Global Illumination

Can you get this with ray tracing?

Rendering Equation

$$I(x,x') = g(x,x')[\varepsilon(x,x') + \int_s \rho(x,x',x'')I(x',x'')dx'']$$

- g() is the "visibility" function
- $\rho()$ is related to BRDF:

$$\rho(x,x',x'') = \rho(\theta'_{in},\phi'_{in},\theta'_{ref},\phi'_{ref})\cos\theta\cos\theta'_{ref}$$

From Watt's p.277

How to Solve It?

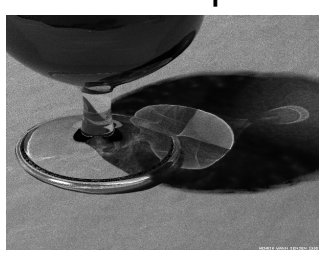
- We must have:
  - $\varepsilon()$: model of the light emitted
  - $\rho()$: BRDF for each surface
  - g(): method to evaluate visibility
- Integral evaluation →Monte Carlo
- Recursive equation → Ray Tracing
- The problem is view independent

Ray Tracing Revisited

- The reflected intensity (or color) at a surface point is computed by:
  - Local reflection model (no interaction with other objects): ambient, diffuse, and specular.
  - Global model: perfect reflection and refraction.
- What if we spawn many reflected rays?

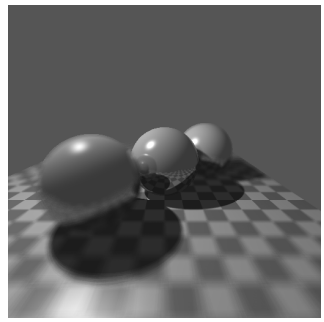
Global Illumination Algorithms

- Radiosity (topic of the next lecture).
- Distributed Ray Tracing.
- RADIANCE
- Photon Map



Distributed Ray Tracing

- Distribute a group of rays at a hit point to sample the “reflection lobe” (similar to a 2D slice of BRDF).
- May also distribute rays along camera aperture, time, and pixel region to produce effects of depth of fields, motion blur, and anti-aliasing.



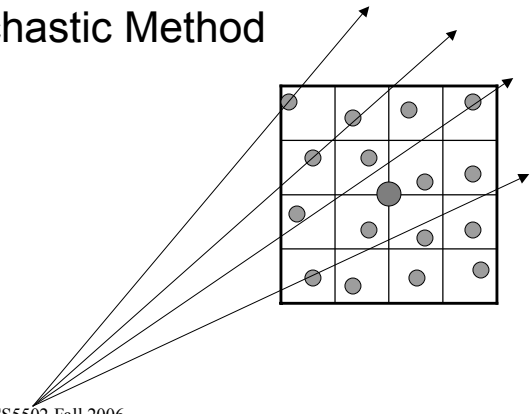
Why Distributed Ray Tracing?

- Anti-Aliasing
- Features
  - Gloss (fuzzy reflections)
  - Fuzzy translucency
  - Penumbras (soft shadows)
  - Depth of field
  - Motion blur

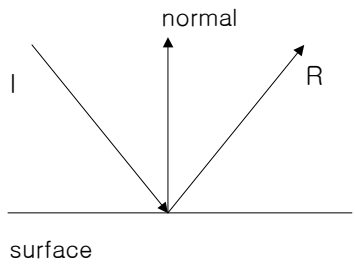
Anti-Aliasing

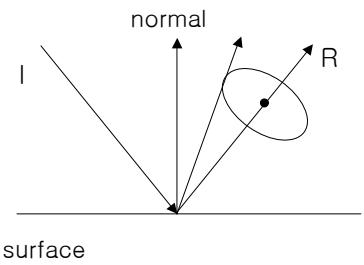
- Supersampling
- Jittering – Stochastic Method

6	10	2	13
3	14	12	8
15	0	7	11
5	9	4	1

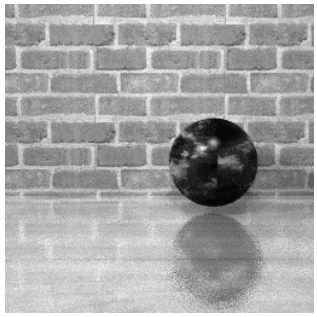


Gloss

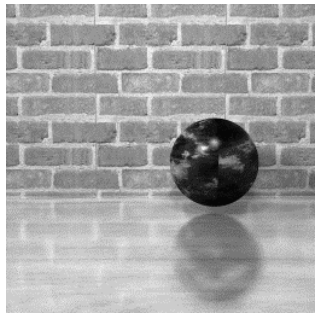




Fuzzy Reflection

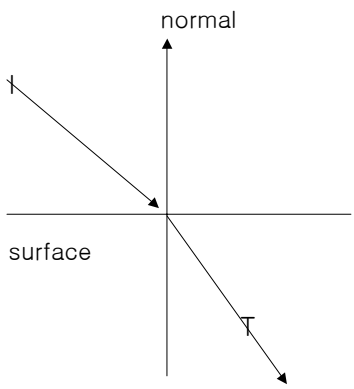


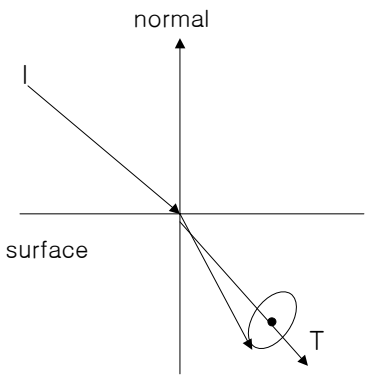
4 rays, 37 seconds

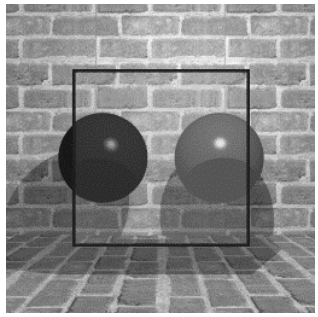


64 rays, 956 seconds

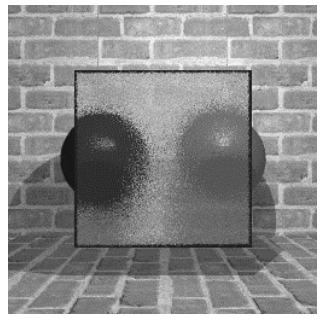
Translucent

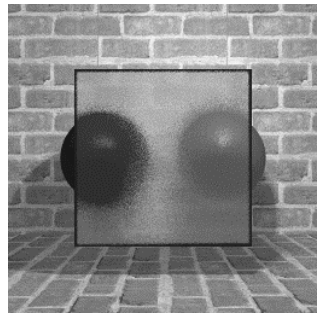






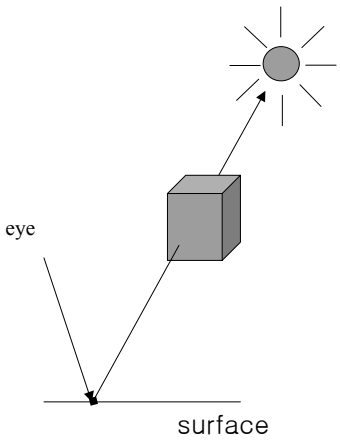
4 rays



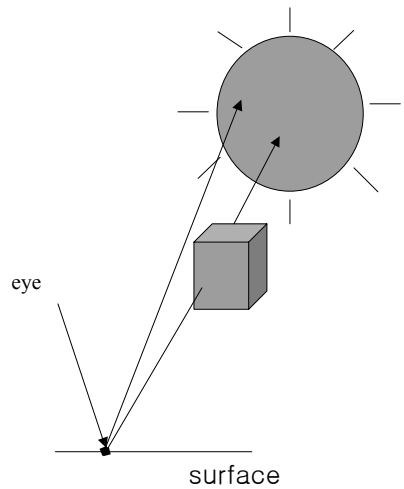


16 rays

Penumbra (Soft Shadow)

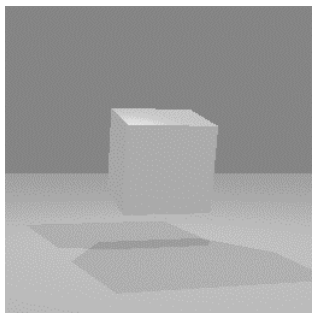


Hard Shadow

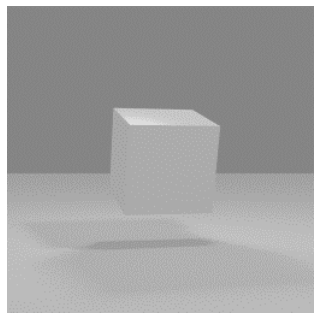


Soft Shadow

Soft shadow - cube

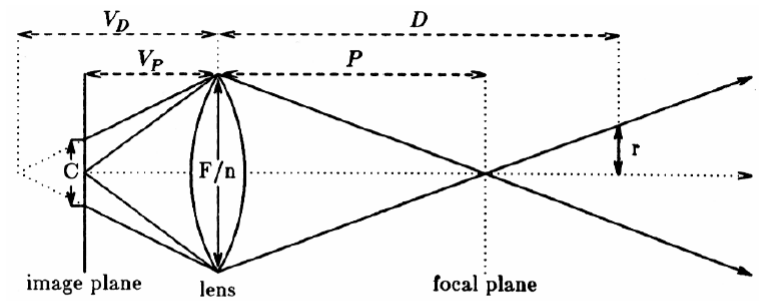


Without penumbra



With penumbra

Depth of Field

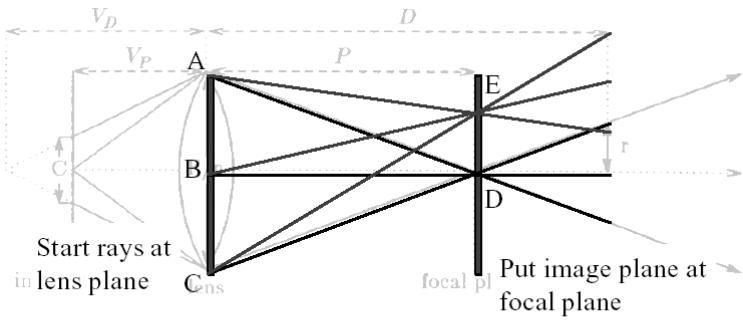


F – focal length
n – aperture number
C – circle of confusion

$V_P = FP/(P\text{-}F)$
$V_D = FD/(D\text{-}F)$
$C = (|V_D - V_P|/V_D)\,(\mathrm{F}/n)$

$r = \frac{1}{2}\,(F/n)\,(D\text{-}P)/P$
$R = (-V_P/D)\,r$
$R = \frac{1}{2}\,C$

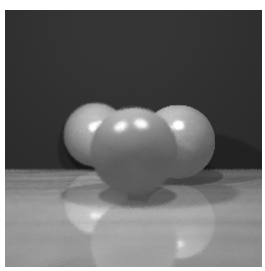
Depth of Field



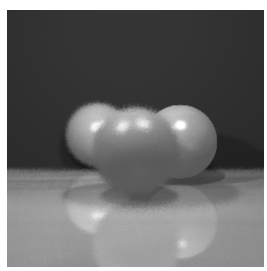
Standard ray tracing:
Pixel D uses ray BD
Pixel E uses ray BE
All rays emanate from B

Distributed ray tracing:
Pixel D uses rays AD, BD, CD
Pixel E uses rays AE, BE, CE
Rays emanate from lens plane

Depth of Field

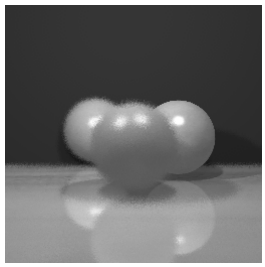


F-Stop = 5.8

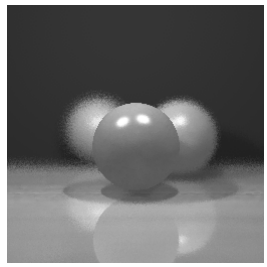


F-Stop = 2.8

Depth of Field



Focal Distance = 13



Focal Distance = 11

Motion Blur

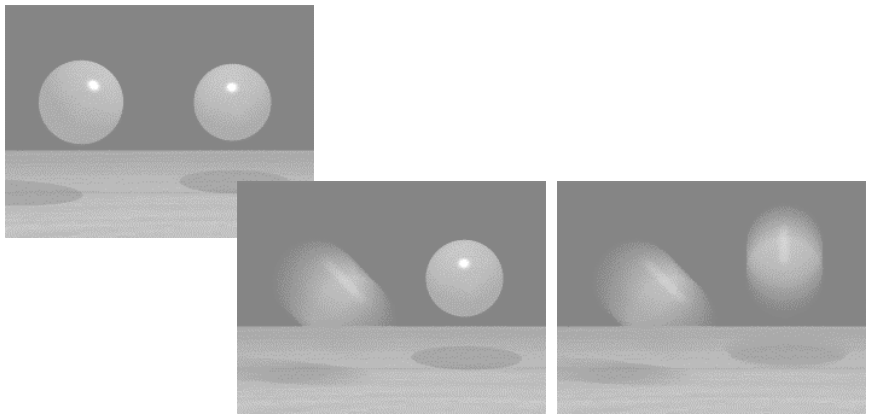
- Sampling in time
- Each element in the cell stands for a time slice
- Jitter time slice to the current time
- Move object via the current time slice

6	10	2	13
3	14	12	8
15	0	7	11
5	9	4	1

Current time = Time Slice + Jitter Time

e.g. time slice at left-upper = 6 + rand()

Motion Blur



Typical Distributed Ray Path