, skip light ( don't skip ambient )
- Self shadowing
  - Add shadow bias
  - Test object ID



Self-shadowing



Correct

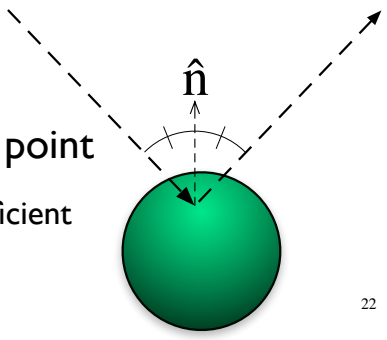


# Reflection Rays

- Recursive shading
  - Ray bounces off object
  - Treat bounce rays (mostly) like eye rays
  - Shade bounce ray and return color
    - Shadow rays
    - Recursive reflections
  - Add color to shading at original point
    - Specular or separate reflection coefficient

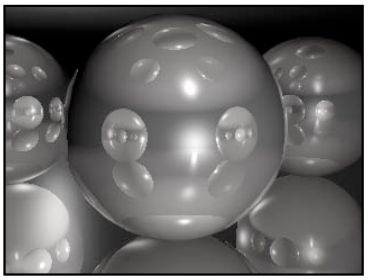
$$\mathrm{R}(t) = \mathrm{S} + t\,\mathrm{B}$$

$$t \in [\varepsilon \ldots + \infty)$$



# Reflection Rays

- Recursion Depth
  - Truncate at fixed number of bounces
  - Multiplier less than J.N.D.

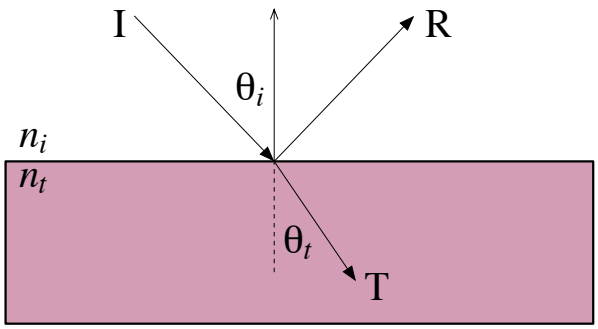




# Refracted Rays

- Transparent materials bend light
  - Snell's Law $\frac{n_i}{n_t} = \frac{\sin\theta_t}{\sin\theta_i}$ ( see clever formula in text... )

$\sin\theta_t > 1 \Rightarrow$ Total (internal) reflection



# Refracted Rays

- Coefficient on transmitted ray depends on $\theta$
  - Schlick approximation to Fresnel Equations

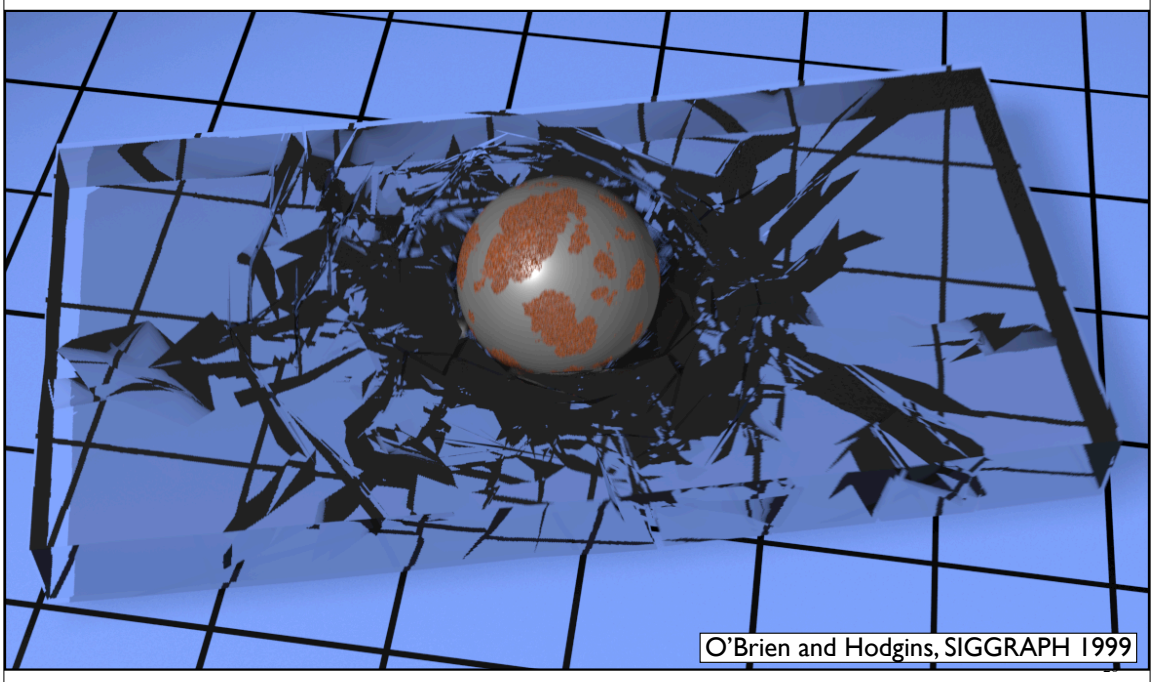
$$k_t(\theta_i) = k_0 + (1 - k_0)(1 - \cos\theta_i)^5$$

$$k_0 = \left(\frac{n_t - 1}{n_t + 1}\right)^2$$

  - Attenuation
    - Wavelength (color) dependant
    - Exponential with distance

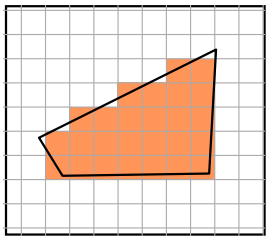


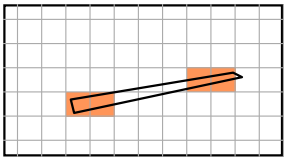
# Refracted Rays



# Anti-Aliasing

- Boolean on/off for pixels causes problems
  - Consider scan conversion algorithm:



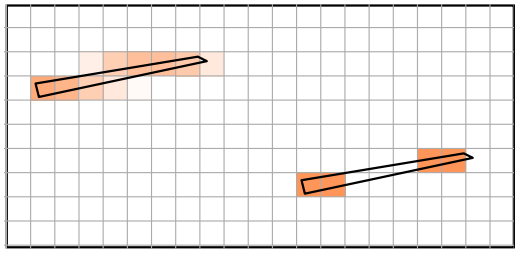


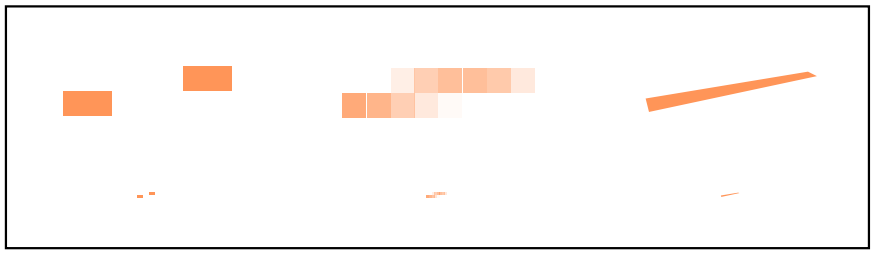
  - Compare to casting a ray through each pixel center
- Recall Nyquist Theorem
  - *Sampling rate ≥ twice highest frequency*



# Anti-Aliasing

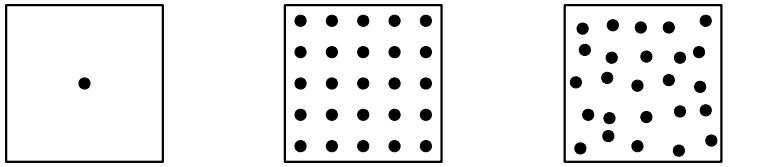
- Desired solution of an integral over pixel





# "Distributed" Raytracing

- Send multiple rays through each pixel



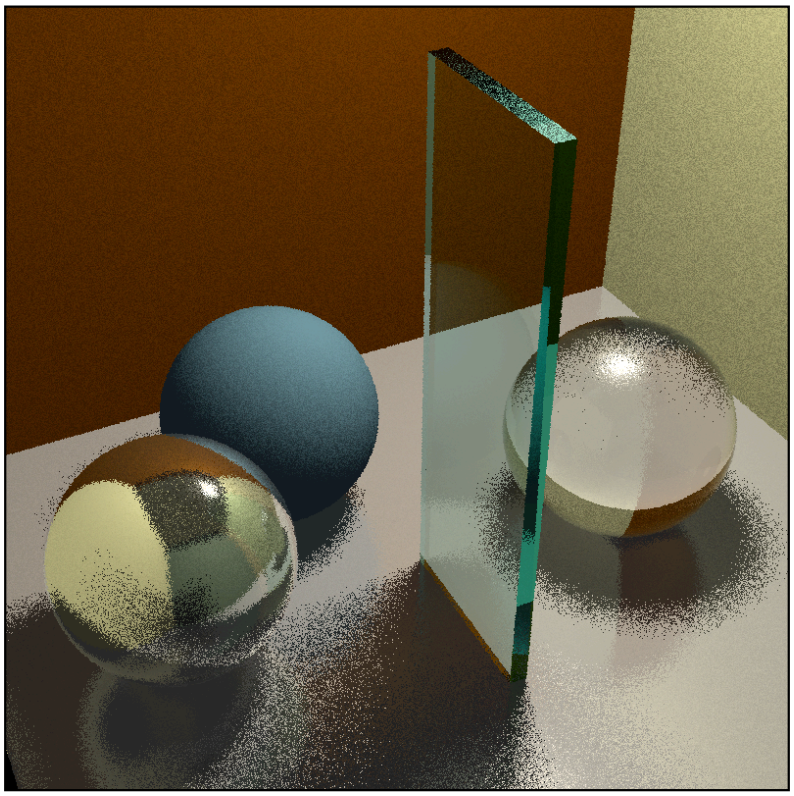
One Sample  5x5 Grid  5x5 Jittered Grid

- Average results together
- Jittering trades aliasing for noise



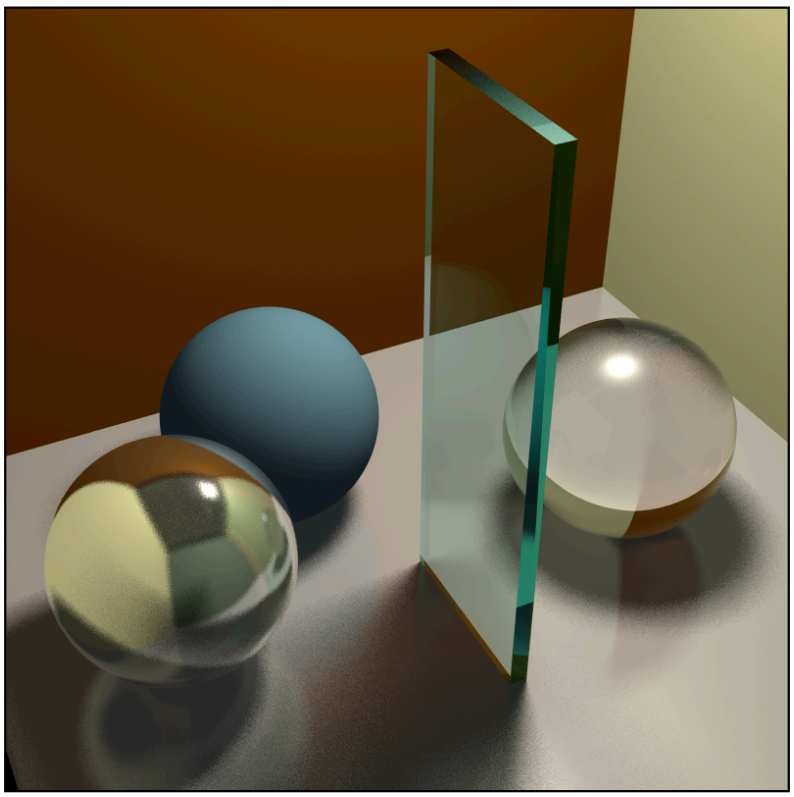
# "Distributed" Raytracing

- Use multiple rays for reflection and refraction
  - At each bounce send out many extra rays
  - Quasi-random directions
  - Use BRDF (or Phong approximation) for weights
- How many rays?

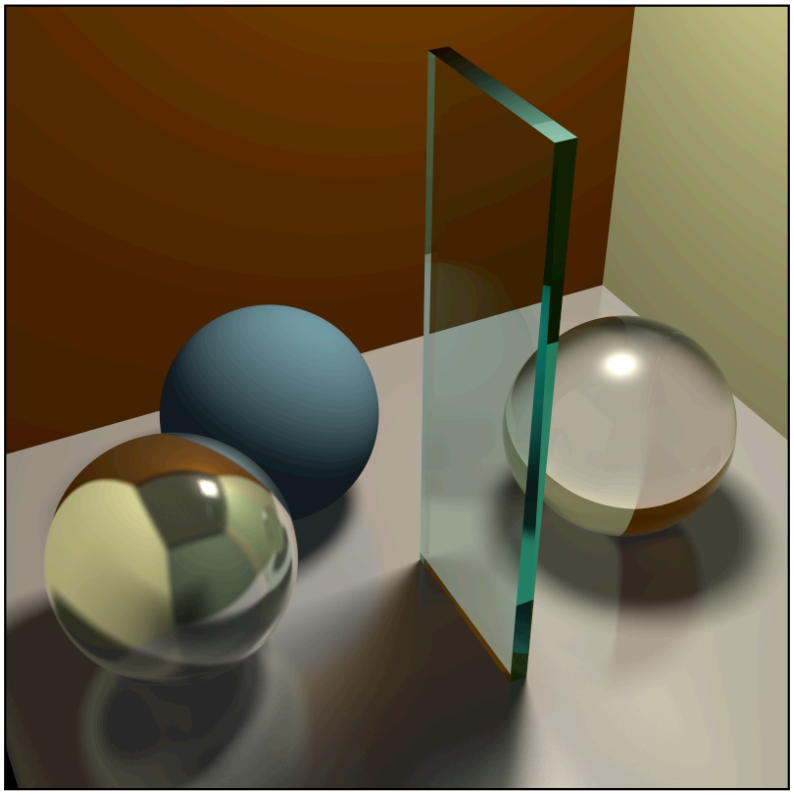


1





16

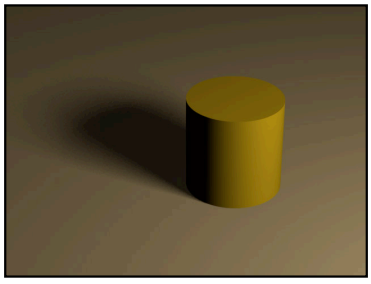


256



# Soft Shadows

- Soft shadows result from non-point lights
  - Some part of light visible, some other part occluded



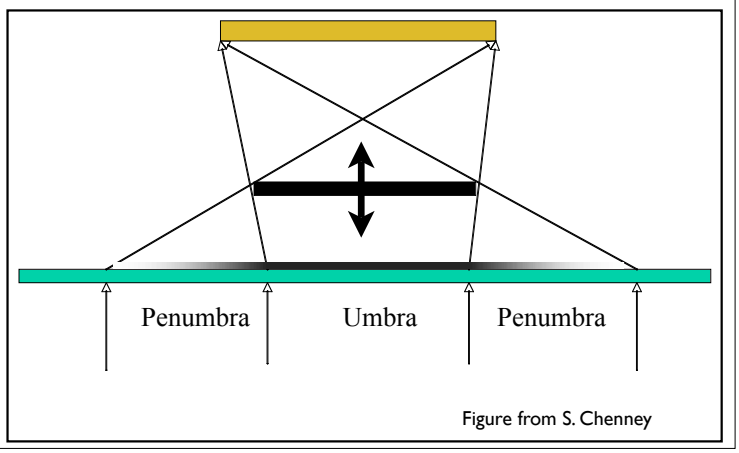
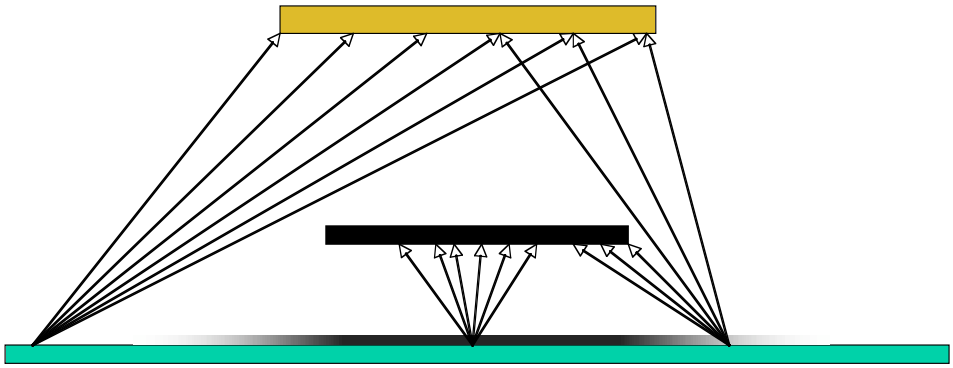



Figure from S. Chenney

# Soft Shadows

- Distribute shadow rays over light surface



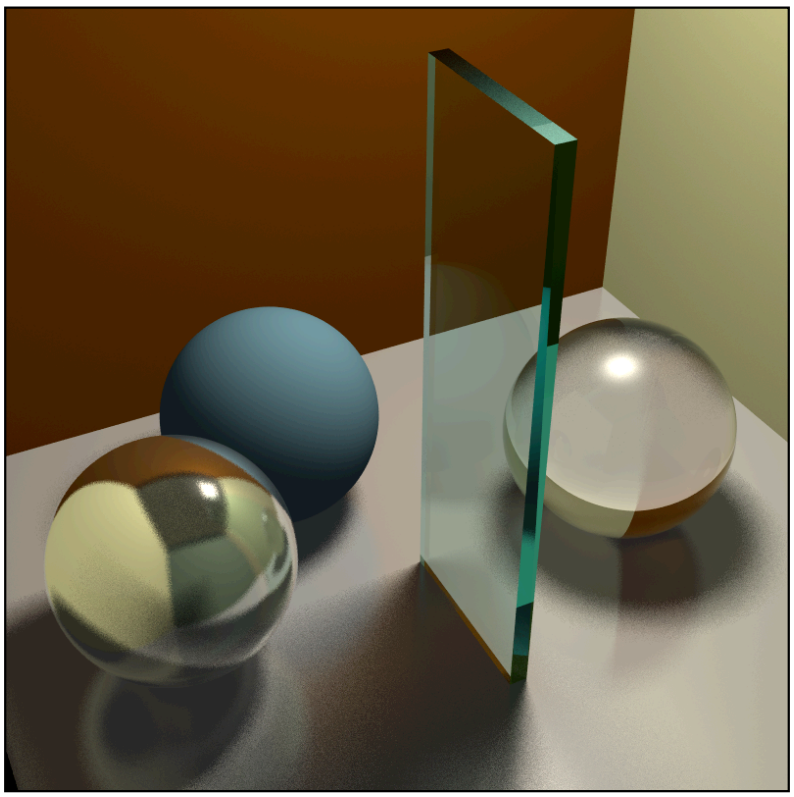
All shadow rays go through

No shadow rays go through

Some shadow rays go through

Figure from S. Chenney





16

# Motion Blur

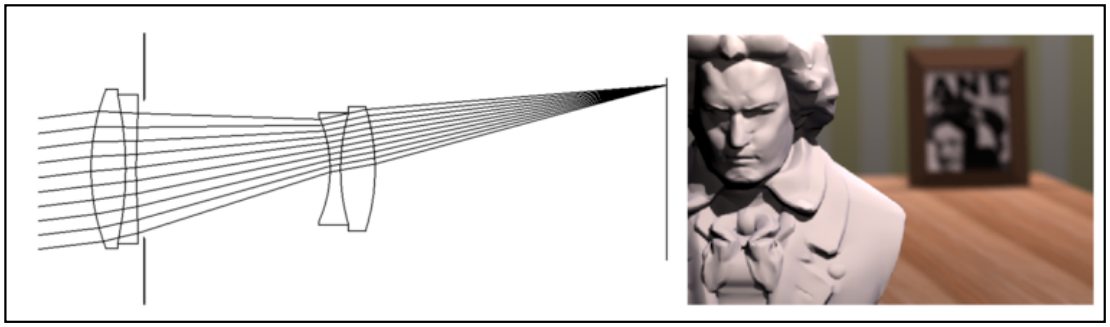
- Distribute rays over *time*
  - More when we talk about animation...

**Pool Balls**
Tom Porter
RenderMan



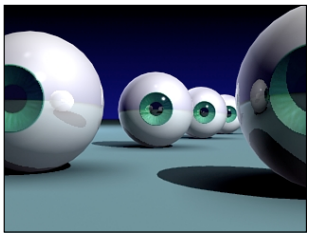
# Depth of Field

- Distribute rays over a lens assembly

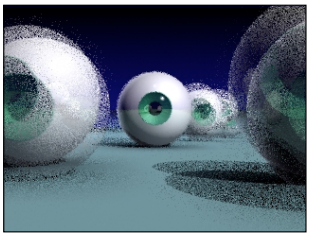


Kolb, Mitchell, and Hanrahan
SIGGRAPH 1995

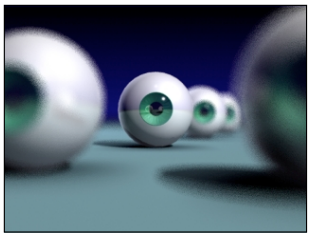
# Depth of Field



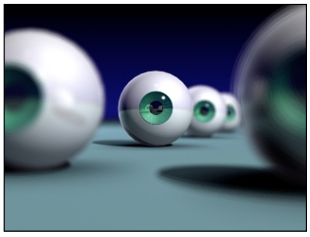
No DoF



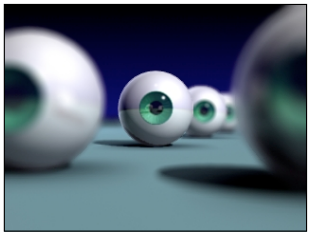
Jittered rays for DoF



More rays



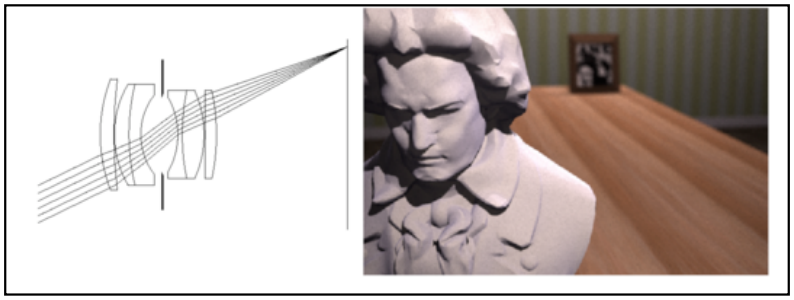
Multiple images for DoF

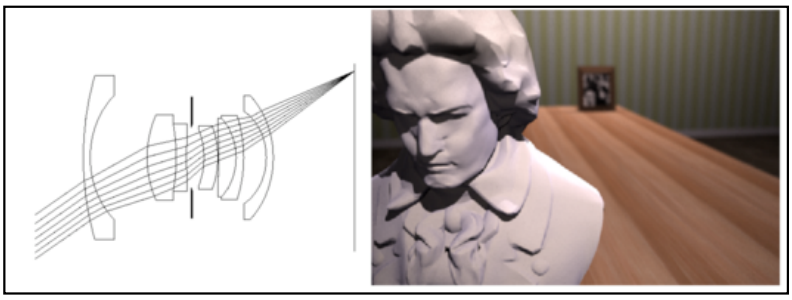


Even more rays



# Other Lens Effects





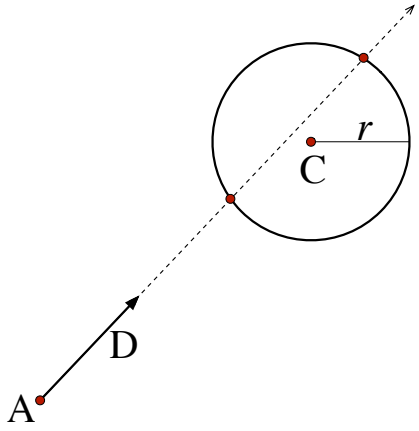
Kolb, Mitchell, and Hanrahan
SIGGRAPH 1995

# Ray -vs- Sphere Test

- Ray equation: $\mathrm{R}(t) = \mathrm{A} + t\,\mathrm{D}$
- Implicit equation for sphere: $|\mathrm{X} - \mathrm{C}|^2 - r^2 = 0$
- Combine:

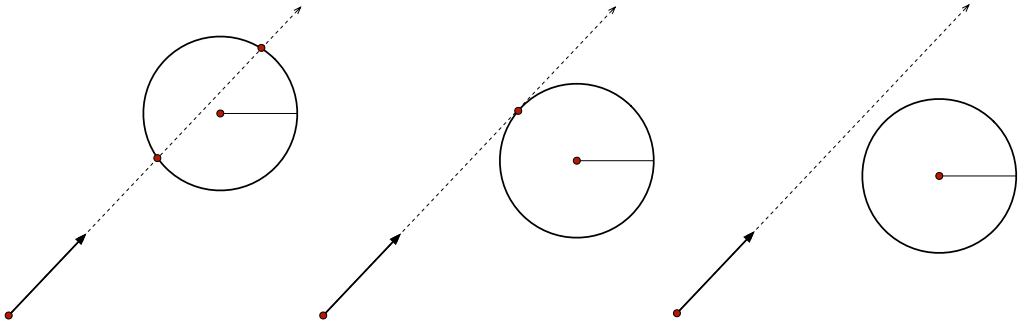
$$|\mathrm{R}(t) - \mathrm{C}|^2 - r^2 = 0$$
$$|\mathrm{A} + t\,\mathrm{D} - \mathrm{C}|^2 - r^2 = 0$$

- Quadratic equation in $t$






# Ray -vs- Sphere Test



Two solutions One solution Imaginary

# Ray -vs- Triangle

- Ray equation: $\mathrm{R}(t) = \mathrm{A} + t\,\mathrm{D}$
- Triangle in barycentric coordinates:
  $$\mathrm{X}(\beta,\gamma) = \mathrm{V}_1 + \beta(\mathrm{V}_2 - \mathrm{V}_1) + \gamma(\mathrm{V}_3 - \mathrm{V}_1)$$
- Combine:
  $$\mathrm{V}_1 + \beta(\mathrm{V}_2 - \mathrm{V}_1) + \gamma(\mathrm{V}_3 - \mathrm{V}_1) = \mathrm{A} + t\,\mathrm{D}$$
- Solve for $\beta$, $\gamma$, and $t$
  - 3 equations 3 unknowns
  - Beware divide by near-zero
  - Check ranges