MIT 6.837
Monte-Carlo
Ray Tracing

Schedule

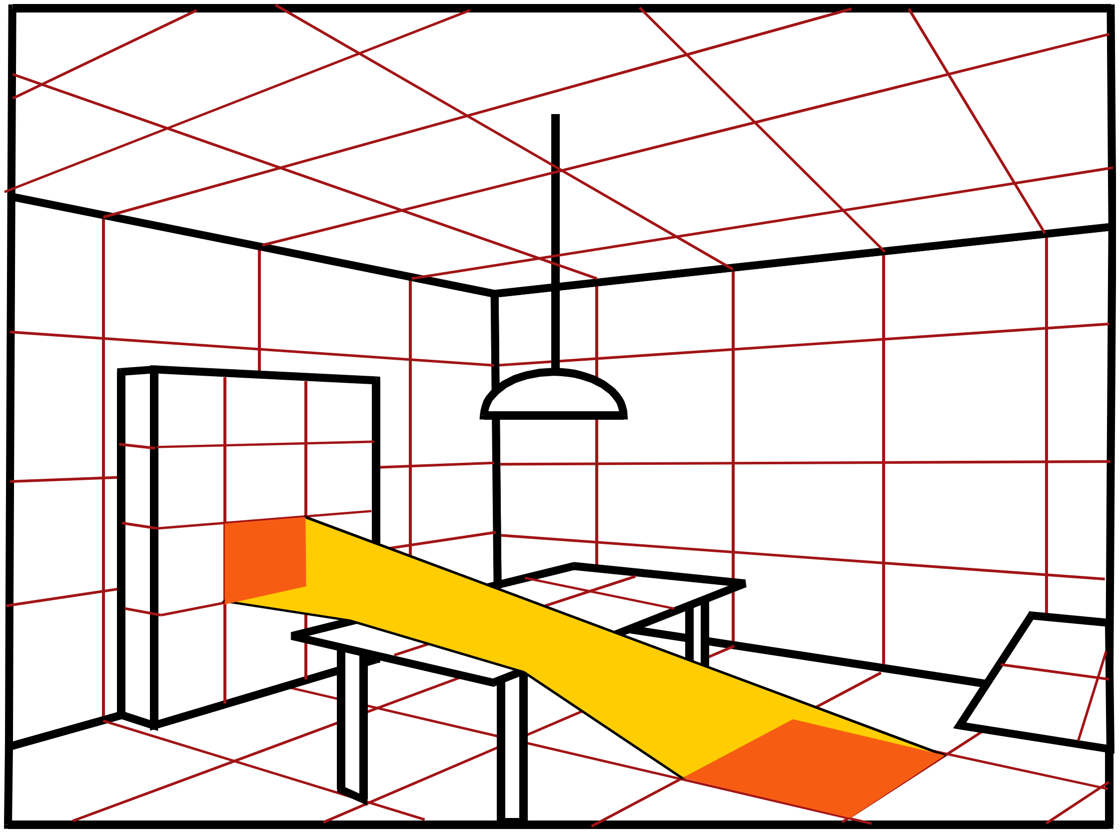
- Review Session:
  Tuesday November 18th, 7:30 pm
  bring lots of questions!

- Quiz 2: Thursday November 20th, in class
  (one weeks from today)

Review of last week?

Radiosity

- Diffuse surfaces
- Subdivide scene
- Radiosity assumed constant over a patch
- Form-factor between patches
- Geometry and visibility
- Big Matrix system



Radiosity

- Smoothing and other gimmicks

Limitations of radiosity

- Diffuse only for basic method
  - Costly extension to specular
- Requires meshing
- Cost of visibility computation
  - If you send rays, why not use ray tracing?
- Memory consumption vs. time scalability

Why still learn radiosity?

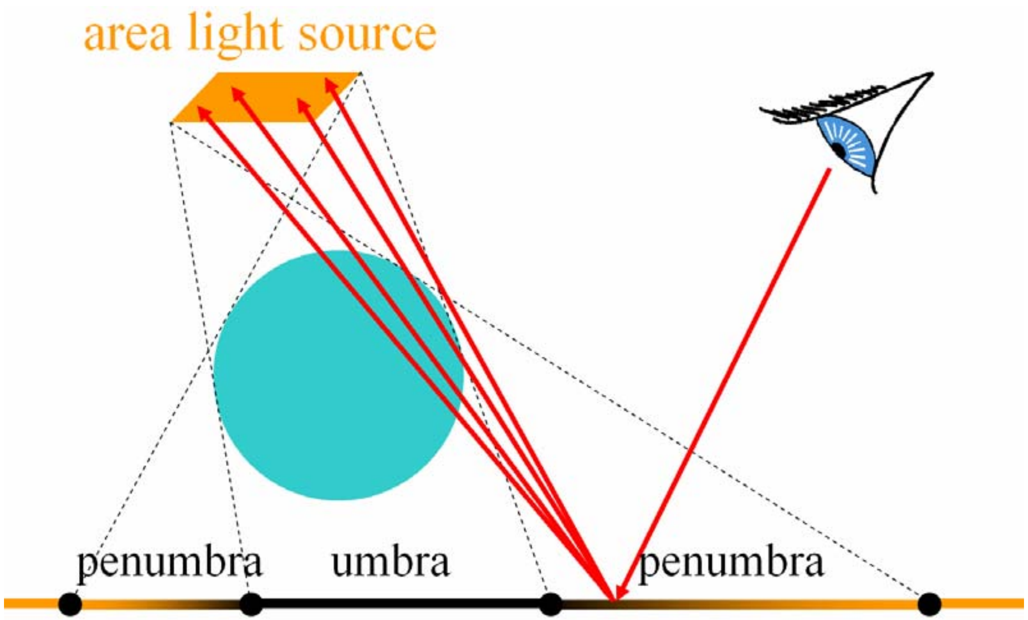
- Still used in architecture (Lightscape)
- Introduction to finite element method
  - Project the problem onto a finite basis of functions
    - In the case of radiosity: piecewise constant
  - Express interaction between elements
  - Get a big matrix system
  - Same as deformable object simulation
- Pre-computed radiance transfer: same strategy
  - Use finite basis function
  - Precompute lighting as a function of primary sources
  - Use in a simplified version in Max Payne 2

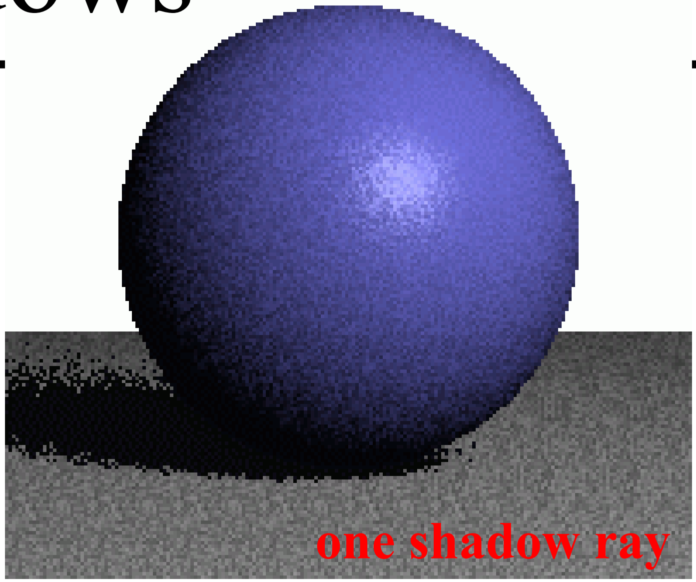
Today: Monte Carlo Ray Tracing

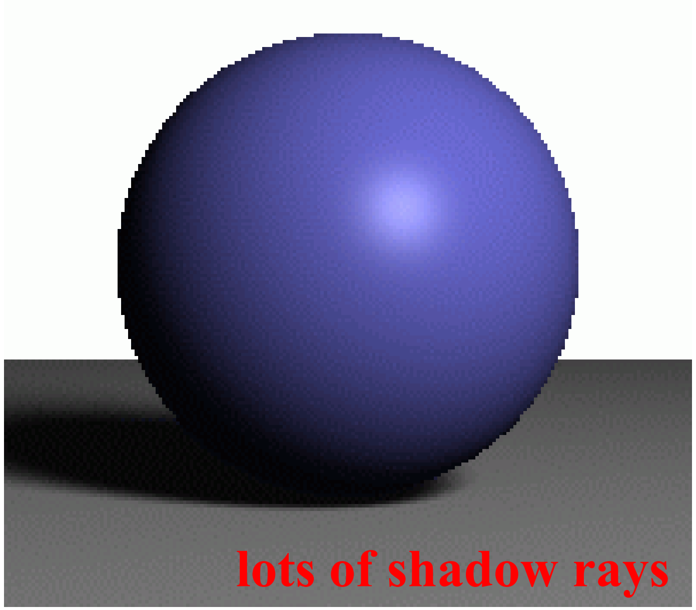
- Principle of Monte-Carlo Ray Tracing
- Monte Carlo integration
- Review of Rendering equation
- Advanced Monte Carlo Ray Tracing

Probabilistic Soft Shadows

- Multiple shadow rays to sample area light source
- Monte-Carlo ray tracing generalizes this

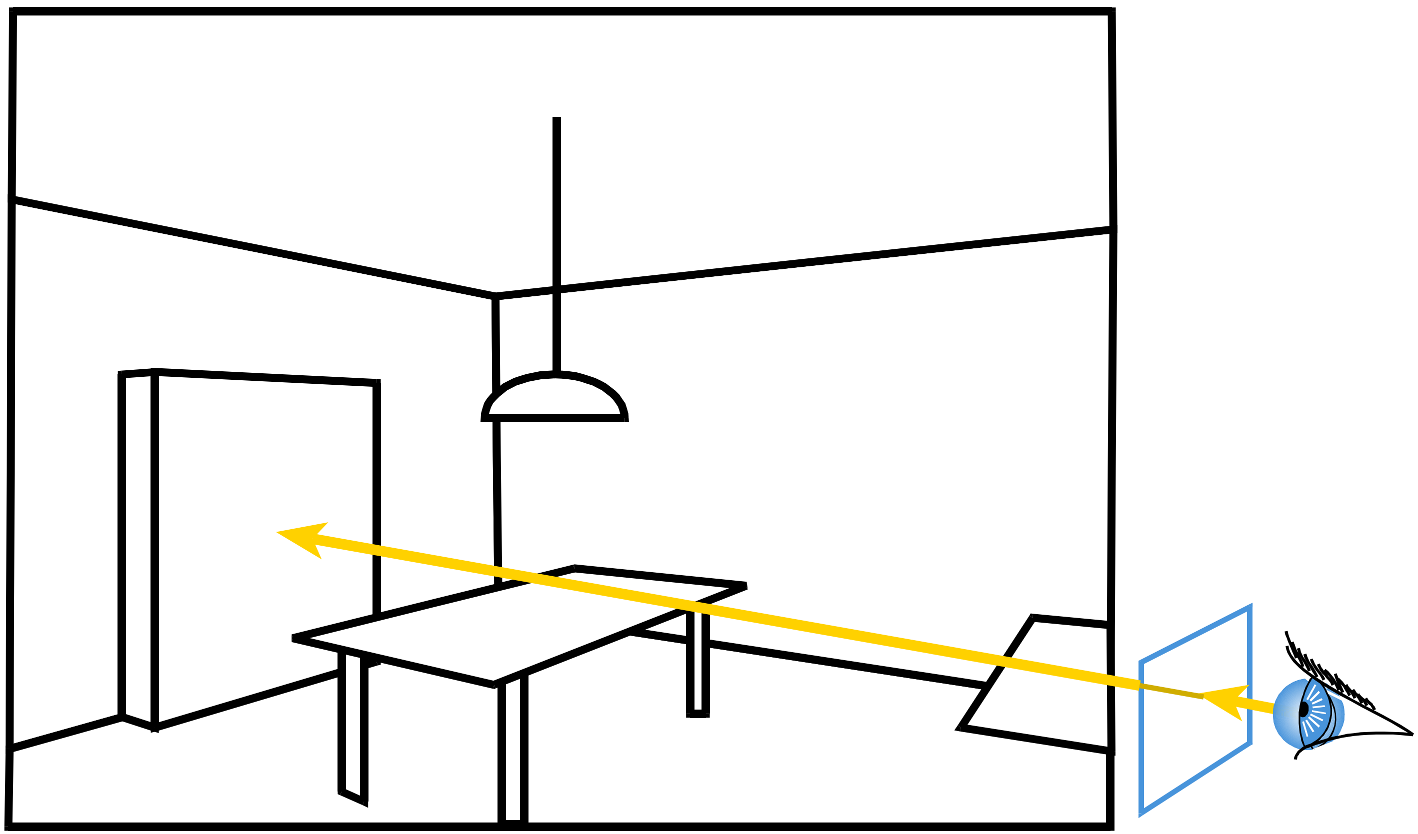






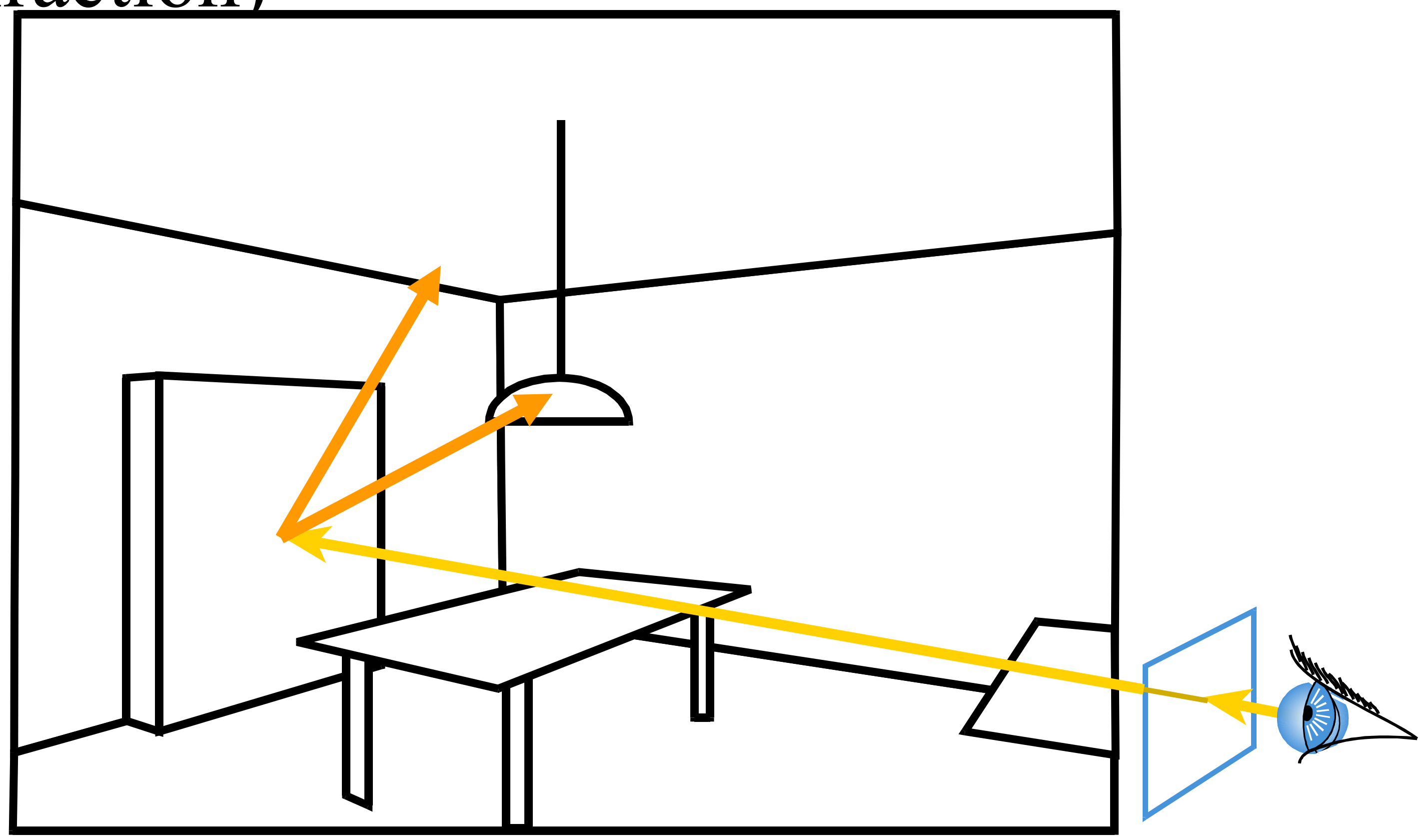
Ray Casting

- Cast a ray from the eye through each pixel



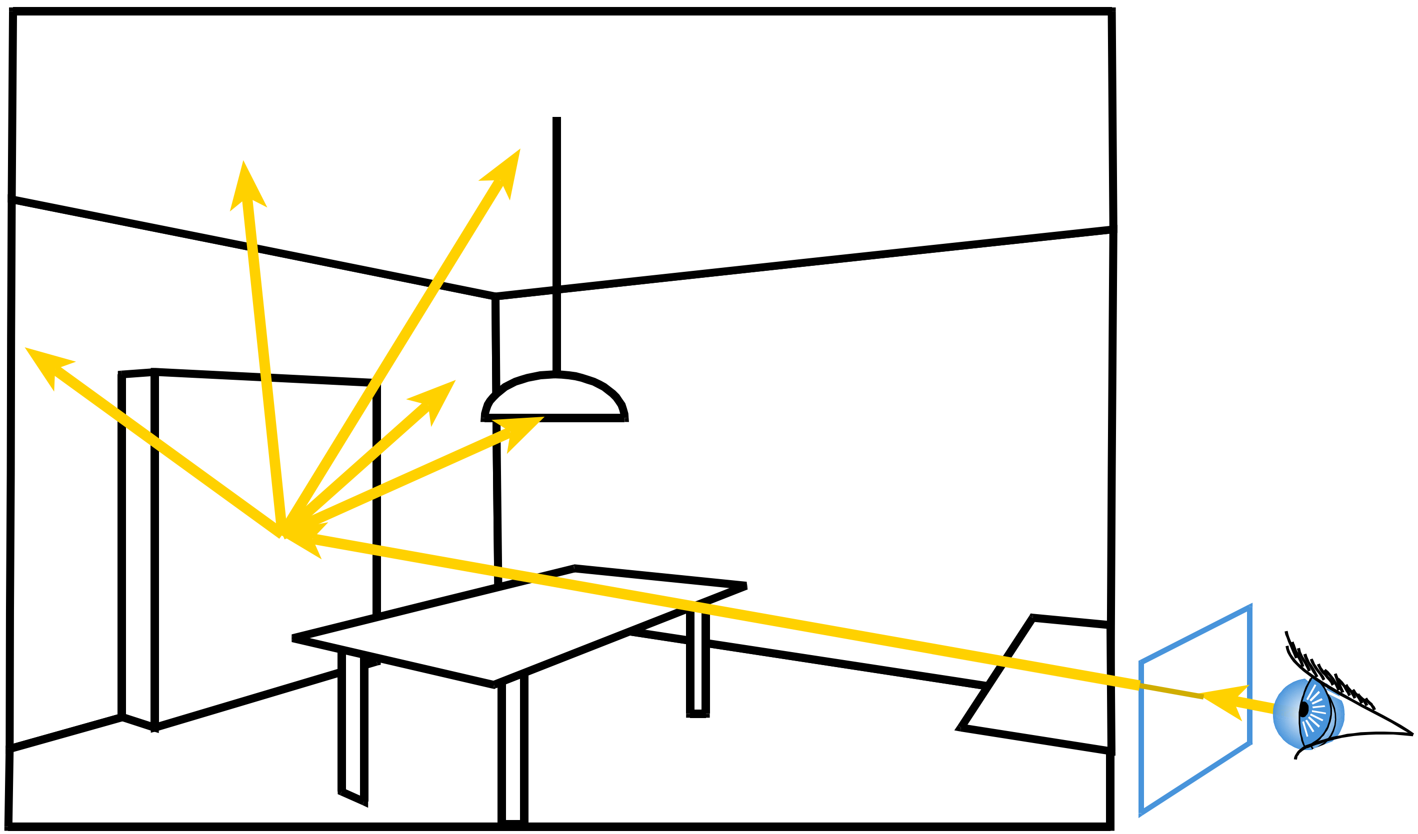
Ray Tracing

- Cast a ray from the eye through each pixel
- Trace secondary rays (light, reflection, refraction)



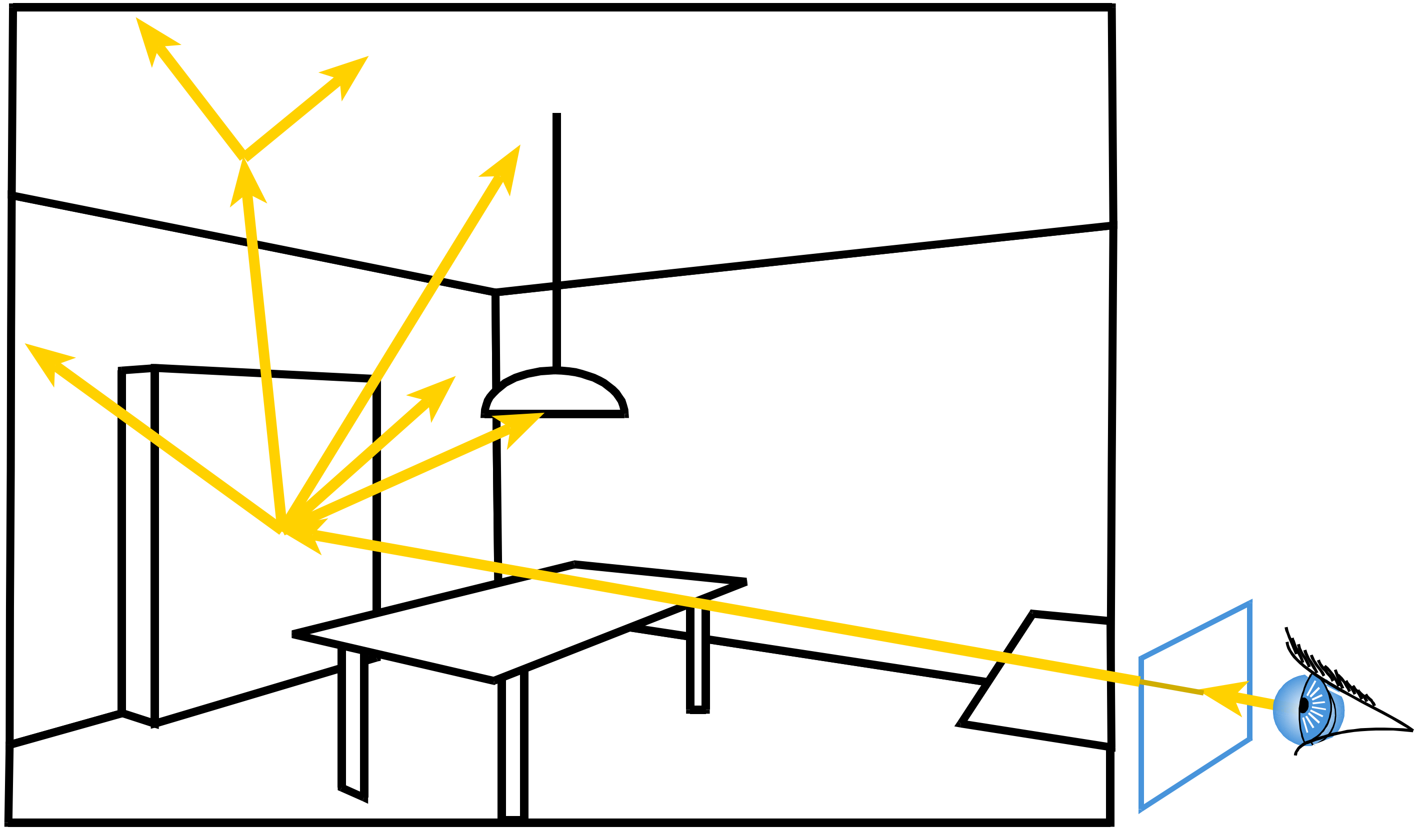
Monte-Carlo Ray Tracing

- Cast a ray from the eye through each pixel
- Cast random rays from the visible point
  - Accumulate radiance contribution



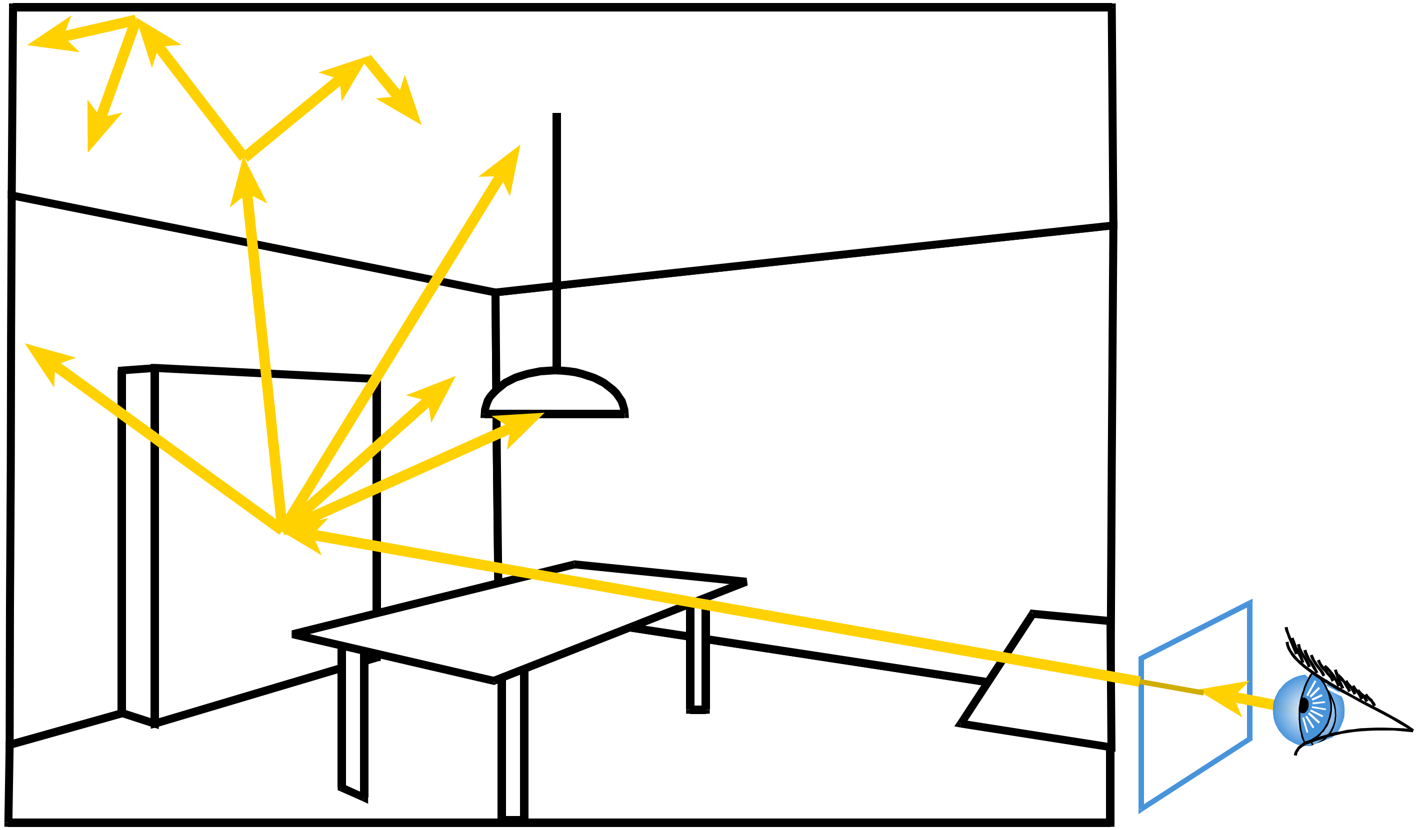
Monte-Carlo Ray Tracing

- Cast a ray from the eye through each pixel
- Cast random rays from the visible point
- Recurse



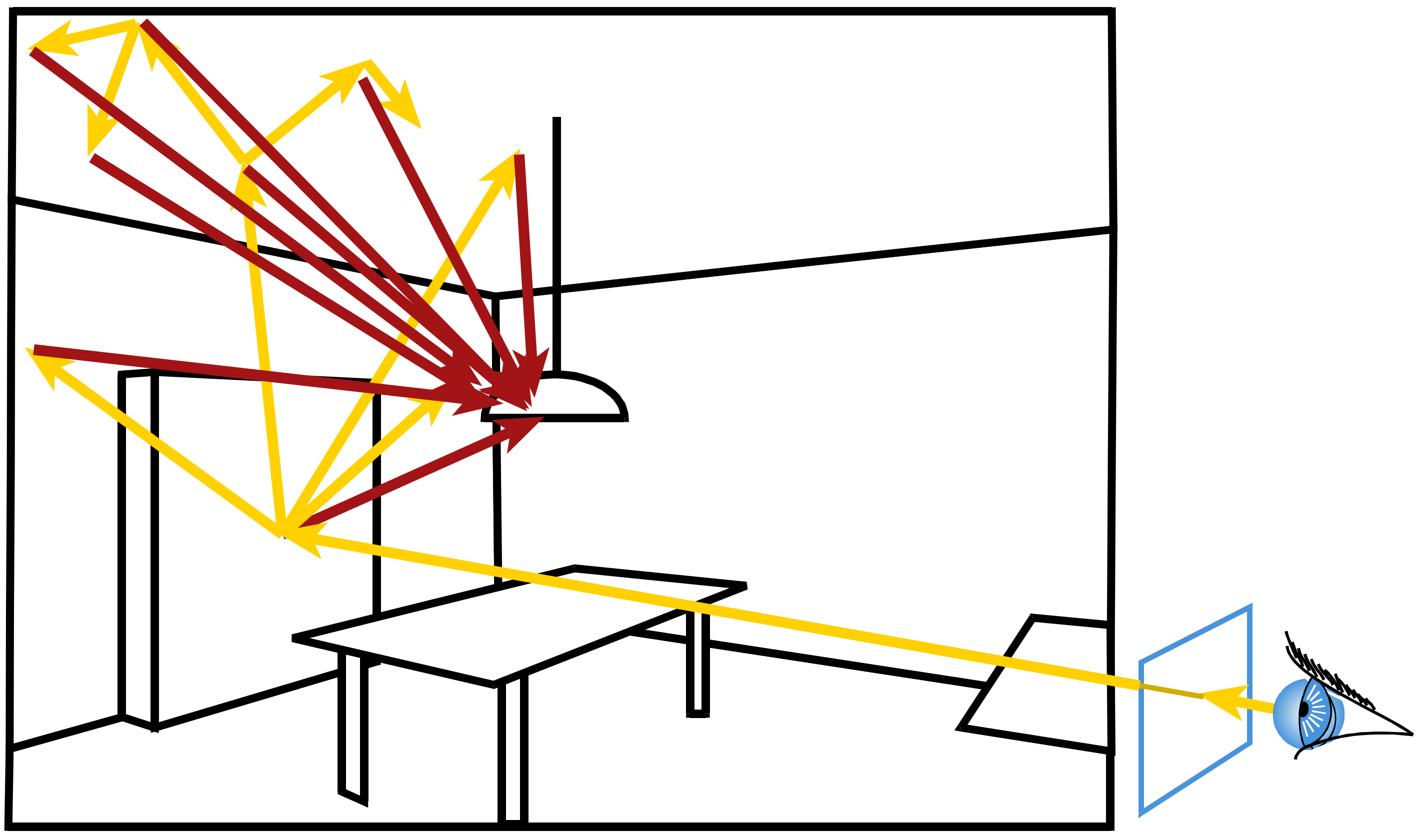
Monte-Carlo

- Cast a ray from the eye through each pixel
- Cast random rays from the visible point
- Recurse



Monte-Carlo

- Systematically sample primary light

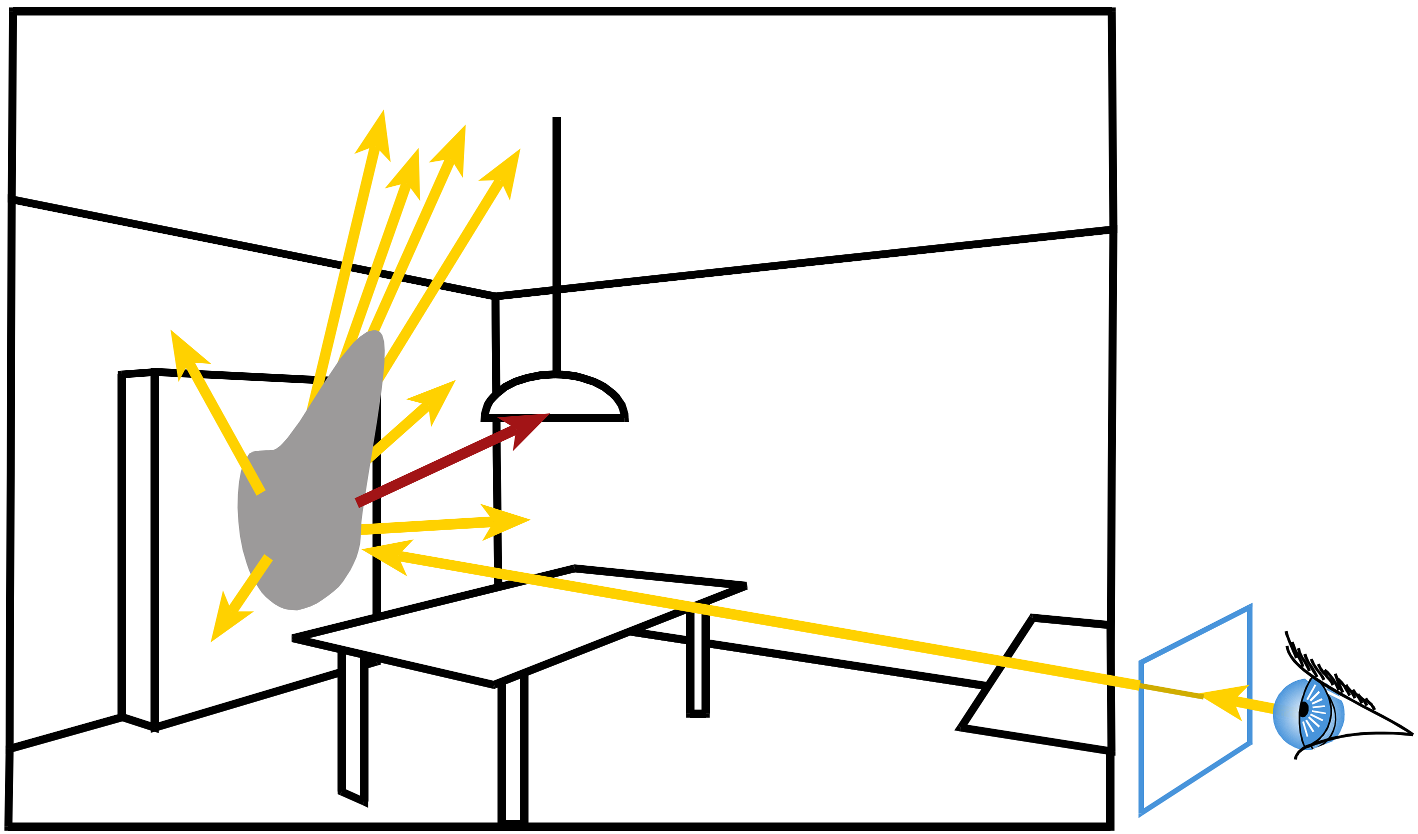


Results

(Image removed due to copyright considerations.)

Monte-Carlo

- Take reflectance into account
  - Multiply incoming radiance by BRDF value



- 1 sample per pixel

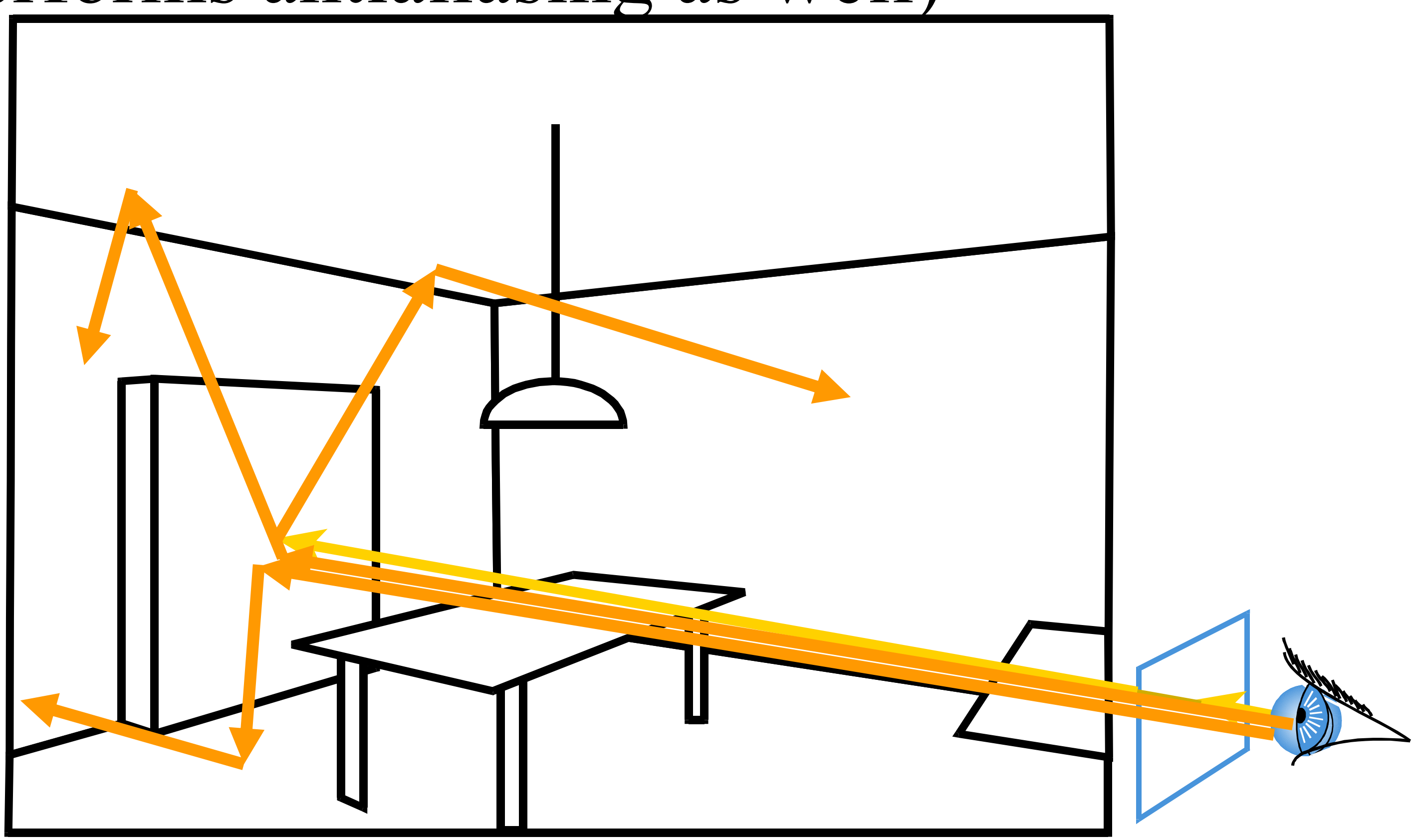
(Image removed due to copyright considerations.)

- 256 samples per pixel

(Image removed due to copyright considerations.)

Monte Carlo Path Tracing

- Trace only one secondary ray per recursion
- But send many primary rays per pixel
- (performs antialiasing as well)



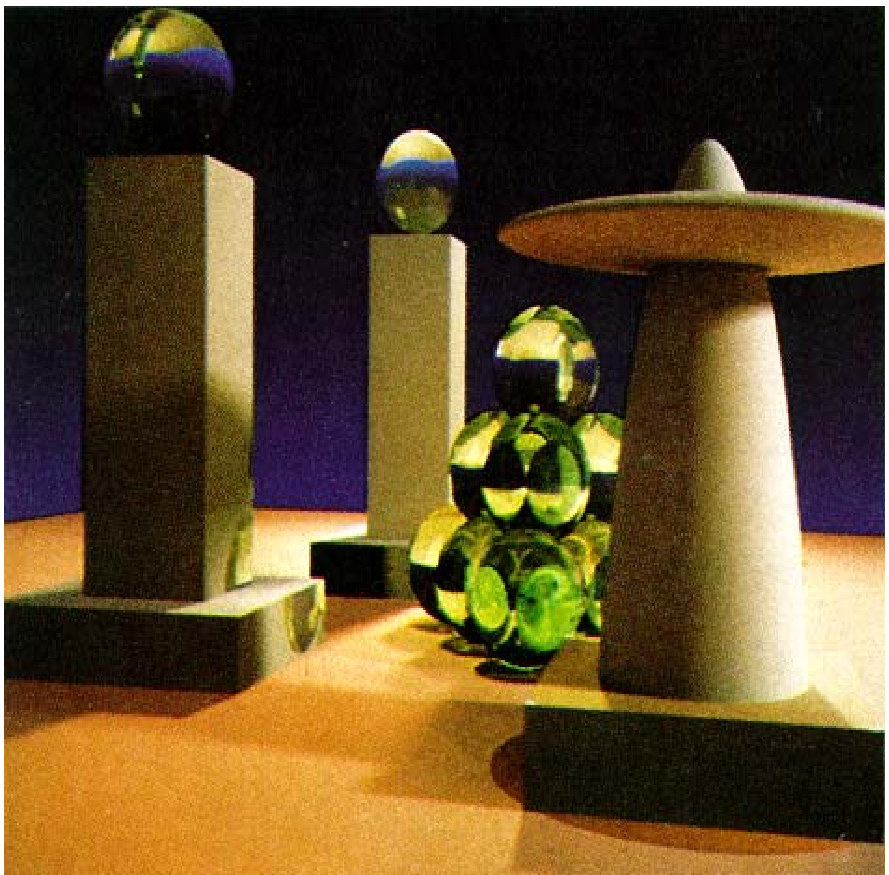
Monte Carlo Path tracing

traceRay
- Intersect all objects
- Shade visible object

Shade visible object
- Trace random ray to sample incoming light
- Shade using brdf

Vintage path tracing image

- by Jim Kajiya (1986)
- Also introduced the rendering equation



Convergence speed

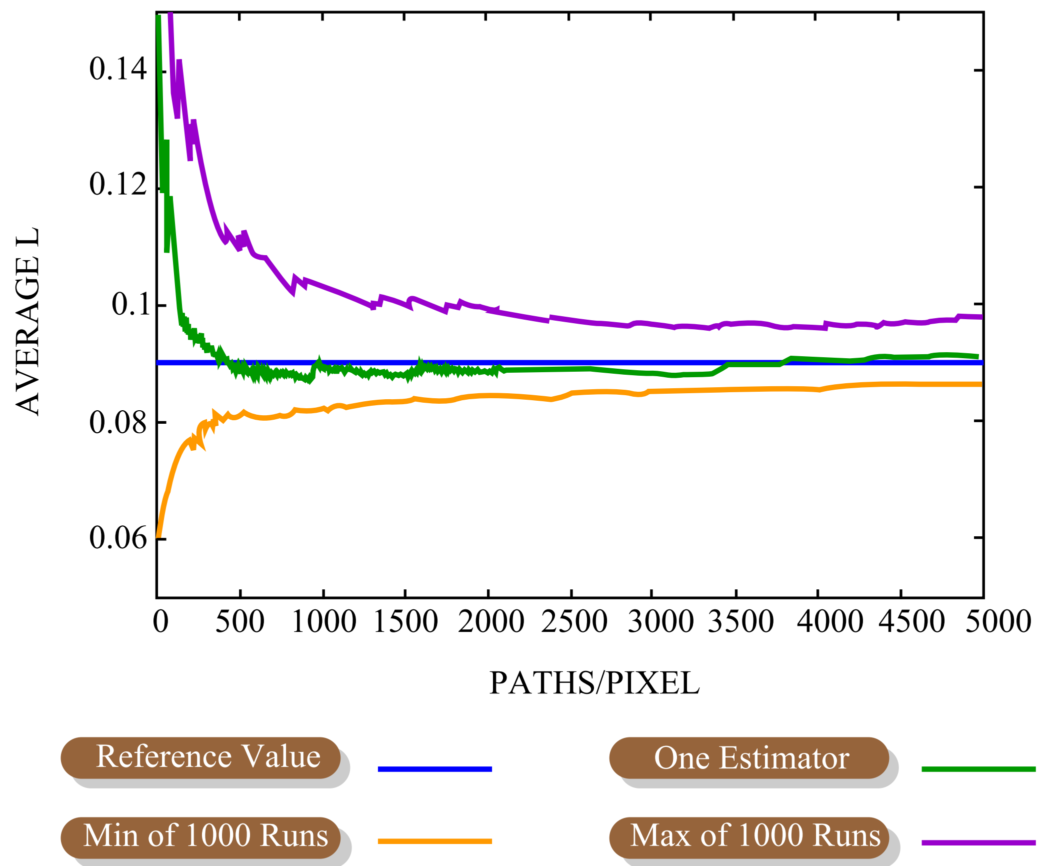


Image adapted from:
Henrik Wann Jensen. *Realistic Image Synthesis Using Photon Mapping.* 1st Ed.
A K Peters, Ltd. July 2001. ISBN: 1568811470.

Results

- 10 paths/pixel

(Image removed due to copyright considerations.)

Results

- 10 paths/pixel

(Image removed due to copyright considerations.)

Results

- 100 paths/pixel

(Image removed due to copyright considerations.)

Why use random numbers?

- Fixed random sequence
- We see the structure in the error

(Image removed due to copyright considerations.)

Recap

- Send random rays
- Sample the rendering equation
- No meshing required, no special storage
- No limitation
  - On reflectance
  - On geometry

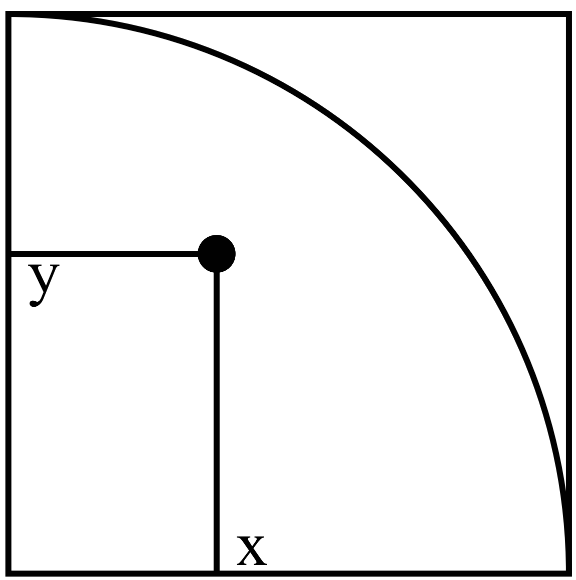
- Extremely flexible
- Can be noisy (or slow)

Today: Monte Carlo Ray Tracing

- Principle of Monte-Carlo Ray Tracing
- Monte Carlo integration
- Review of Rendering equation
- Advanced Monte Carlo Ray Tracing

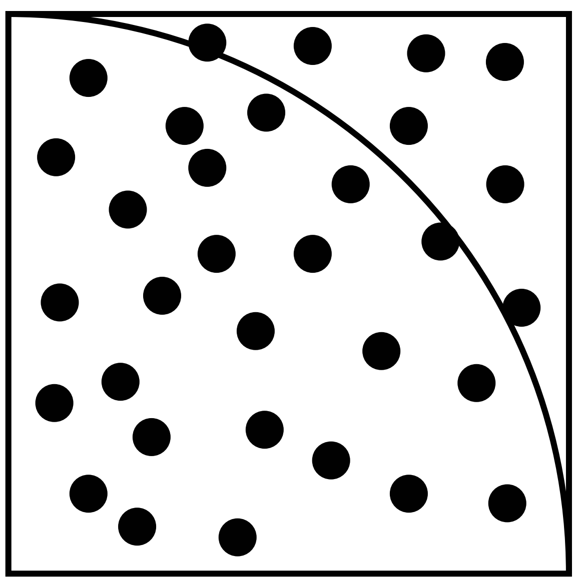
Monte-Carlo computation of π

- Take a square
- Take a random point (x,y) in the square
- Test if it is inside the ¼ disc ($x^2+y^2 < 1$)
- The probability is $\pi /4$



Monte-Carlo computation of π

- The probability is π /4
- Count the inside ratio n = # inside / total # trials
- $\pi \approx n * 4$
- The error depends on the number or trials

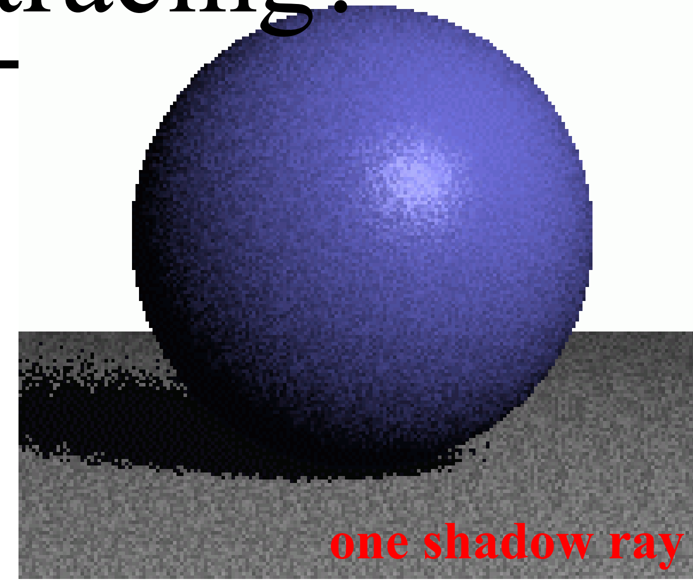


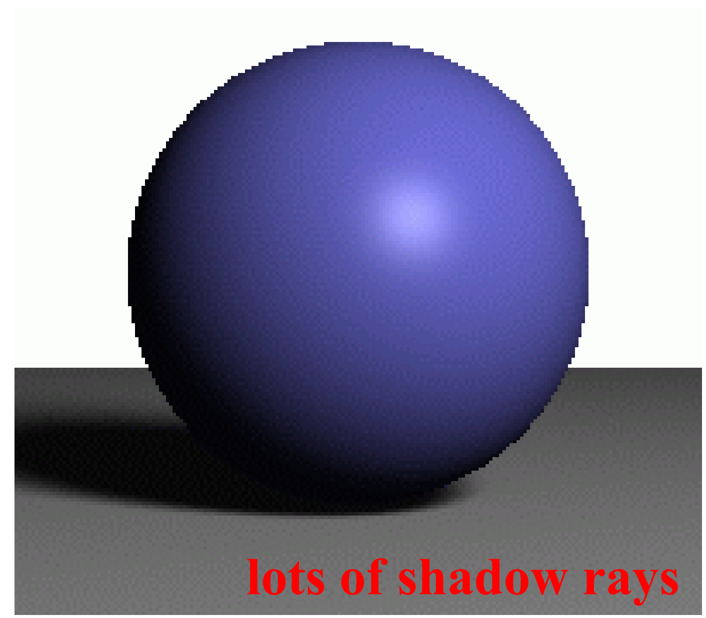
Why not use Simpson integration?

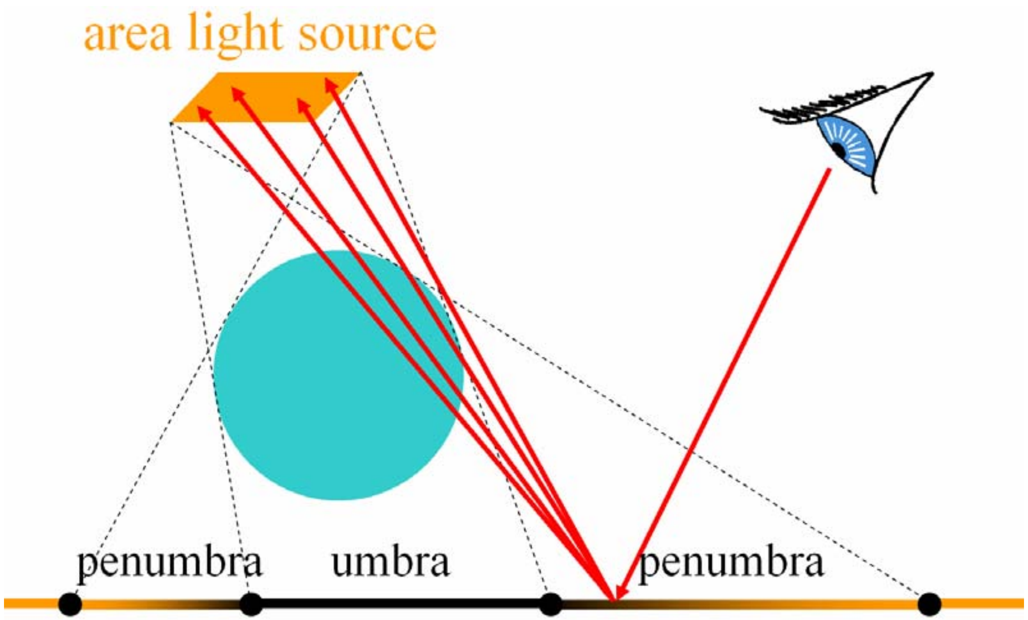
- Yeah, to compute π,
  Monte Carlo is not very efficient
- But convergence is independent of dimension
- Better to integrate high-dimensional functions
- For d dimensions, Simpson requires N^d domains

What's the link to ray tracing?

- Light source is square
- Occluder is sphere
- Some pixels have exactly $\pi/4$ visibility ;-)







Dumbest Monte-Carlo integration

- Compute 0.5 by flipping a coin
- 1 flip: 0 or 1=> average error =0.5
- 2 flips: 0, 0.5, 0.5 or 1 =>average error=0. 25
- 4 flips: 0 (*1),0.25 (*4), 0.5 (*6), 0.75(*4), 1(*1) => average error =0.1875

- Does not converge very fast
- Doubling the number of samples does not double accuracy

Convergence of Monte Carlo integration

- Variance decrease in 1/n
- Convergence is $\frac{1}{\sqrt{n}}$
- Independent of dimension

Monte Carlo integration

- Want to evaluate $$\int_a^b f(x)dx$$
- Use random variable x_i with uniform probability over the domain
$$\frac{1}{n}\sum_{i=1}^{n} f(x_i)$$
- Note that achieving uniform probability can be tough for complex domain (e.g. sphere)

Monte Carlo integration – smarter

- Want to evaluate $\int_a^b f(x)dx$
- Use random variable x_i **with probability p_i**

$$\frac{1}{n}\sum_{i=1}^{n}\frac{f(x_i)}{p_i}$$

- The whole trick is to choose the x_i and p_i

Sampling strategy

- Sample directions vs. sample light source

(Image removed due to copyright considerations.)

Sampling strategies

(Images removed due to copyright considerations.)

Sampling more the light

Sampling more the BRDF

Monte Carlo recap

- Turn integral into finite sum
- Use random samples
- Convergence $\frac{1}{\sqrt{n}}$
- Independent of dimension
- Very flexible

- Tweak sampling/probabilities for optimal result
- A lot of integration and probability theory to get things right

What can we integrate?

- Pixel: antialiasing
- Light sources: Soft shadows
- Lens: Depth of field
- Time: Motion blur
- BRDF: glossy reflection
- Hemisphere: indirect lighting

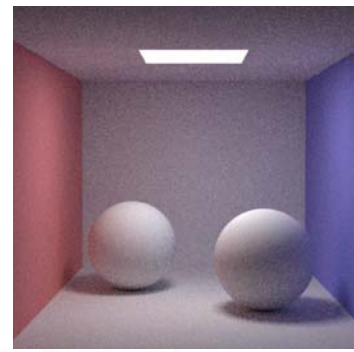
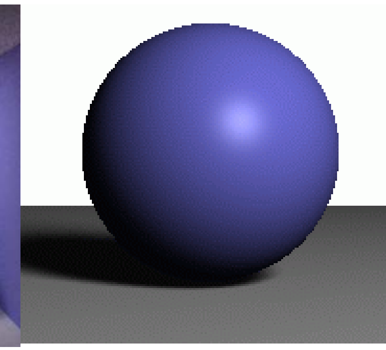
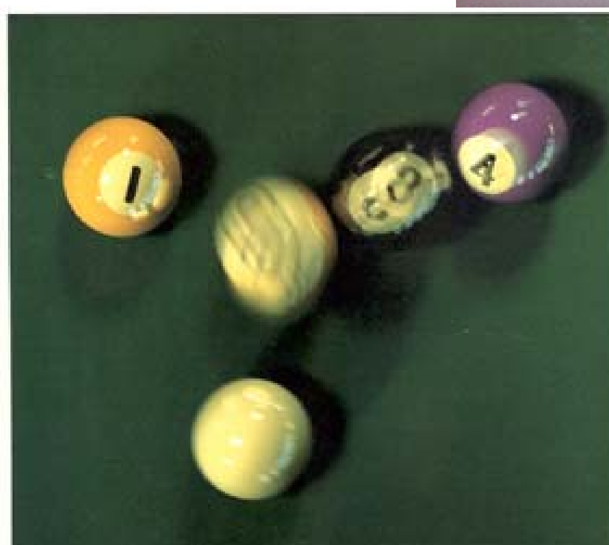
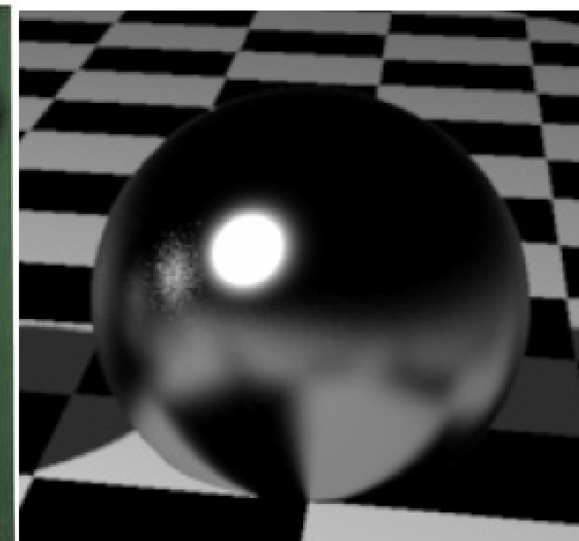



Image removed due to copyright considerations.
Please see *Computer Graphics* 25, no. 4
(July,1991): 157-164. ISBN: 0097-8930.

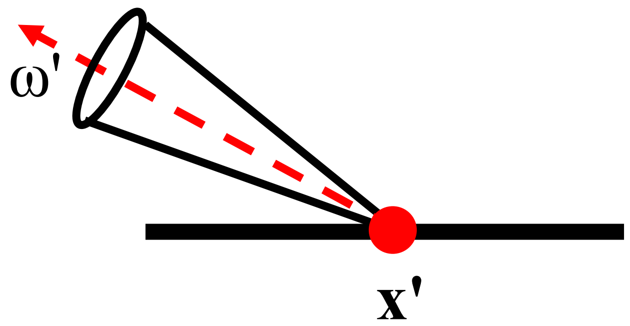
Questions?

(Images removed due to copyright considerations.)

Today: Monte Carlo Ray Tracing

- Principle of Monte-Carlo Ray Tracing
- Monte Carlo integration
- Review of Rendering equation
- Advanced Monte Carlo Ray Tracing

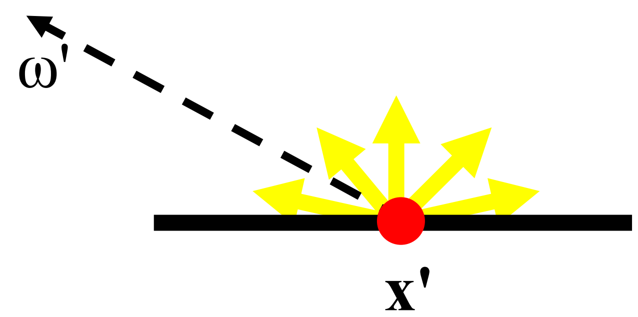
The Rendering Equation



$$L(x',\omega') = E(x',\omega') + \int \rho_{x'}(\omega,\omega')L(x,\omega)G(x,x')V(x,x')\, dA$$

L (x',ω') is the radiance from a point on a surface in a given direction ω'

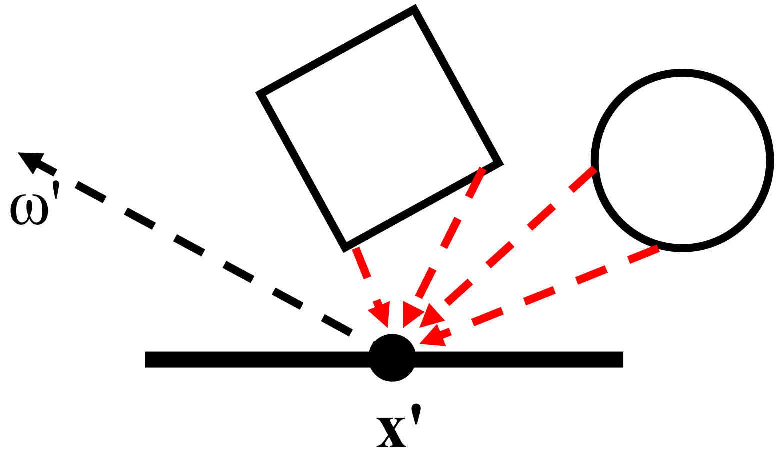
The Rendering Equation



$$L(x',\omega') = E(x',\omega') + \int \rho_{x'}(\omega,\omega')L(x,\omega)G(x,x')V(x,x')\,dA$$

E(x',ω') is the emitted radiance from a point: E is non-zero only if x' is emissive (a light *source*)

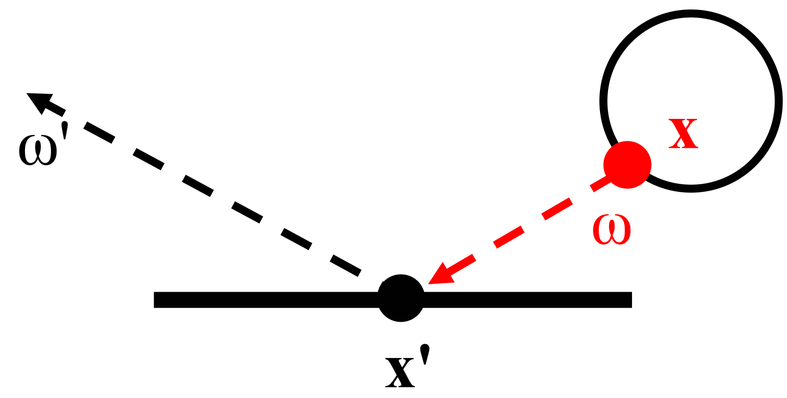
The Rendering Equation



$$L(x',\omega') = E(x',\omega') + \underbrace{\int \rho_{x'}(\omega,\omega') L(x,\omega) G(x,x') V(x,x')\, dA}$$

Sum the contribution from all of the other surfaces in the scene

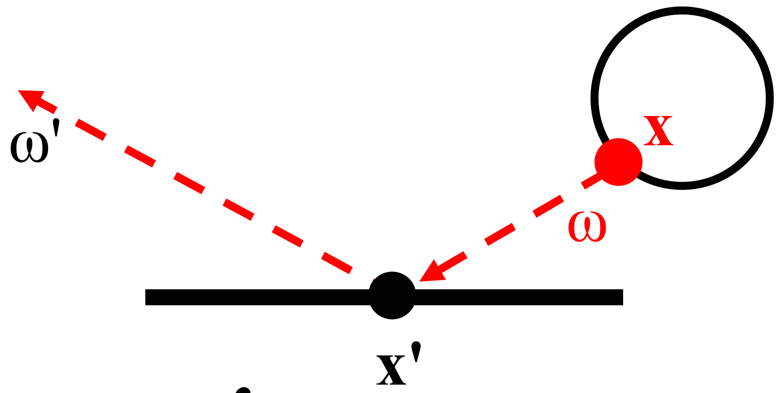
The Rendering Equation



$$L(x',\omega') = E(x',\omega') + \int \rho_{x'}(\omega,\omega') L(x,\omega) G(x,x') V(x,x')\, dA$$

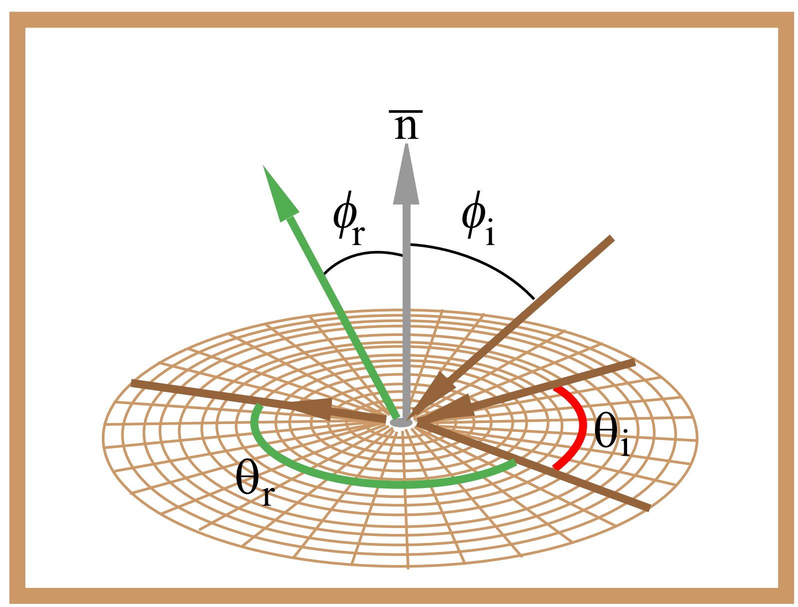
For each x, compute L(x, ω), the radiance at point x in the direction ω (from x to x')

The Rendering Equation

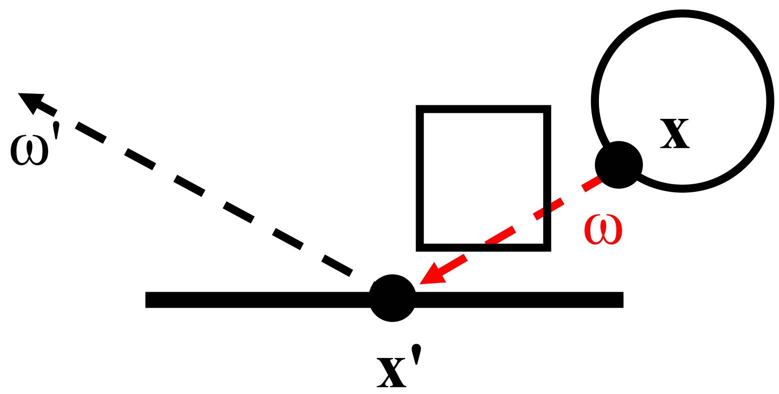


$$L(x',\omega') = E(x',\omega') + \int \rho_{x'}(\omega,\omega') L(x,\omega) G(x,x') V(x,x')\, dA$$

scale the contribution by $\rho_{x'}(\omega,\omega')$, the reflectivity (BRDF) of the surface at x'



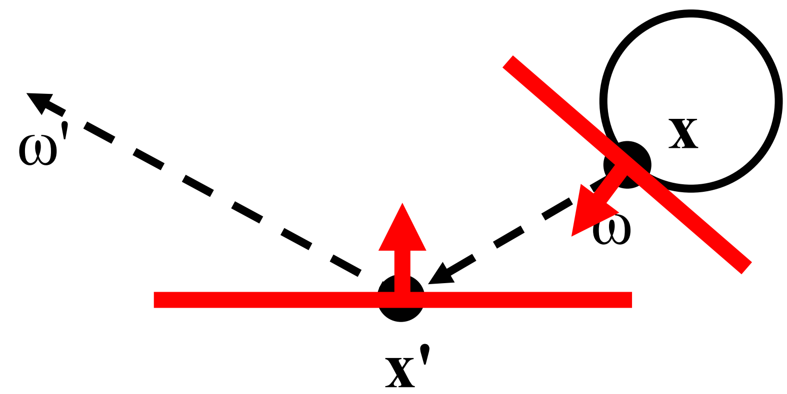
The Rendering Equation



$$L(x',\omega') = E(x',\omega') + \int \rho_{x'}(\omega,\omega')L(x,\omega)G(x,x')V(x,x')\,dA$$

For each x, compute V(x,x'),
the visibility between x and x':
1 when the surfaces are unobstructed
along the direction ω, 0 otherwise

The Rendering Equation

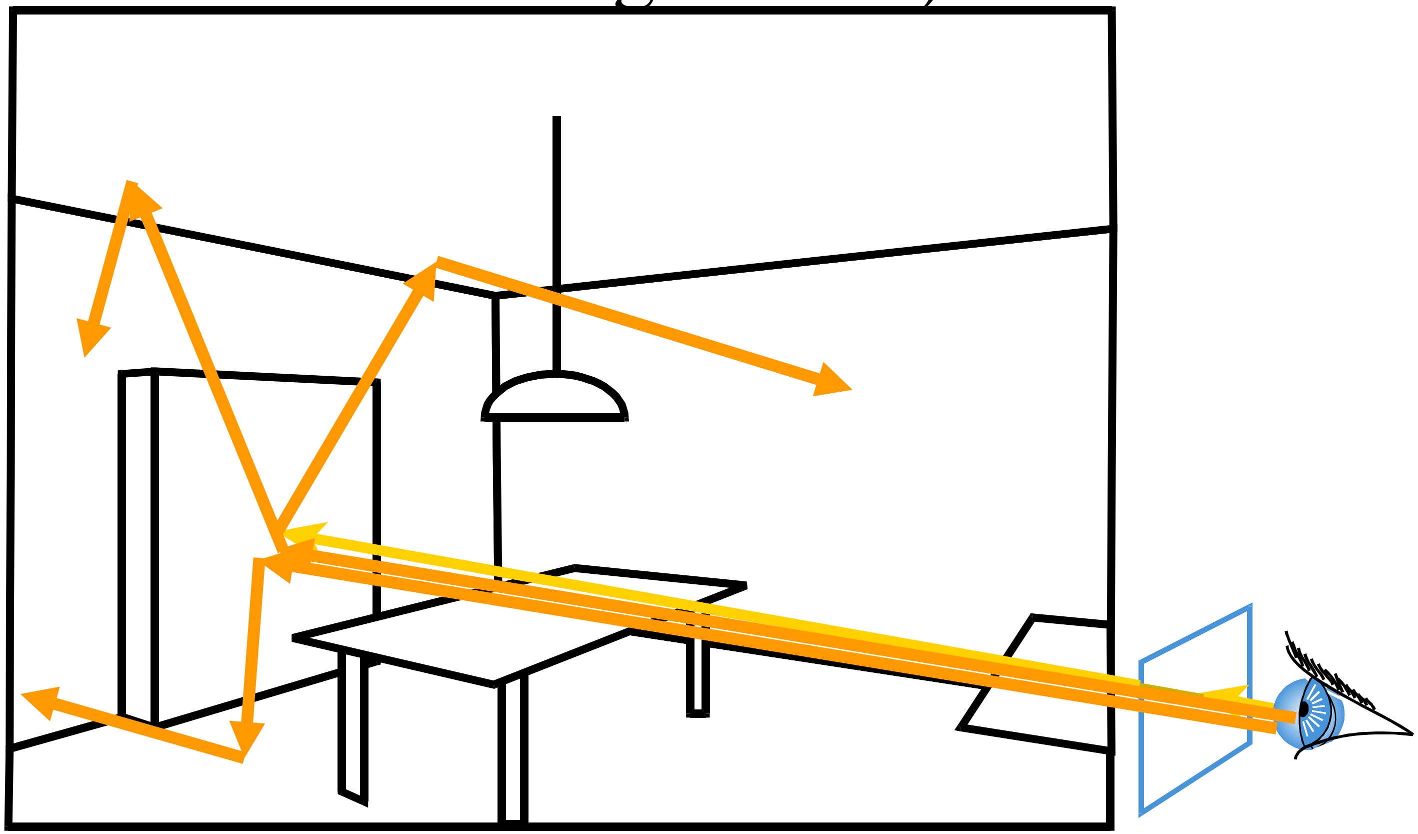


$$L(x',\omega') = E(x',\omega') + \int \rho_{x'}(\omega,\omega') L(x,\omega) G(x,x') V(x,x')\, dA$$

For each x, compute G(x, x'), which describes the on the geometric relationship between the two surfaces at x and x’

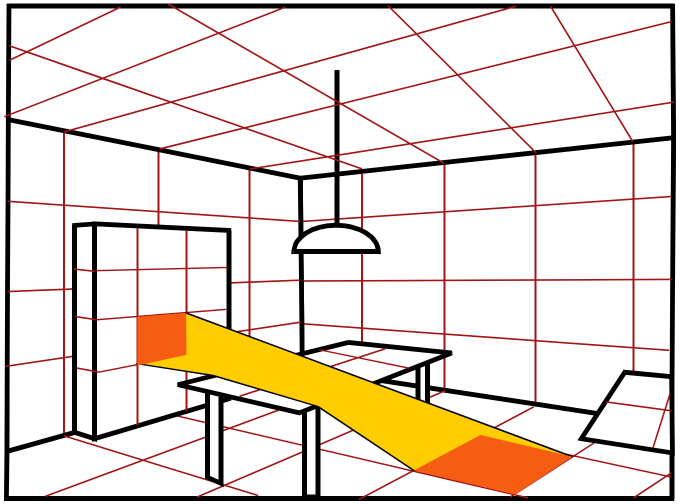
Monte Carlo Path Tracing

- Trace only one secondary ray per recursion
- But send many primary rays per pixel
- (performs antialiasing as well)

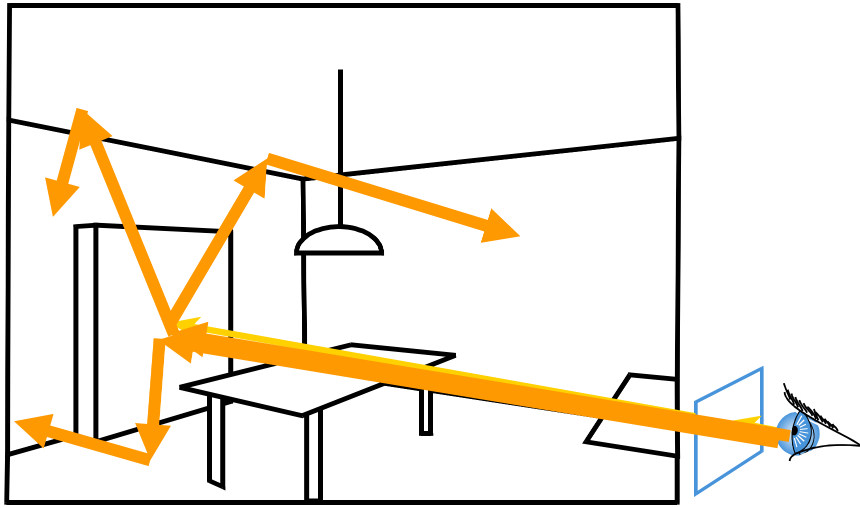


Radiosity vs. Monte Carlo

- We have an integral equation on an infinite space
- Finite elements (Radiosity)
  - Project onto finite basis of functions
  - Linear system

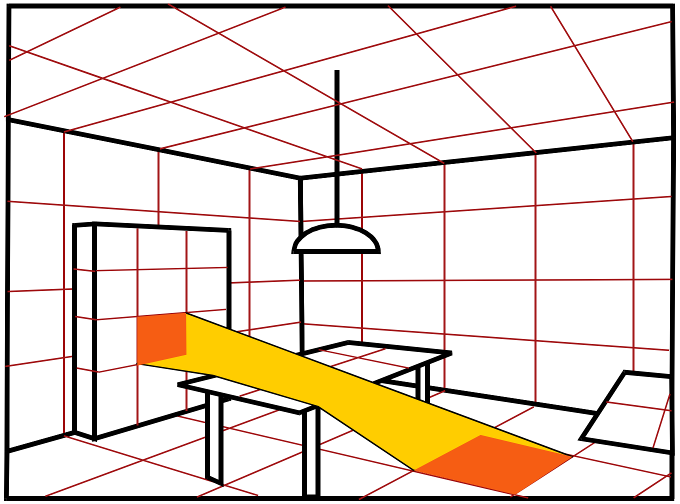


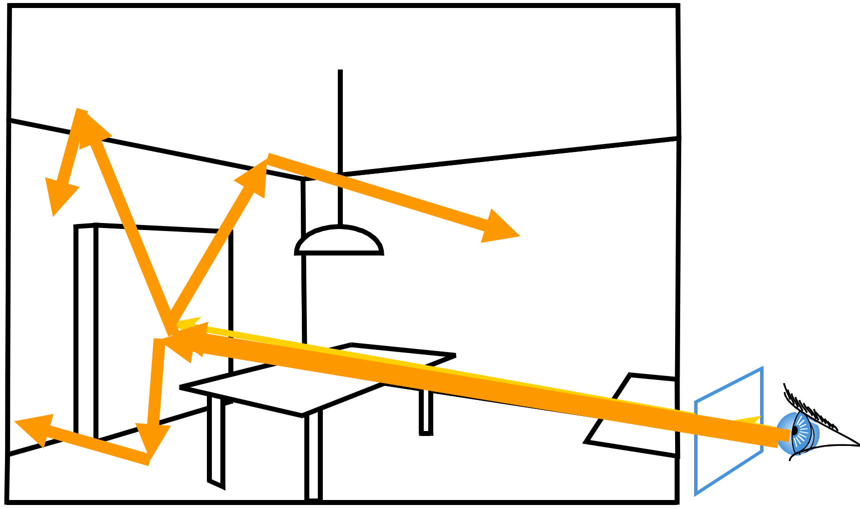
- Monte Carlo
  - Probabilistic sampling



Radiosity vs. Monte Carlo

- We have an integral equation on an infinite space
- Finite elements (Radiosity)
  - Project onto finite basis of functions
  - Linear system
  - View-independent (no angular information)
- Monte Carlo
  - Probabilistic sampling
  - View-dependent (but angular information)





References

- See also Eric Veach's PhD
  http://graphics.stanford.edu/papers/veach_thesis/

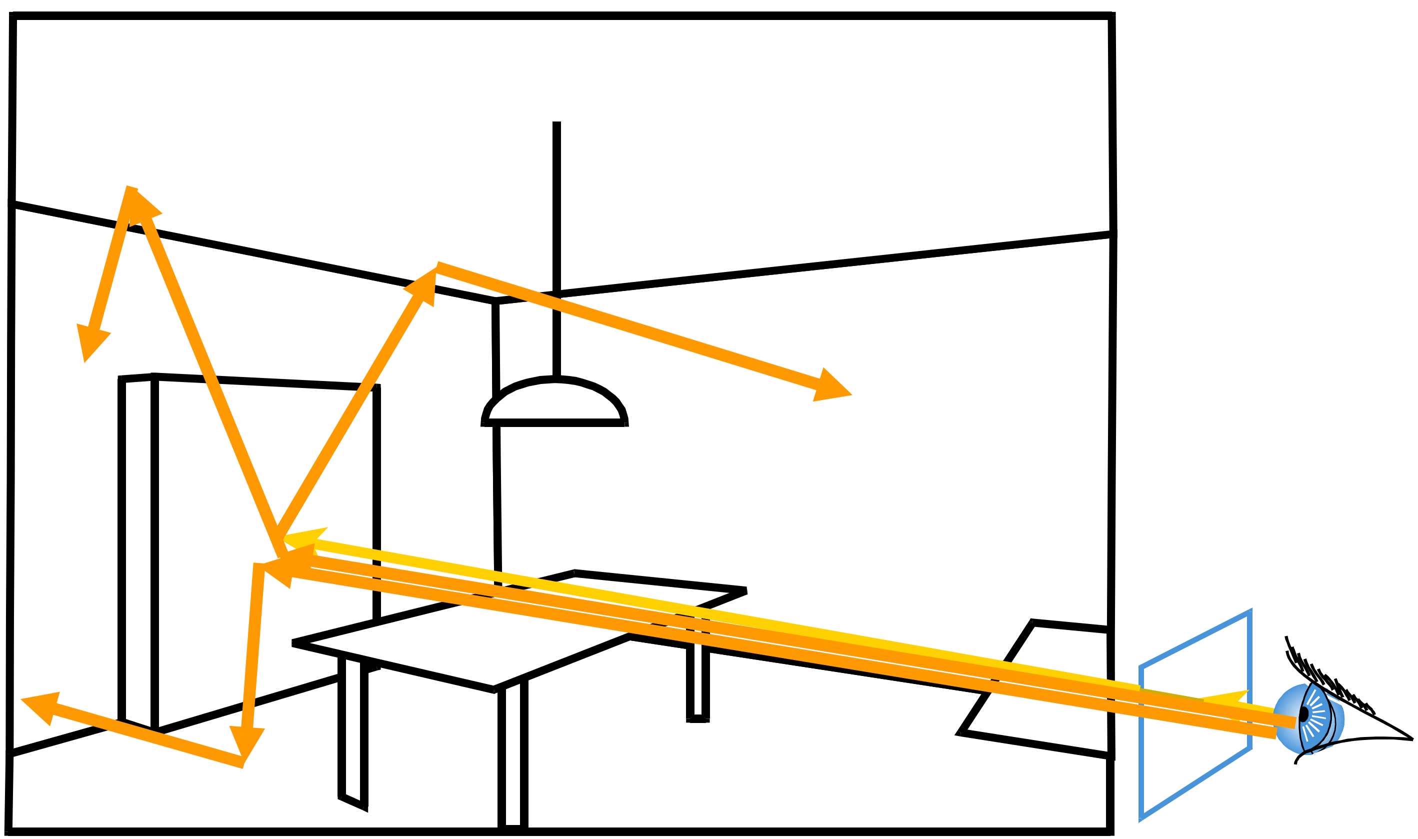
- Dutre, P., P. Bekaert and K. Bala. *Advanced Global Illumination*. AK Peters, Ltd. 2003. ISBN:1568811772.
- Shirley, Pete. *Realistic Ray Tracing*. AK Peters, Ltd. 1st. Ed. 2000. ISBN:1568811101.
- Jensen, Henrik Wann. *Realistic Image Synthesis Using Photon Mapping*. AK Peters, Ltd. 1st Ed. 2001. ISBN:1568811470.

Today: Monte Carlo Ray Tracing

- Principle of Monte-Carlo Ray Tracing
- Monte Carlo integration
- Review of Rendering equation
- Advanced Monte Carlo Ray Tracing
  - Irradiance caching
  - Photon mapping

Path Tracing is costly

- Needs tons of rays per pixel



Direct illumination

(Image removed due to copyright considerations.)

Global Illumination

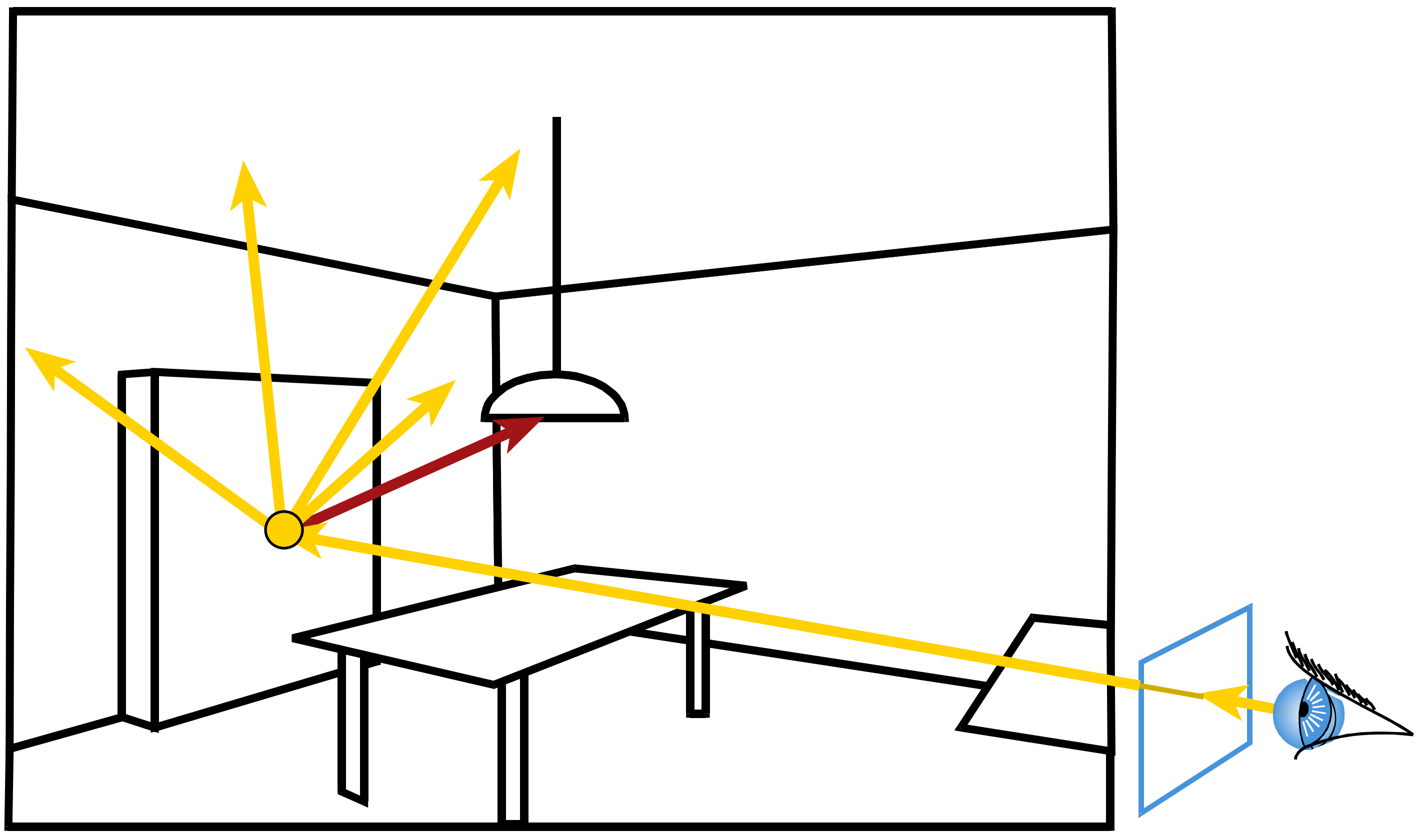
(Image removed due to copyright considerations.)

Indirect illumination: smooth

(Image removed due to copyright considerations.)

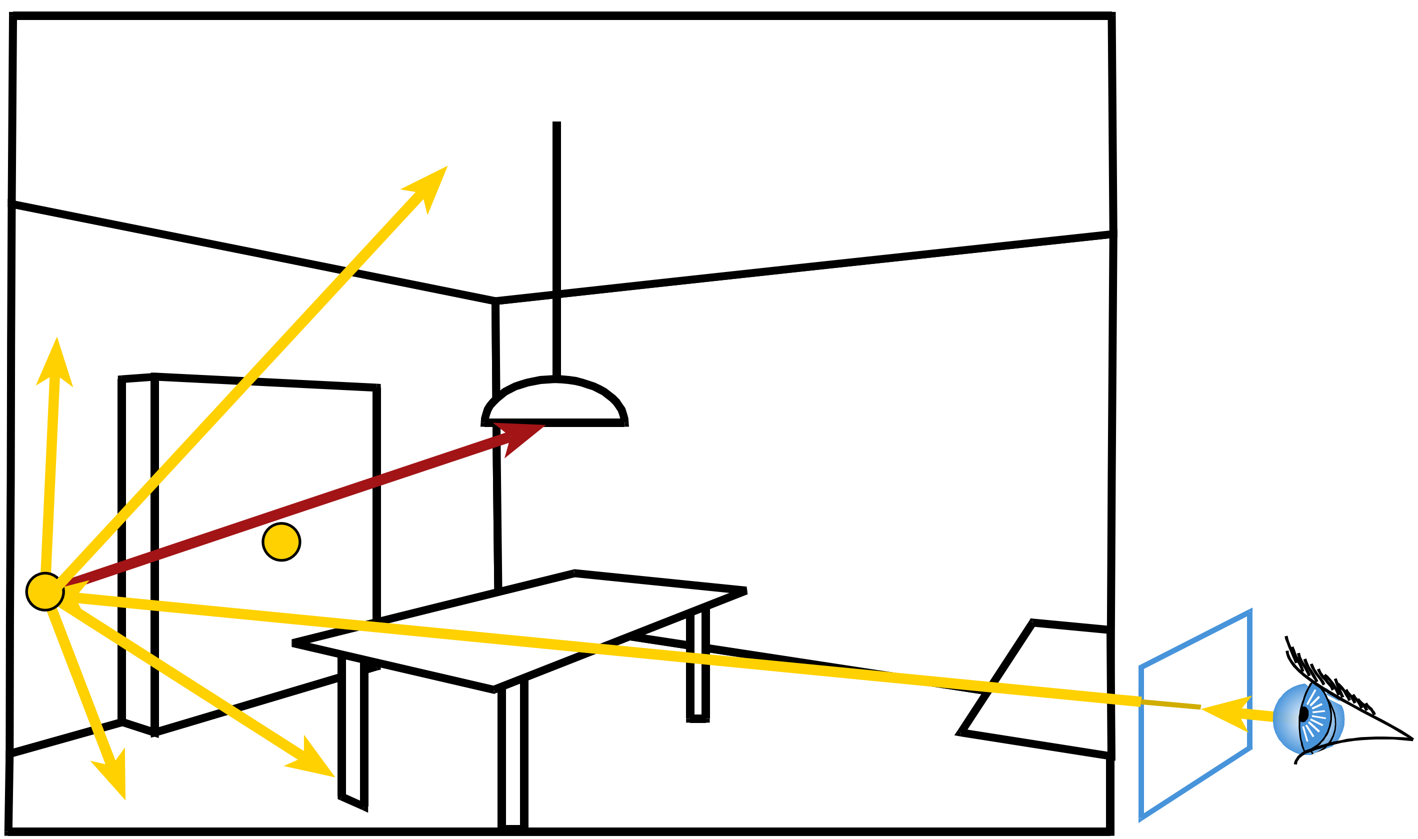
Irradiance cache

- The indirect illumination is smooth



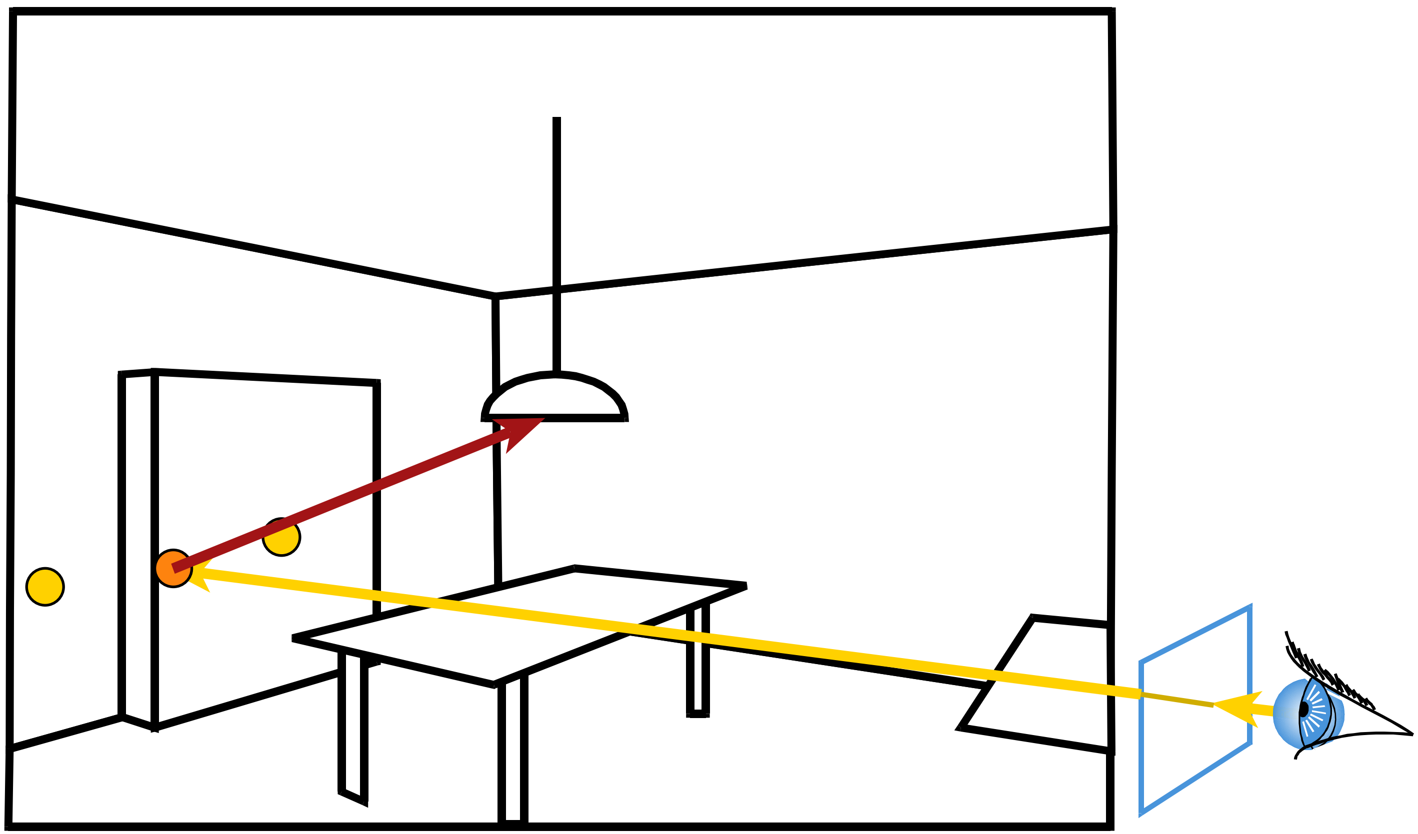
Irradiance cache

- The indirect illumination is smooth



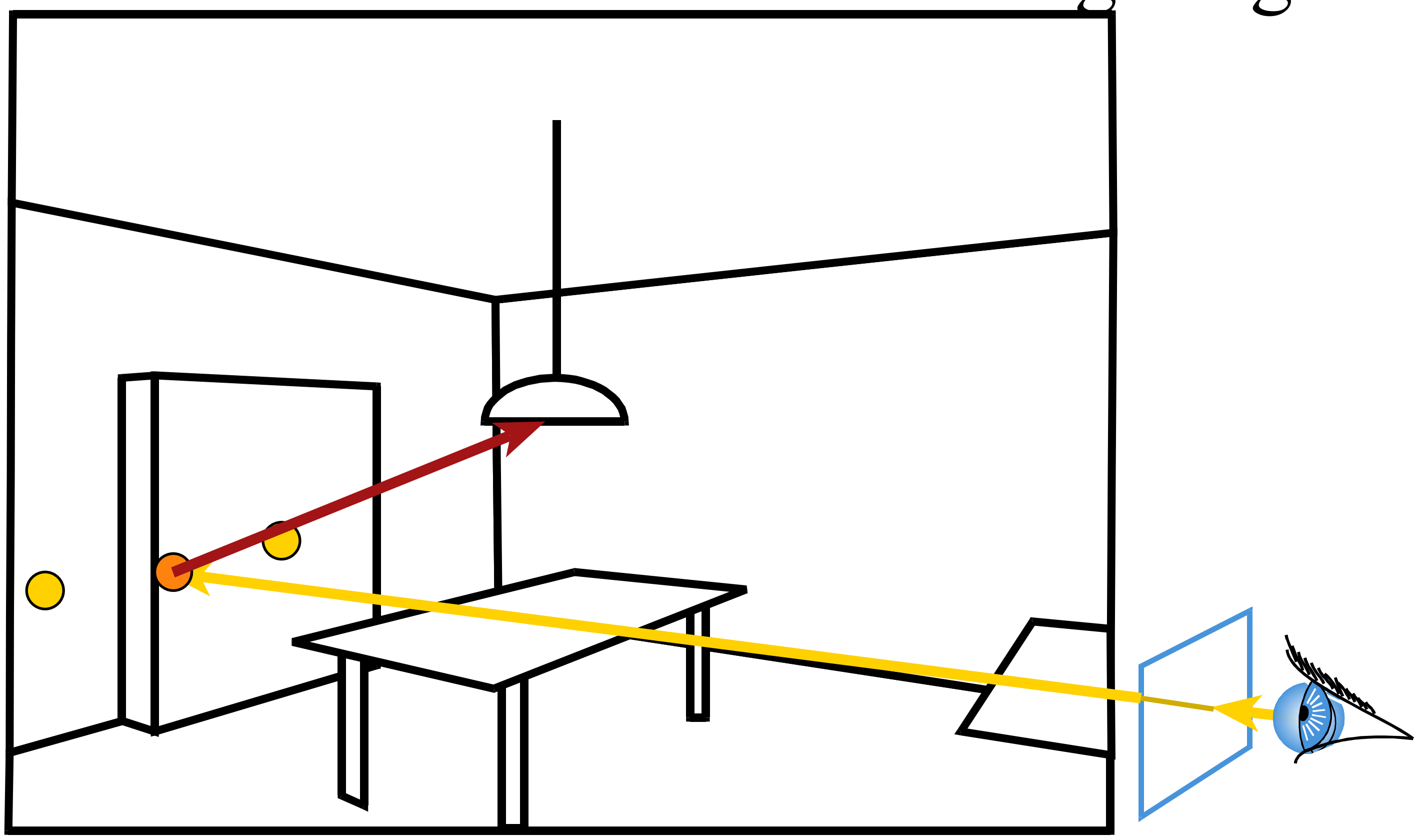
Irradiance cache

- The indirect illumination is smooth
- Interpolate nearby values



Irradiance cache

- Store the indirect illumination
- Interpolate existing cached values
- But do full calculation for direct lighting

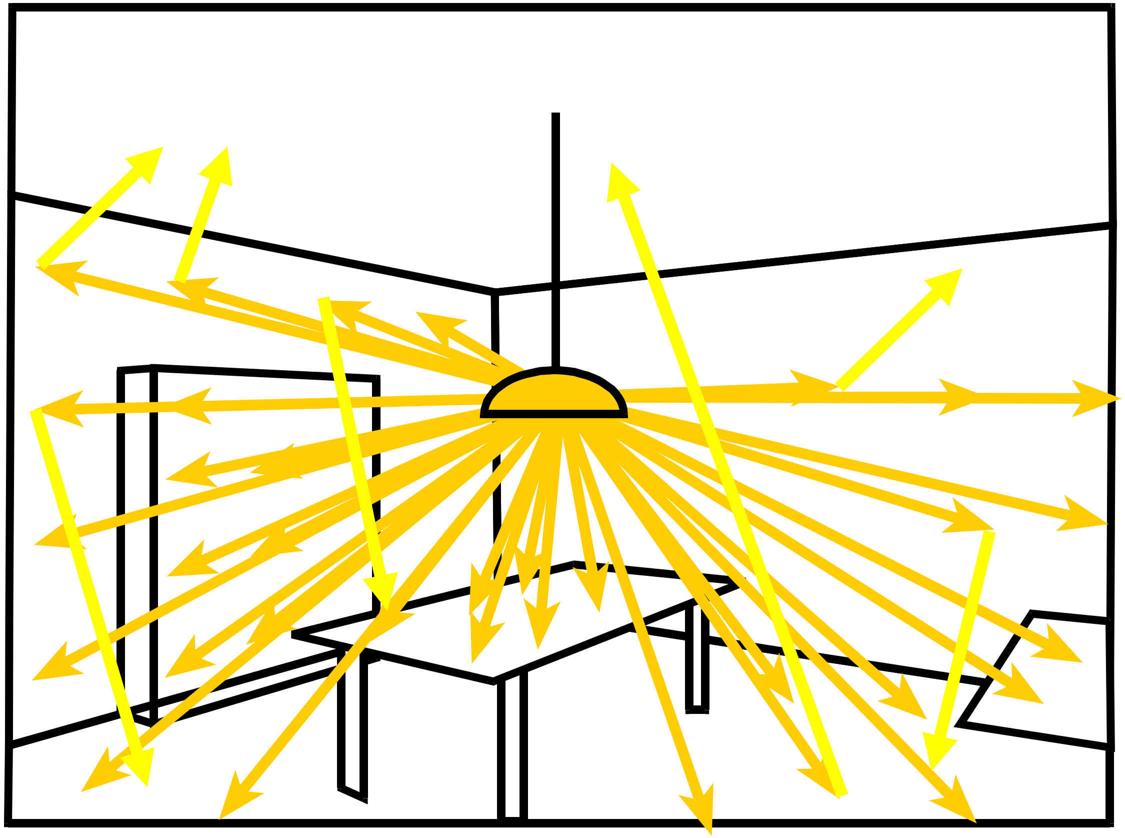


Irradiance caching

(Image removed due to copyright considerations.)

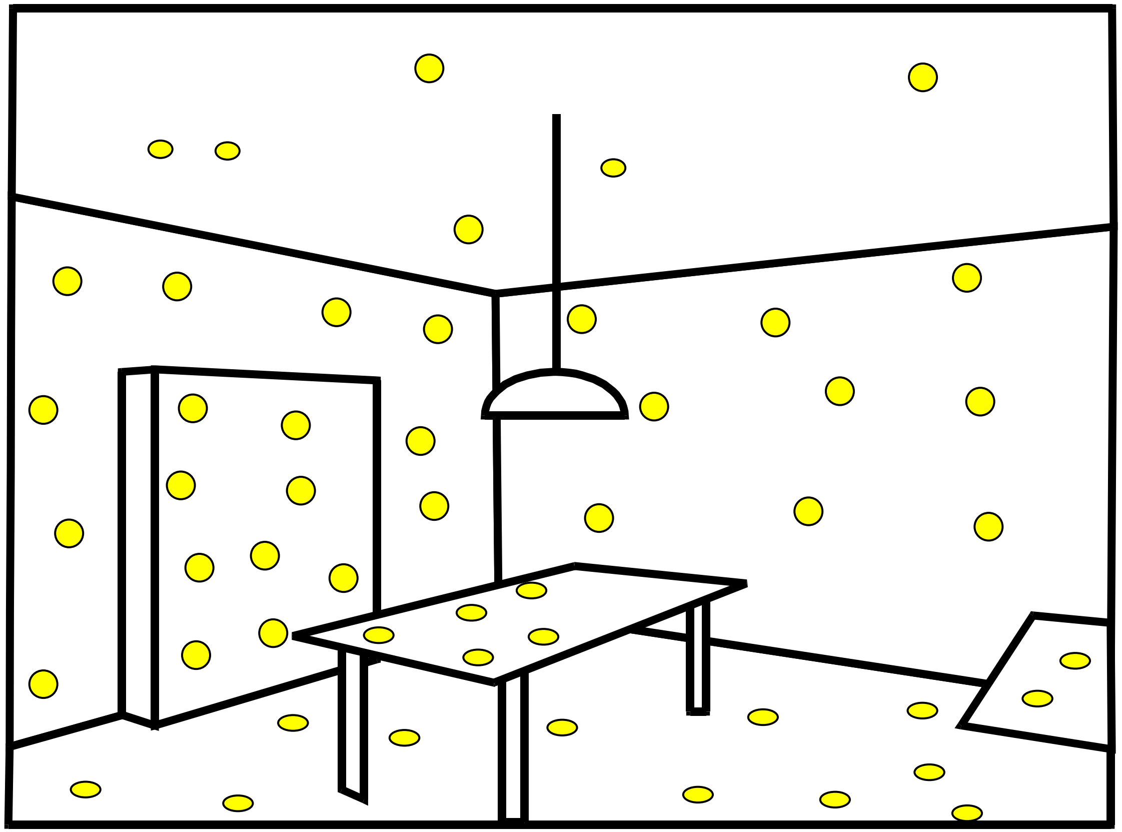
Photon mapping

- Preprocess: cast rays from light sources
- Store photons



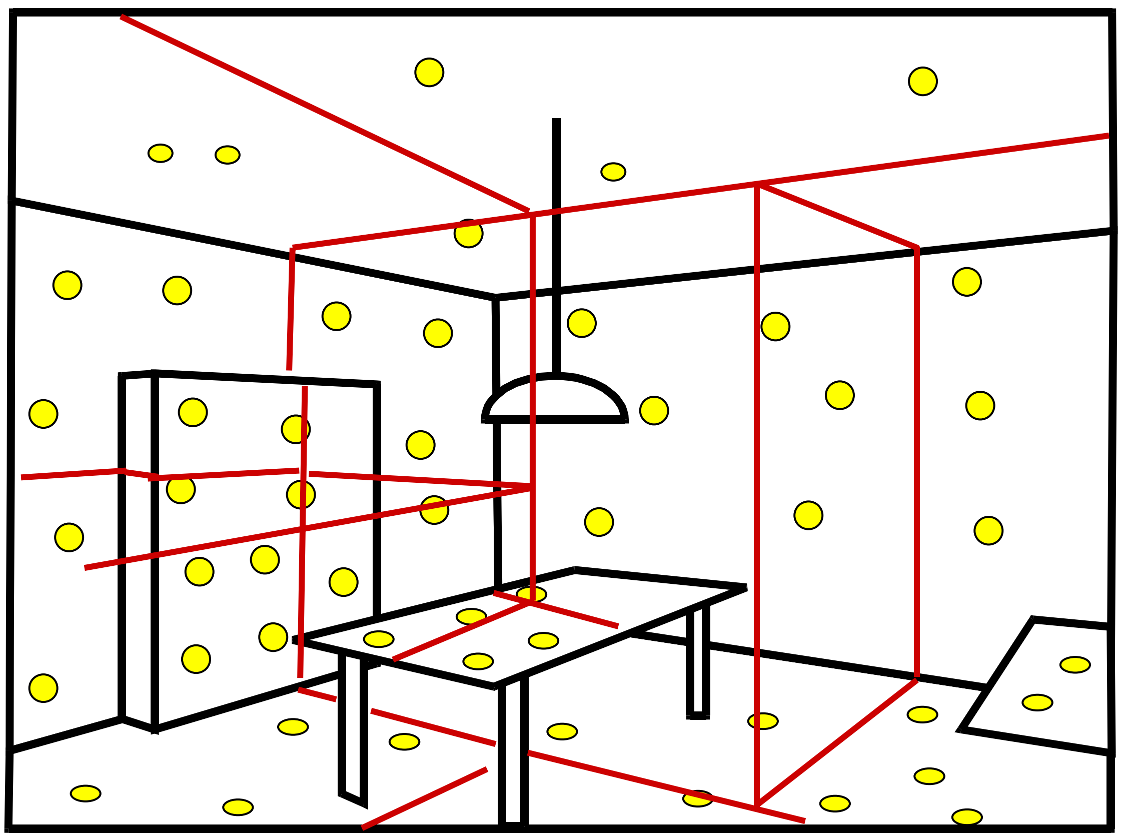
Photon mapping

- Preprocess: cast rays from light sources
- Store photons (position + light power + incoming direction)



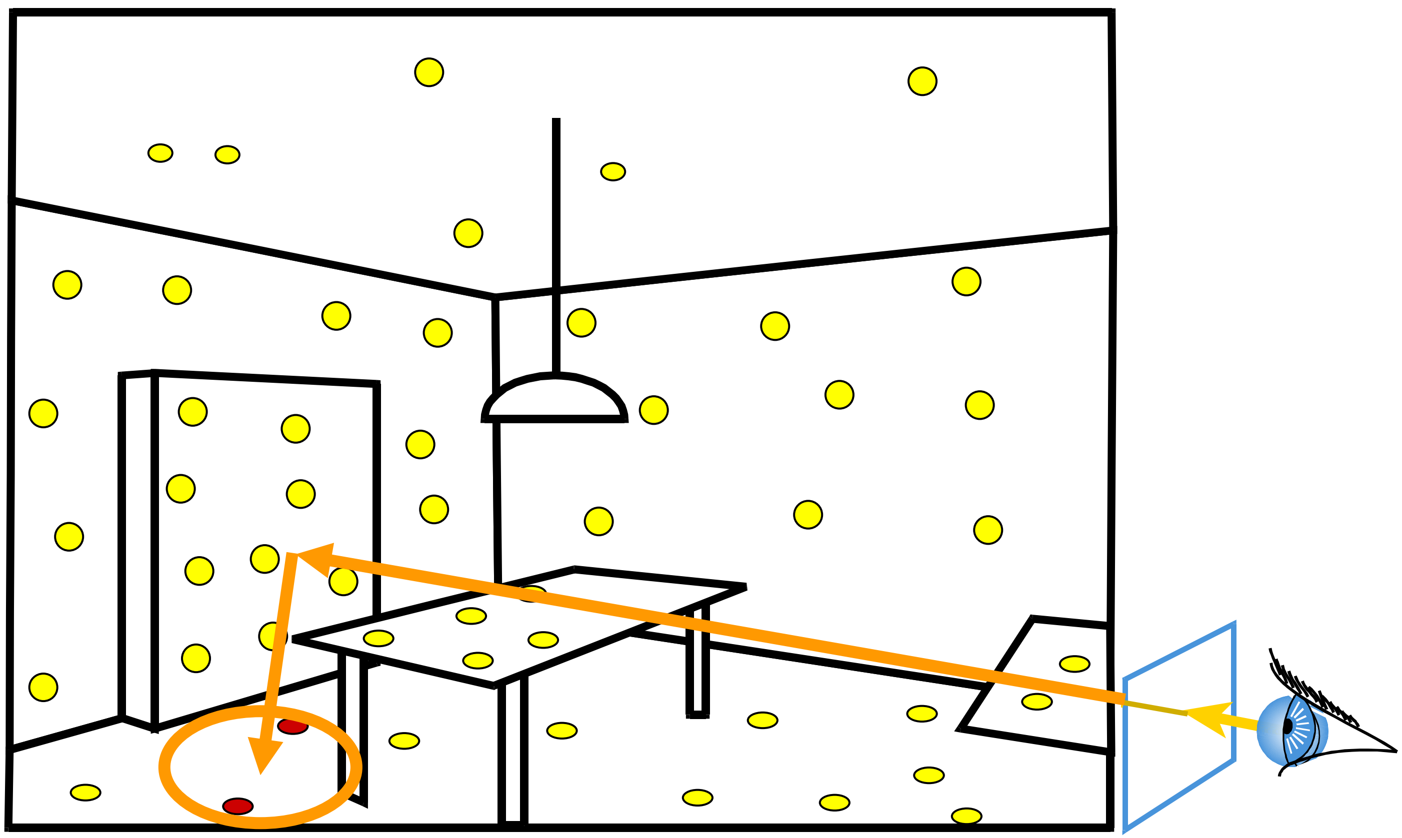
Photon map

- Efficiently store photons for fast access
- Use hierarchical spatial structure (kd-tree)



Photon mapping - rendering

- Cast primary rays
- For secondary rays
  - reconstruct irradiance using adjacent stored photon
  - Take the k closest photons
- Combine with irradiance caching and a number of other techniques

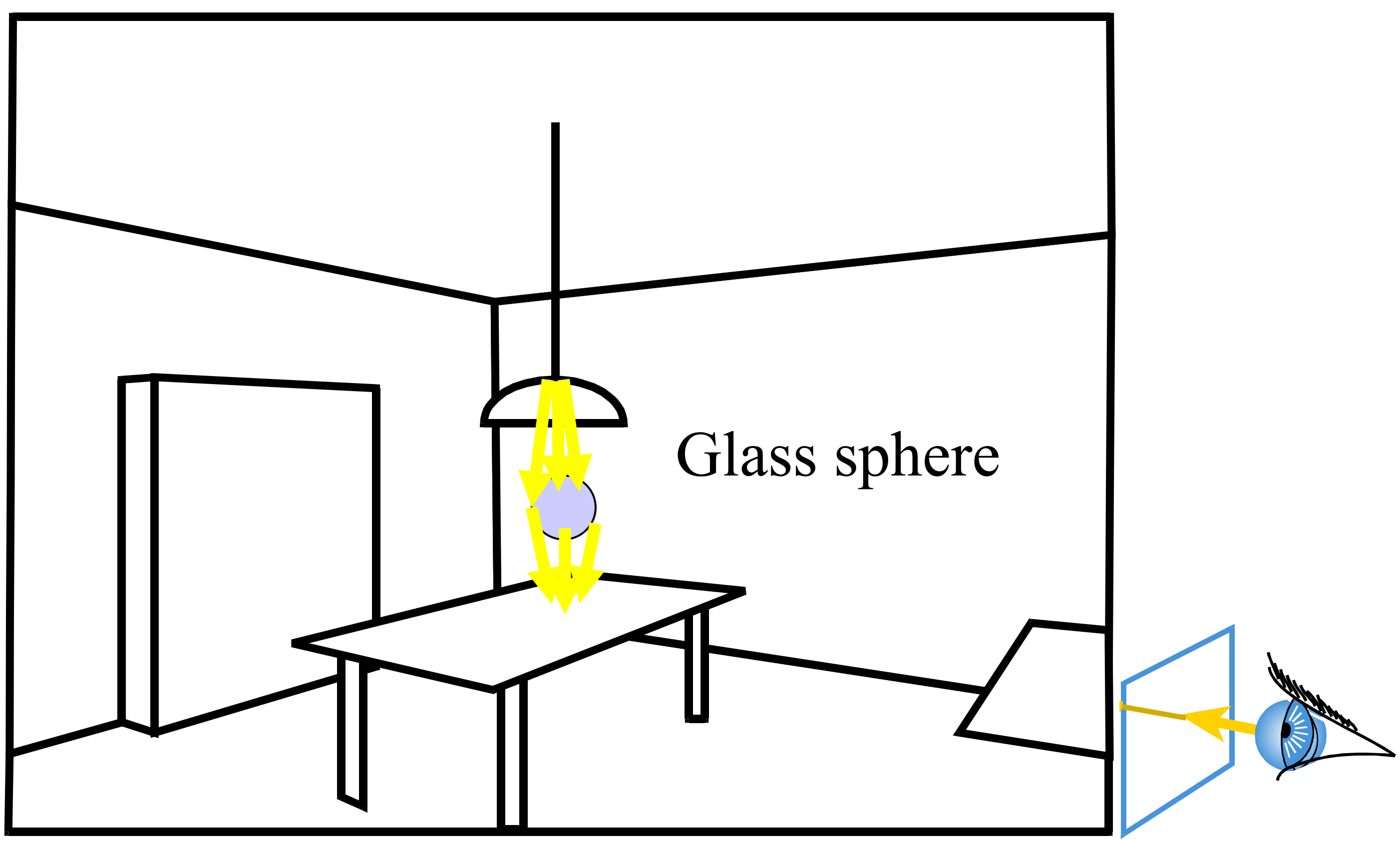


Photon map results

(Image removed due to copyright considerations.)

Photon mapping - caustics

- Special photon map for specular reflection and refraction



- 1000 paths/pixel

(Image removed due to copyright considerations.)

- Photon mapping

(Image removed due to copyright considerations.)

Photon mapping recap

- Preprocess: Path tracing
  - Store photons
  - Special map for caustics (because not as uniform)
- Rendering
  - Primary rays
  - Direct lighting
  - Indirect lighting
    - Reconstruct irradiance using k nearest photons
    - Irradiance caching
  - Reconstruct caustics from caustic map

Questions?

(Image removed due to copyright considerations.)

Why is the sky blue?

(Image removed due to copyright considerations.)

Answer: because sunset is red

(Image removed due to copyright considerations.)

Why is the sky blue?

(Image removed due to copyright considerations.)

Sampling strategies

(Image removed due to copyright considerations.)

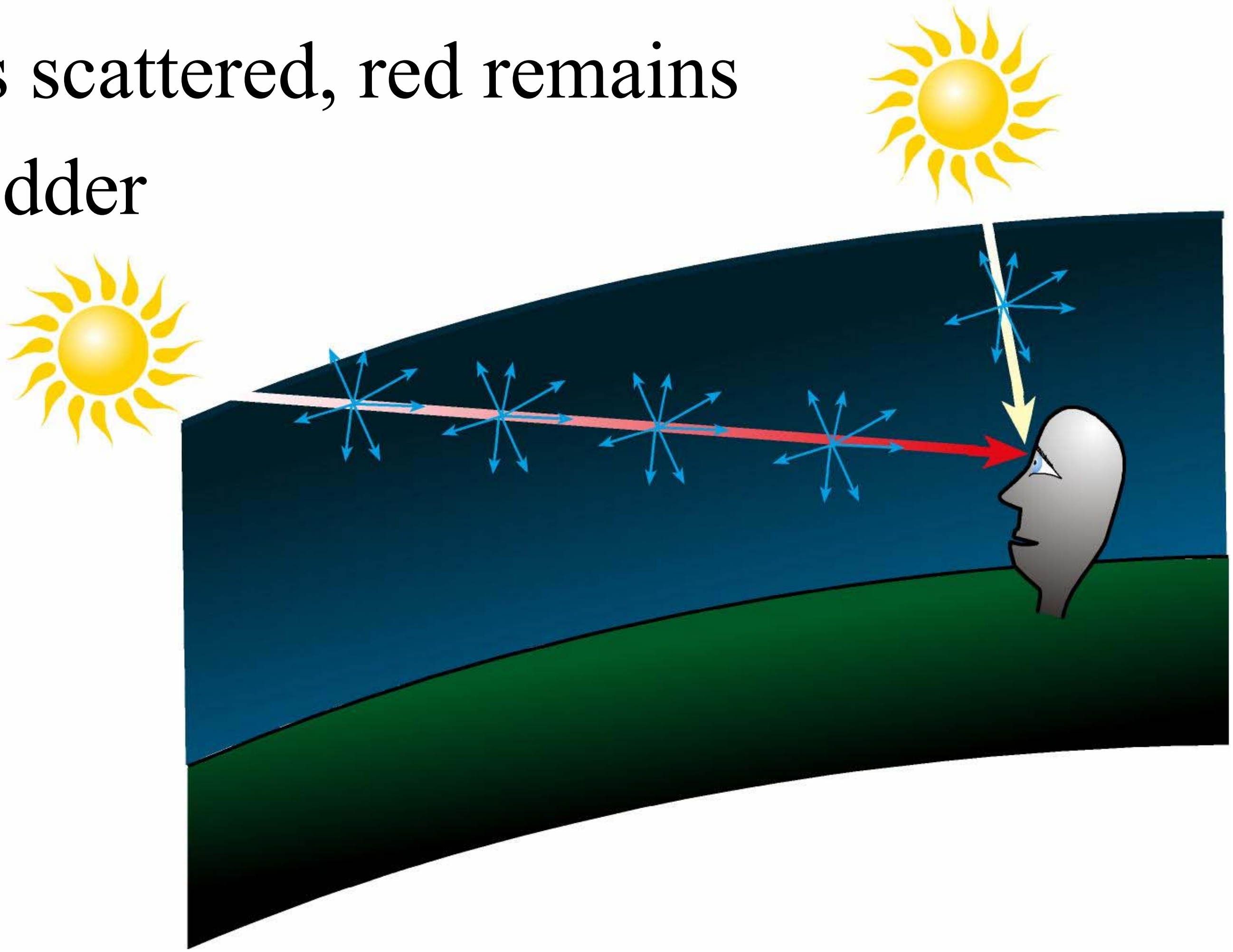
Sampling more the light

Sampling more the BRDF

Why is the sky blue?

Sun Color

- At sunset, longer traversal of atmosphere
- More blue is scattered, red remains
- Therefore redder



Computer graphics sky models

- E.g. Preetham et al. SIGGRAPH 99.

(Images removed due to copyright considerations.)

Night Sky Model

- Jensen, Durand, Stark, Premoze, Dorsey, Shirley 2001

(Images removed due to copyright considerations.)

Questions?

(Image removed due to copyright considerations.)