# Fundamentals of Photorealistic Rendering (Monte Carlo Integration and Ray Tracing)

Lecturer: Bochang Moon
GIST

# Monte Carlo (MC) Ray Tracing

- MC rendering (ray tracing): rendering (ray tracing) algorithms that solve the light transport integral via Monte Carlo integration.
  - Considered impractical 10 - 20 years ago, even for offline scenarios.
    - e.g., Only limited effects (ambient occlusions) are simulated via MC ray tracing.
  - Become a de facto standard in production.
    - A popular combination: Path tracing + Image-space denoising
  - Ref.
    - Special Issue on Production Rendering (Volume 37, Issue 3, ACM Transactions on Graphics)
  - Not popular in real-time applications, but it started being integrated into real-time rendering engines.

# References

- Books
  - Fundamentals of Computer Graphics, 4th edition, Steve Marschner and Peter Shirley
  - Physically Based Rendering From Theory to Implementation, 3rd edition, Matt Pharr, Wenzel Jakob, Greg Humphreys
  - Realistic Ray Tracing, 2nd edition, Peter Shirley and R. Keith Morley
    - Recommend it as a starting point if you are not familiar with rendering.
  - Realistic Image Synthesis Using Photon Mapping, Henrik Wann Jensen
  - Density Estimation for Statistics and Data Analysis, B. W. Silverman
- Papers

# Outline

- Backgrounds: Light, Irradiance, Radiance, BRDF
- Light Transport Equation
- Background: Monte Carlo Integration
- Monte Carlo Ray Tracing: Distributed Ray Tracing

# Light

- A form of energy
- Q. How do we measure the light?
- Radiometry is a set of techniques for measuring light
- International System of Units (SI)
  - e.g., meter (m), gram (g), joule (J)

# Photons

- Described as collections of a large number of photons
- Photons
  - A photon is a quantum of light
  - Position, direction of propagation, a wavelength $\lambda$
  - c.f., SI unit for $\lambda$ is nm ($1nm = 10^{-9}m$)
  - Amount of energy carried by a photon
    - $q = hf = \frac{hc}{\lambda}$
    - $h = 6.63 \times 10^{-34} Js$ (Plank's Constant)
    - Frequency $f = \frac{c}{\lambda}$
    - c is the photon's speed, and it depends on the refractive index of the medium where light propagates

# Spectral Energy

- The total energy can be measured by summing the energy $q_i$ of each photon
  - i = index of a photon

- Spectral Energy ($Q_\lambda$)
  - Measure the amount of light energy across wavelengths
  - e. g. , When the summed energy is 10.2 J between $\lambda = 500$ nm and $\lambda = 600$ nm
    - $Q_\lambda[500,600] = \frac{10.2}{100} = 0.12\ J(nm)^{-1}$

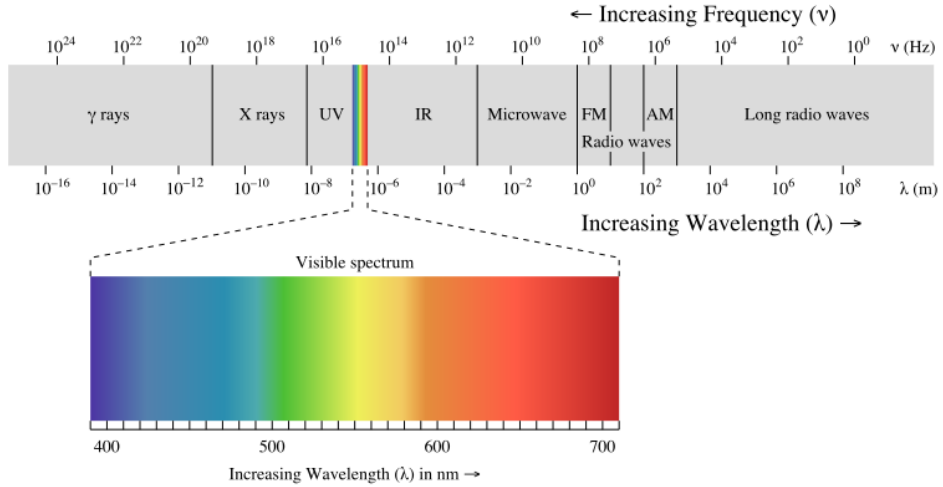



Image: wikipedia.org

# Power

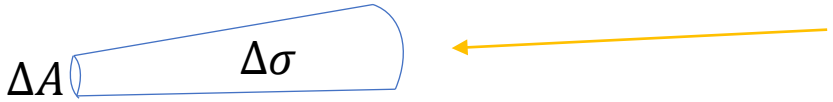
- Power
  - Rate of energy produced by light sources
  - Watts (W) = joules / second
  - e.g., 100-watt light bulb
    - Generate 100J per second
  - Spectral power $(W(nm)^{-1})$
    - Assume: a light emits a 100W power evenly across wavelengths 400nm to 800nm
    - Spectral power $\Phi_\lambda \equiv \Phi = \frac{100W}{400nm} = 0.25W(nm)^{-1}$
    - Can we calculate the spectral power for the case that a shutter of a measurement device opens for a time interval $\Delta t$ centered at time t?
      - Spectral power $\Phi = \frac{\Delta q}{\Delta t \Delta \lambda}$

# Irradiance

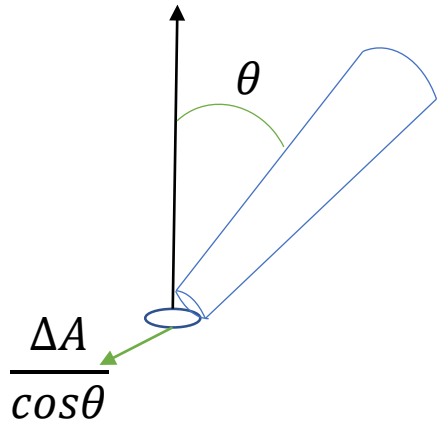
- Spectral irradiance H
  - Intuitively, it describes how much light arrives at a point
  - $\Delta\Phi/\Delta A$ (Power per unit area)
  - $H = \frac{\Delta q}{\Delta A \Delta t \Delta \lambda}$
  - Note that we use a finite area $\Delta A$ instead of a point
  - Units for the irradiance are $Jm^{-2}s^{-1}(nm)^{-1}$
- Radiant exitance (emittance), E
  - Describe how much light leaves from a point

# Radiance

- It describes how much light with a specific direction arrives at a point
- (spectral) radiance $= \frac{\Delta H}{\Delta \sigma} = \frac{\Delta \mathrm{q}}{\Delta A \Delta \sigma \Delta t \Delta \lambda}$
  - $\Delta \sigma$: solid angle



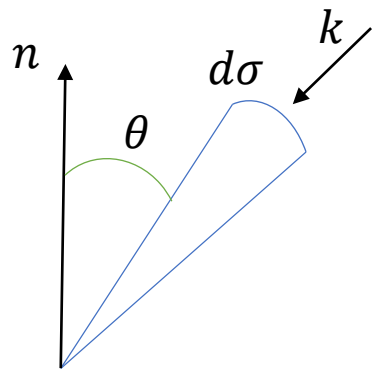

- (spectral) radiance at a surface $= \frac{\Delta H}{\Delta \sigma cos\theta} = \frac{\Delta \mathrm{q}}{\Delta A \cos\theta \Delta \sigma \Delta t \Delta \lambda}$
  - Radiance that hits a surface




- Surface radiance
  - Radiance that leaves a surface
  - $L_s = \frac{\Delta E}{\Delta \sigma cos\theta}$
- Field radiance
  - Radiance incident at a surface
  - $L_f = \frac{\Delta H}{\Delta \sigma cos\theta}$

# Radiance

- Irradiance can be expressed by summing radiance terms
- $H = \int_{all\ \boldsymbol{k}} L_f(k) cos\theta d\sigma$
- k is a unit vector
  - Incident direction
- $(\theta, \phi)$ is a spherical coordinates w.r.t. the surface normal
- Differential solid angle has the following relation in spherical coordinates:
  - $d\sigma = sin\theta d\theta d\phi$
- As a result,
  - $H = \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\frac{\pi}{2}} L_f cos\theta sin\theta d\theta d\phi = \pi L_f$
  - Q. What's the assumption for the equation above?
  - A. Constant field radiance $L_f$

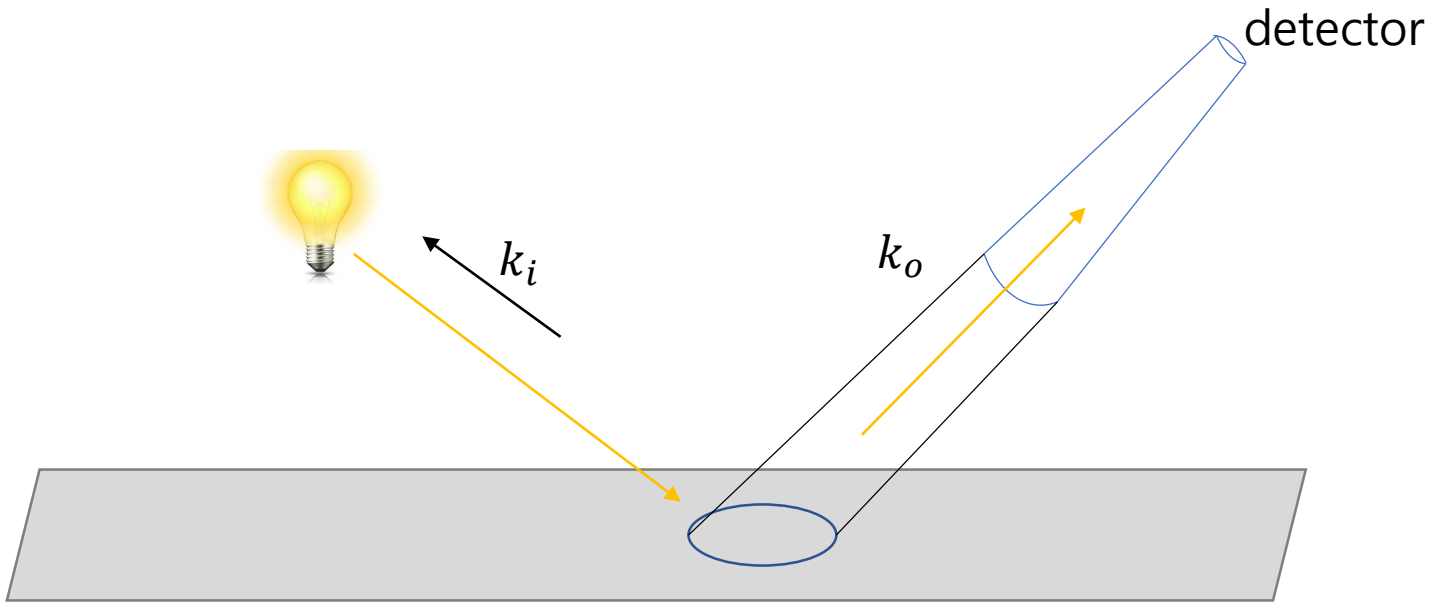


# Radiance

- Can we compute power of hitting a surface?
  - $\Phi = \int_{all\ \boldsymbol{x}} H(\boldsymbol{x}) dA$
    - Indicate that the power can be computed by integrating the irradiance across the surface area
    - **x**: a point on the surface

# BRDF

- Bidirectional reflectance distribution function (BRDF) that defines how light is reflected at a surface
- BDRF controls surface appearance
  - $\rho(\boldsymbol{k_i}, \boldsymbol{k_o})$



# BRDF

- Directional Hemispherical Reflectance: Describes the fraction of reflected light from incident light with direction $\boldsymbol{k_i}$
  - $R(\boldsymbol{k_i}) = \frac{power\ in\ all\ outgoing\ directions\ k_o}{power\ in\ a\ beam\ from\ direction\ k_i}$
  - $R(\boldsymbol{k_i})$ should be between 0 to 1.

- The reflectance can be represented with summing BRDF:
  - $R(k_i) = \int_{all\ \boldsymbol{k_o}} \rho(\boldsymbol{k_i}, \boldsymbol{k_o}) cos\theta_o d\sigma_o$

# Example: Ideal Diffuse Surface

- Ideal diffuse surface with the Lambertian BRDF
  - $\rho = C$, i.e., reflects light equally in all directions

- $R(k_i) = \int_{all\ k_o} C cos\theta_o d\sigma_o = \int_{\phi_o=0}^{2\pi} \int_{\theta_o}^{\frac{\pi}{2}} C cos\theta_o sin\theta_o d\theta_o d\phi_o = \pi C$

- When $R(k_i) = 1$
  - $\rho = C = \frac{1}{\pi}$

- When $R(k_i) = r$
  - $\rho(\boldsymbol{k_i}, \boldsymbol{k_o}) = C = \frac{r}{\pi}$

# Surface Reflection

- Related terms
  - BTDF (Bidirectional transmittance distribution function)
    - describes the transmission at a surface
  - BSDF (Bidirectional scattering distribution function)
    - includes BRDF and BTDF
  - BSSRDF (Bidirectional scattering surface reflectance distribution function)
    - Describes the ratio of outgoing radiance at a point with a direction from incoming radiance at another point with another direction
    - BRDF is an approximation of BSSRDF
    - Requires to simulate subsurface scattering (e.g., human skin)

# Representation of BSDF

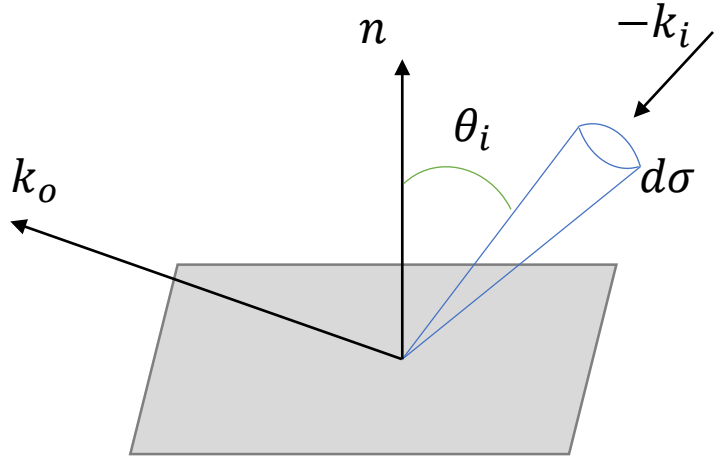
- Examples
  - Measured (or simulated) data
    - Tabularized or fitted with basis functions
  - Phenomenological (현상학적) models
    - Functional forms with controllable user-determined parameters
    - Parameters should be intuitive so that a user can easily control them (e.g., roughness).

# Outline

- Backgrounds: Light, Irradiance, Radiance, BRDF
- Light Transport Equation
- Background: Monte Carlo Integration
- Monte Carlo Ray Tracing: Distributed Ray Tracing

# Rendering Equation

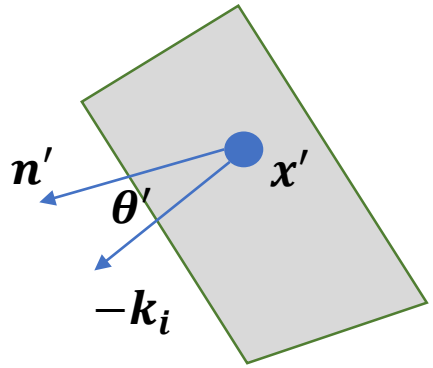
- $L_s(k_o) = \int_{all\ k_i} \rho(k_i, k_o) L_f(k_i) cos\theta_i d\sigma_i$
  - $L_f(k_i)$: field radiance from $k_i$ direction
  - $L_s(k_o)$: surface radiance measured in $k_o$ direction
  - *Rendering equation* [Immel, Cohen & Greenberg, 1986]

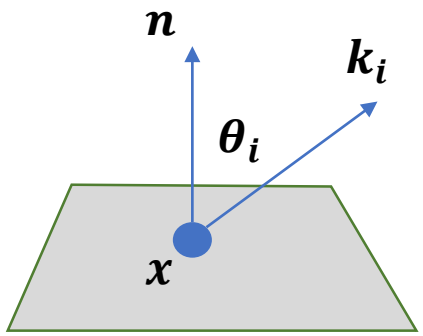



  - We can also write the equation with surface radiances only [Kajiya, 1986]

# Rendering Equation

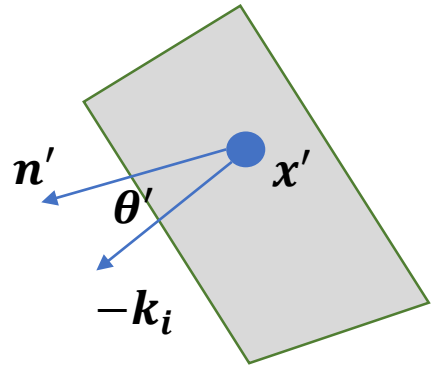
- $L_s(k_o) = \int_{all\ k_i} \rho(k_i, k_o) L_f(k_i) cos\theta_i d\sigma_i$
- $L_s(-k_i) = L_f(k_i)$
- Solid angle subtended by the point x'
  - Area on a unit sphere
  - $\Delta\sigma_i = \frac{\Delta A' cos\theta'}{||x - x'||^2}$
  - $\Delta A'$ is the area associated with x'
- Differential solid angle
  - $d\sigma_i = \frac{dA' cos\theta'}{||x - x'||^2}$

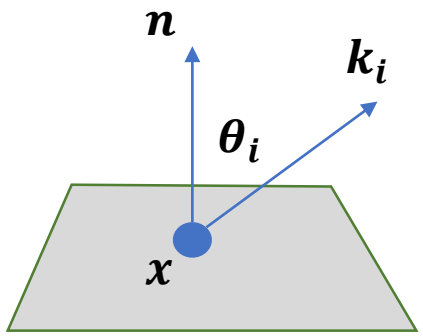


# Rendering Equation

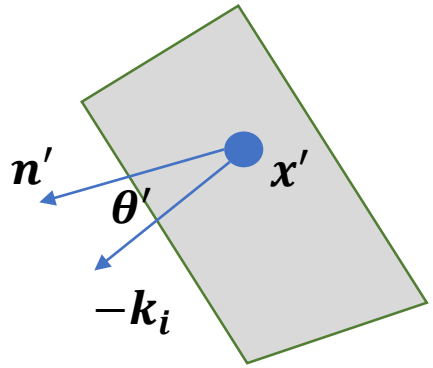
- $L_s(k_o) = \int_{all\ x' visible\ to\ x} \frac{\rho(k_i,k_o)L_s(x',x-x')cos\theta_i cos\theta'}{||x-x'||^2} dA'$

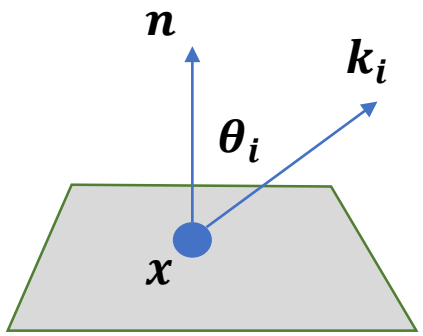


# Rendering Equation

- $L_s(k_o) = \int_{all\ x'} \frac{\rho(k_i,k_o)L_s(x',x-x')v(x,x')cos\theta_i cos\theta'}{||x-x'||^2} dA'$
  - Surface-based rendering equation [Kajiya, 1986]
  - $v(x,x')$: visibility function
  - 1 if x and x' are mutually visible
  - 0 otherwise



# Rendering Equation

- $L_s(k_o) = \int_{all\ k_i} \rho(k_i, k_o) L_f(k_i) cos\theta_i d\sigma_i$

- $L_s(k_o) = \int_{all\ x'} \frac{\rho(k_i,k_o) L_s(x', x-x') v(x,x') cos\theta_i cos\theta'}{||x-x'||^2} dA'$

- Discussion
  - Both equations are the same in theory.
  - When those are approximated via a sampling method (Monte Carlo ray tracing), the sampling domain is different.
    - e.g., Generating a ray with $k_i$ or toward $x'$

# Outline

- Backgrounds: Light, Irradiance, Radiance, BRDF
- Light Transport Equation
- Background: Monte Carlo Integration
- Monte Carlo Ray Tracing: Distributed Ray Tracing

# Background: Probability

- $Probability(x \in [a,b]) = \int_a^b p(x)\,dx$
  - $p(x)$ is the probability density function (pdf)
  - $p(x) \geq 0$
  - $\int_{-\infty}^{+\infty} p(x)\,dx = 1$

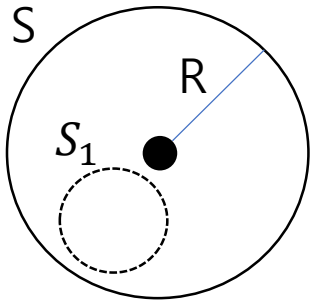
- Example: a random variable $\xi$ with uniform probability
  - $q(\xi) = \begin{matrix} 1 & if\ 0 \leq \xi < 1 \\ 0 & otherwise \end{matrix}$
  - $Probability(a \leq \xi \leq b) = \int_a^b 1\,dx = b - a,$
  - where $[a,b] \in [0,1)$

# Background: Expected Value

- Expected value $E\big(f(x)\big)$
  - $\int f(x)p(x)dx$
    - $x$ is the random variable with a pdf p(x)
  - $\sum_{i=1}^{\infty} f(x_i)p(x_i)$
- Linearity:
  - $E(x+y) = E(x) + E(y)$
  - $E\big(f(x)+g(y)\big) = E\big(f(x)\big) + E\big(g(y)\big)$
    - A function of a random variable is also a random variable
    - This holds for the random variables, which are both independent and dependent

# Background: Multidimensional Random Variables

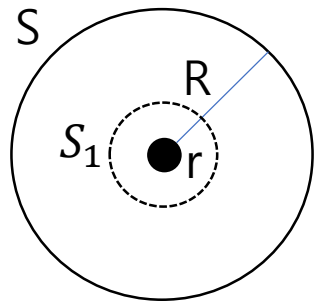
- Pdf $p: S \rightarrow \mathbb{R}$
  - S: a space has a measure $\mu$
  - e.g., S: surface of a sphere, $\mu$: area
- x is a random variable with $x \sim p$
- Probability that x will take from a region $S_i \in S$
  - $Probability(x \in S_i) = \int_{S_i} p(x) d\mu$
- e.g., a uniformly distributed random variable $\alpha \in S$
  - $p(\alpha) = \frac{1}{\pi R^2}$
- e.g., probability that $\alpha$ is in a region $S_1 \subset S$
  - $Probability(\alpha \in S_1) = \int_{S_1} \frac{1}{\pi R^2} dA$



# Background: Multidimensional Random Variables

- e.g., given a uniformly distributed random variable $\alpha \in S$,

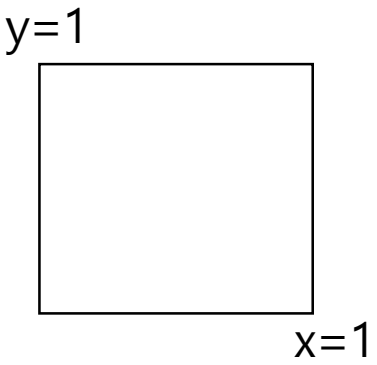
what's probability that $\alpha$ is in a region $S_1 \subset S$?




- $\alpha = (r, \phi)$ in polar coordinates
- $r = \frac{R}{2}$
- Differential area $dA = r\ dr\ d\phi$
- $Probability(\alpha \in S_1) = \int_0^{2\pi} \int_0^{\frac{R}{2}} \frac{1}{\pi R^2} r\ dr\ d\phi = 0.25$

# Background: Multidimensional Random Variables

- e.g., what's the expected value of the x coordinate:
  - $p(x, y) = 4xy$
  - $S = [0,1] \times [0,1]$




  - $E(x) = \int_S f(x, y)p(x, y)dA$
  - $= \int_0^1 \int_0^1 4x^2 y dx dy \qquad (\because f(x, y) = x)$
  - $= \frac{2}{3}$

# Background: Variance

- Variance of a random variable x
  - $V(x) \equiv E([x - E(x)]^2) = E(x^2) - [E(x)]^2$

- Standard deviation
  - $\sigma(x) = \sqrt{V(x)}$

- Properties:
  - $V(ax) = a^2 V(x)$
  - $V(x + a) = V(x)$
  - $V(ax + by) = a^2 V(x) + b^2 V(y) + 2abCov(X, Y)$
    - If x and y are independent, $Cov(X, Y) = 0$

# Background: Estimated Mean

- Suppose independent and identically distributed (iid) random variables
  - Random variables $x_i$ are independent and have a common pdf p
  - Estimated mean = an estimate of the expectation $E(x)$
    - $E(x) \approx \frac{1}{N}\sum_{i=1}^{N} x_i$
    - As N increases, the variance of this estimate decreases. Why?
      - $V(x_i) = \sigma^2$ ($\because$ identically distributed)
      - $V\left(\frac{1}{N}\sum_{i=1}^{N} x_i\right) = \frac{1}{N^2}\sum_{i=1}^{N} V(x_i) = \frac{1}{N}\sigma^2$ ($\because$ independent)
  - Law of Large Numbers
    - $Probability\left[E(x) = \lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} x_i\right] = 1$

# Monte Carlo Integration

- Given a function $f: S \rightarrow \mathbb{R}$, a random variable $x \sim p$
  - The expected value of $f(x)$
    - $E\big(f(x)\big) = \int_{x \in S} f(x)p(x)d\mu \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)$
  - By substituting $g = fp$,
    - $E\big(f(x)\big) = \int_{x \in S} g(x)d\mu \approx \frac{1}{N}\sum_{i=1}^{N} \frac{g(x_i)}{p(x_i)}$
    - $p$ should be positive for nonzero $g$
  - How can we apply this technique to the transport equation?
    - $L_s(k_o) = \int_{all\ x'} \frac{\rho(k_i,k_o)L_s(x',x-x')v(x,x')cos\theta_i cos\theta'}{||x-x'||^2} dA'$

# Comparisons of Estimators

- Monte Carlo ray tracing (together with other image processing) is simply a pixel estimator, which produces an estimate $\hat{\mu}_c$ of the unknown $\mu_c$ at pixel $c$.
- Actual error: $\|\hat{\mu}_c - \mu_c\|^2$
  - Knowing the actual error is equivalent to knowing the unknown.
  - Used only when the unknown is available.
- MSE: $E[\|\hat{\mu}_c - \mu_c\|^2] = V(\hat{\mu}_c) + bias^2(\hat{\mu}_c)$
  - Widely used for optimizing the parameters in the estimator.
  - It requires estimation of the (unknown) MSE.

# Comparisons of Estimators

- MSE: $E[\|\hat{\mu}_c - \mu_c\|^2] = V(\hat{\mu}_c) + bias^2(\hat{\mu}_c)$
  - Widely used for optimizing the parameters in the estimator.
  - It requires an estimation of the (unknown) MSE.
- Unbiased, $bias(\hat{\mu}_c) = 0$
- Biased, $bias(\hat{\mu}_c) \neq 0$
- Consistent
  - Zero MSE with an infinite sample count
- e.g.
  - path tracing: Unbiased and consistent
  - Progressive photon mapping: Biased but consistent

# Outline

- Backgrounds: Light, Irradiance, Radiance, BRDF
- Light Transport Equation
- Background: Monte Carlo Integration
- Monte Carlo Ray Tracing: Distributed Ray Tracing

# Sampling

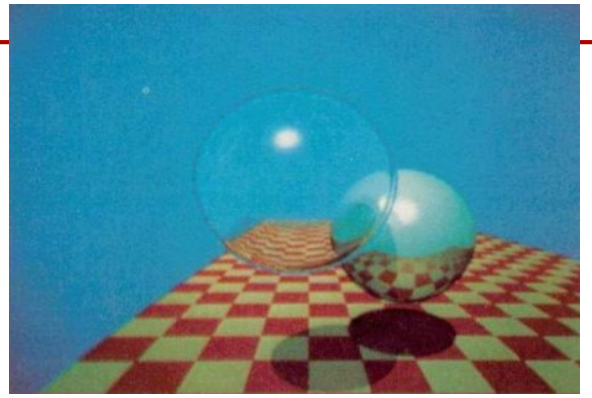
- To solve the MC integration, we need to draw random samples. This process is called sampling.

# Review: Distributed Ray Tracing

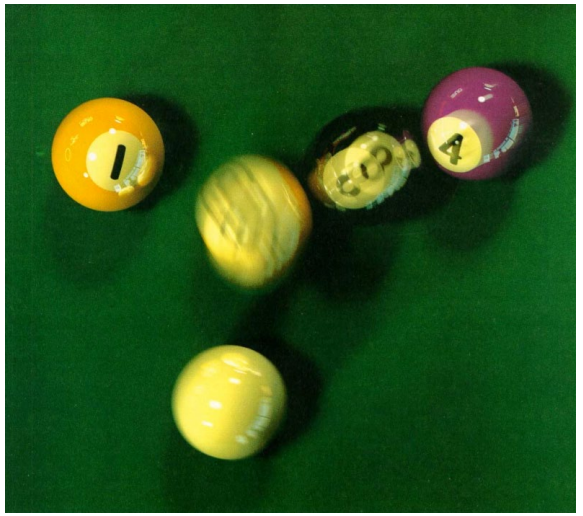
- Whitted-style ray tracing produces very clean images (look fake)
  - Perfect focus
  - Perfect reflections
  - Sharp shadows

- Distributed Ray Tracing [Cook et al. 1984]
  - A Monte Carlo ray tracing
  - The main idea is to replace the single ray with a distribution of rays

- Add randomness to rendering
  - Antialiasing
  - Soft shadows
  - Depth-of-field
  - Motion blur
  - Glossy reflections



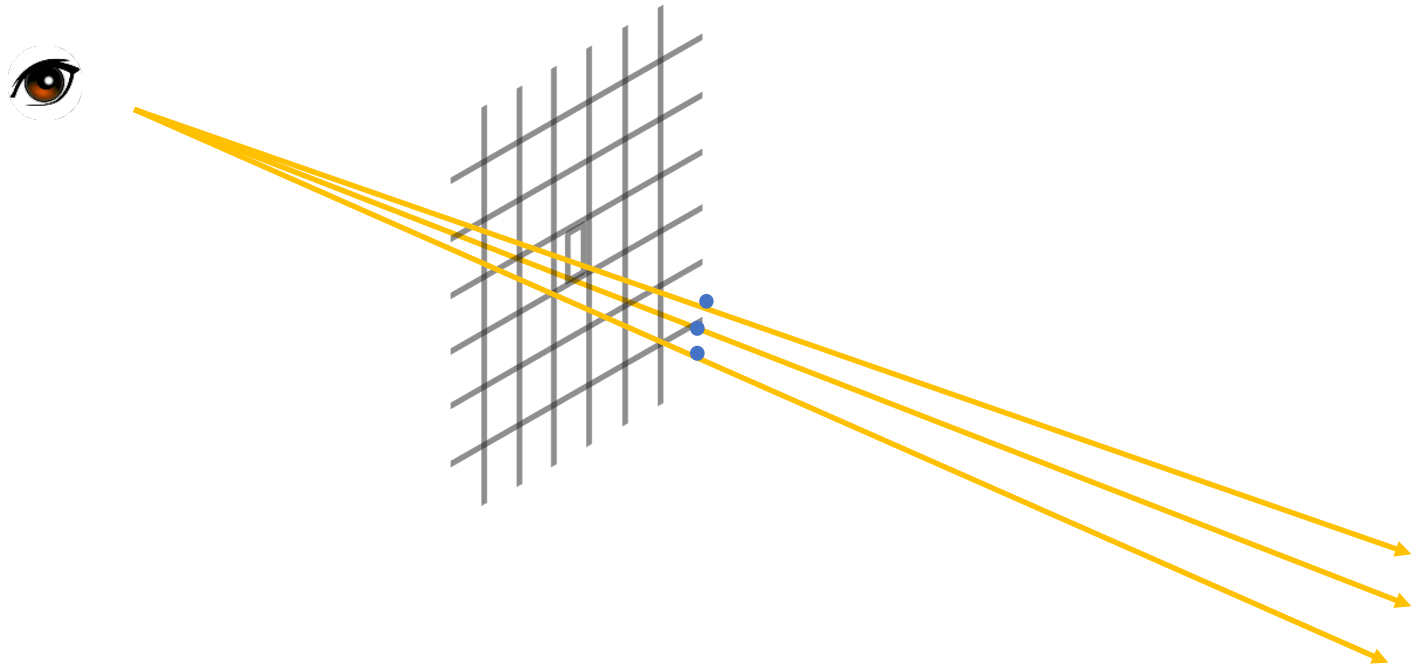
[Whitted 1980]



[Cook et al. 1984]

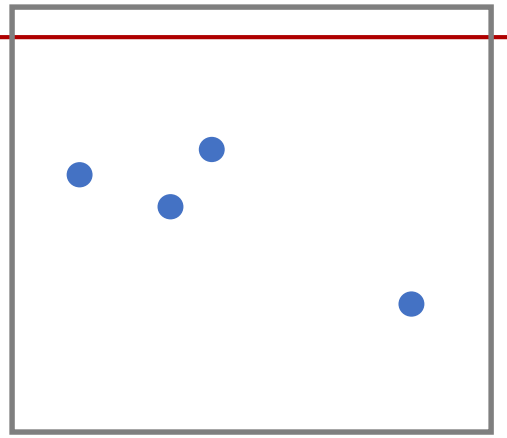
# Review: Antialiasing

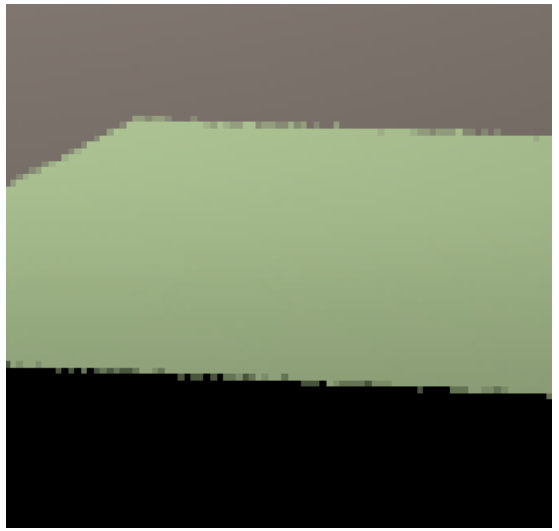
- To reduce image aliasing, we need to compute a pixel color by averaging multiple samples, instead of taking a ray sample only at the center point



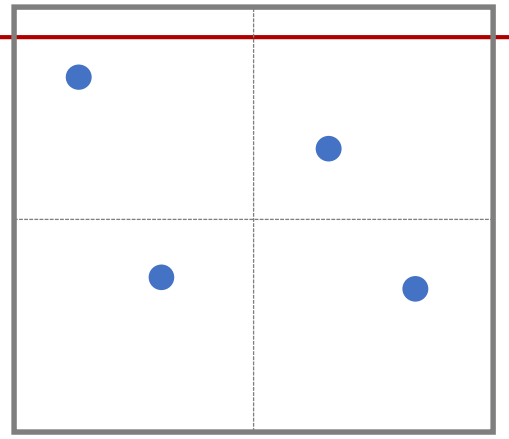
# Review: Antialiasing

- e.g. four samples / pixel
- Random sampling: randomly generate $n^2$ rays
- for each pixel (x, y) do
  - $c(x, y) = 0$
  - for p = 0 to $n^2 - 1$ do
    - $c(x, y) = c(x, y) + trace(x + \epsilon_1, y + \epsilon_2)$
  - $c(x, y) = c(x, y)/n^2$
- $\epsilon \in [0,1)$ is a random number
- The regular pattern is converted into image noise

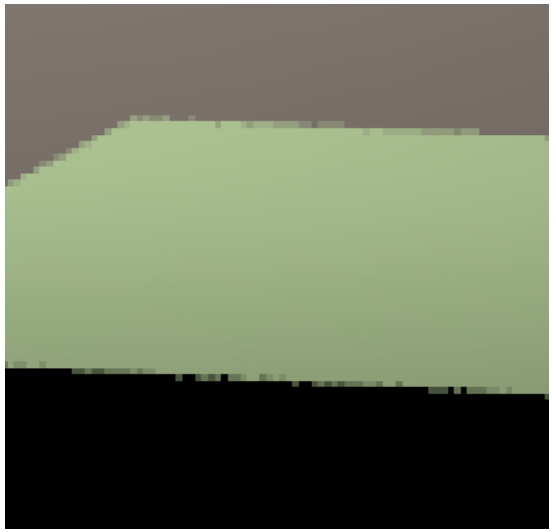




# Review: Antialiasing

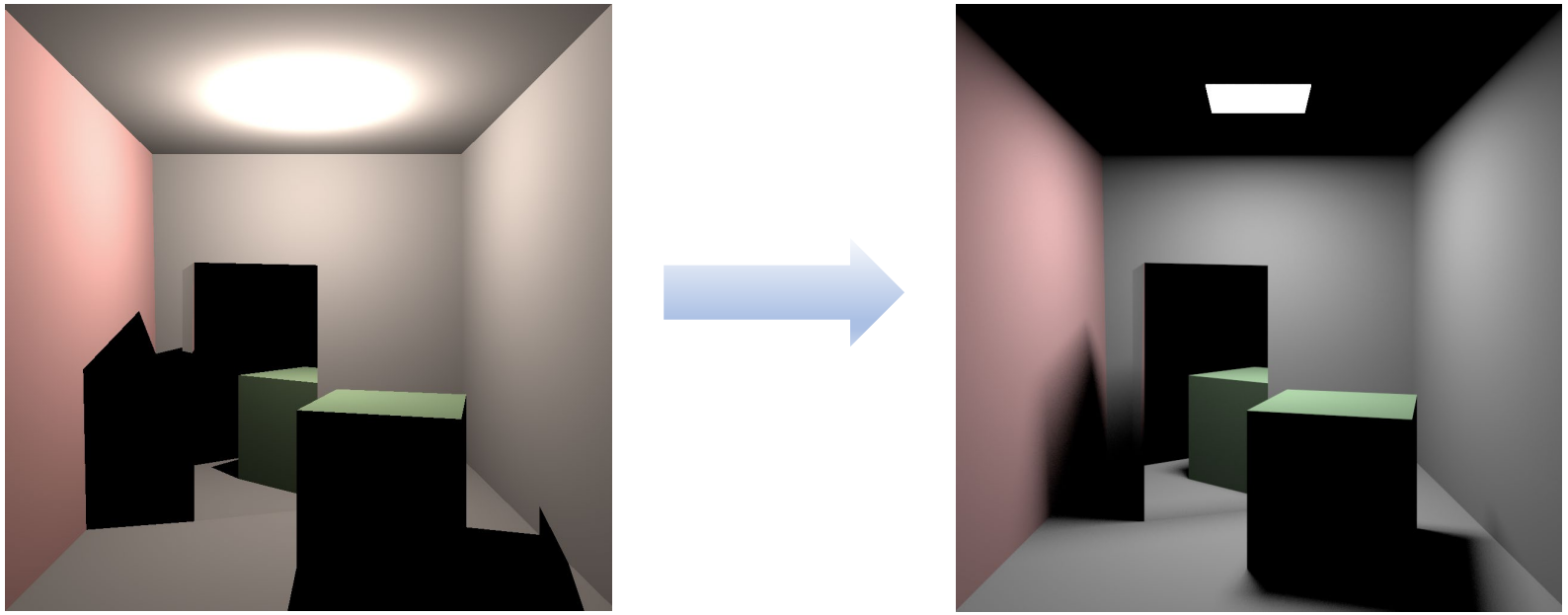


- e.g. four samples / pixel
- Jittering (stratified sampling): randomly generate a ray within each grid
- for each pixel (x, y) do
  - $c(x, y) = 0$
  - for p = 0 to n – 1 do
    - for q = 0 to n – 1 do
      - $c(x, y) = c(x, y) + trace\left(x + \frac{p+\epsilon_1}{n}, y + \frac{q+\epsilon_2}{n}\right)$
    - $c(x, y) = c(x, y)/n^2$



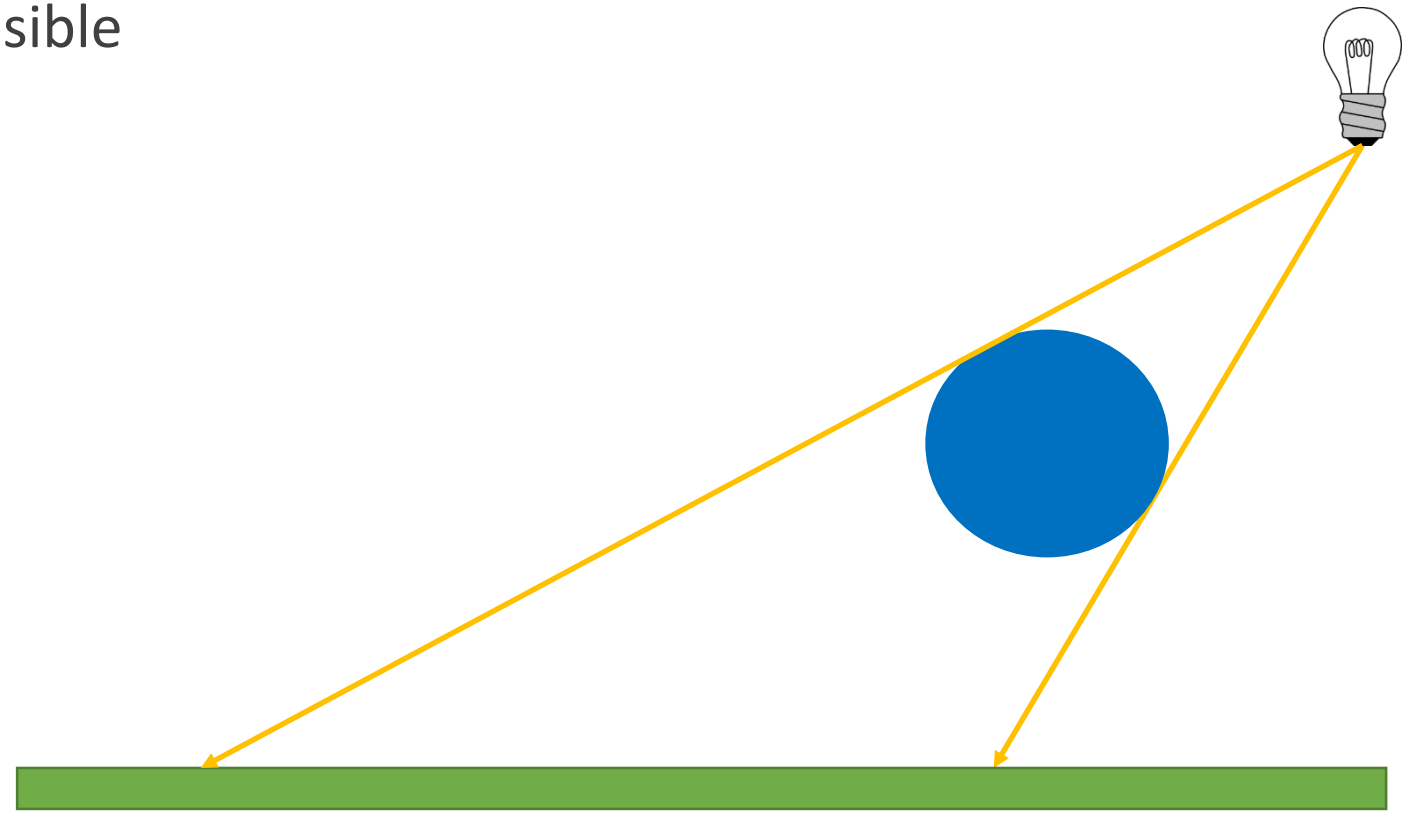
- This is a hybrid approach between the regular and random sampling

# Review: Soft Shadows



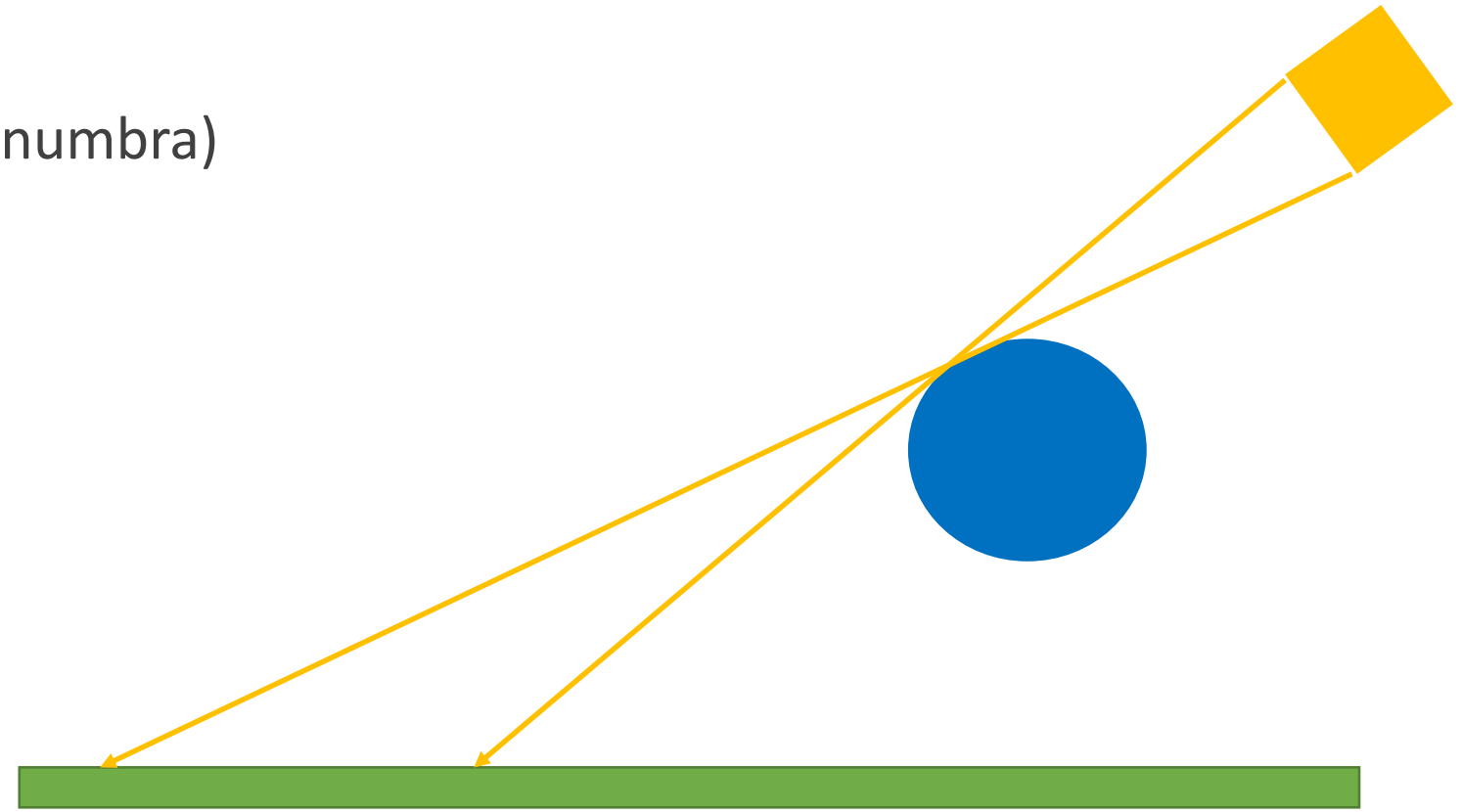
# Review: Soft Shadows

- A point light source introduces hard shadows
  - Visible or invisible



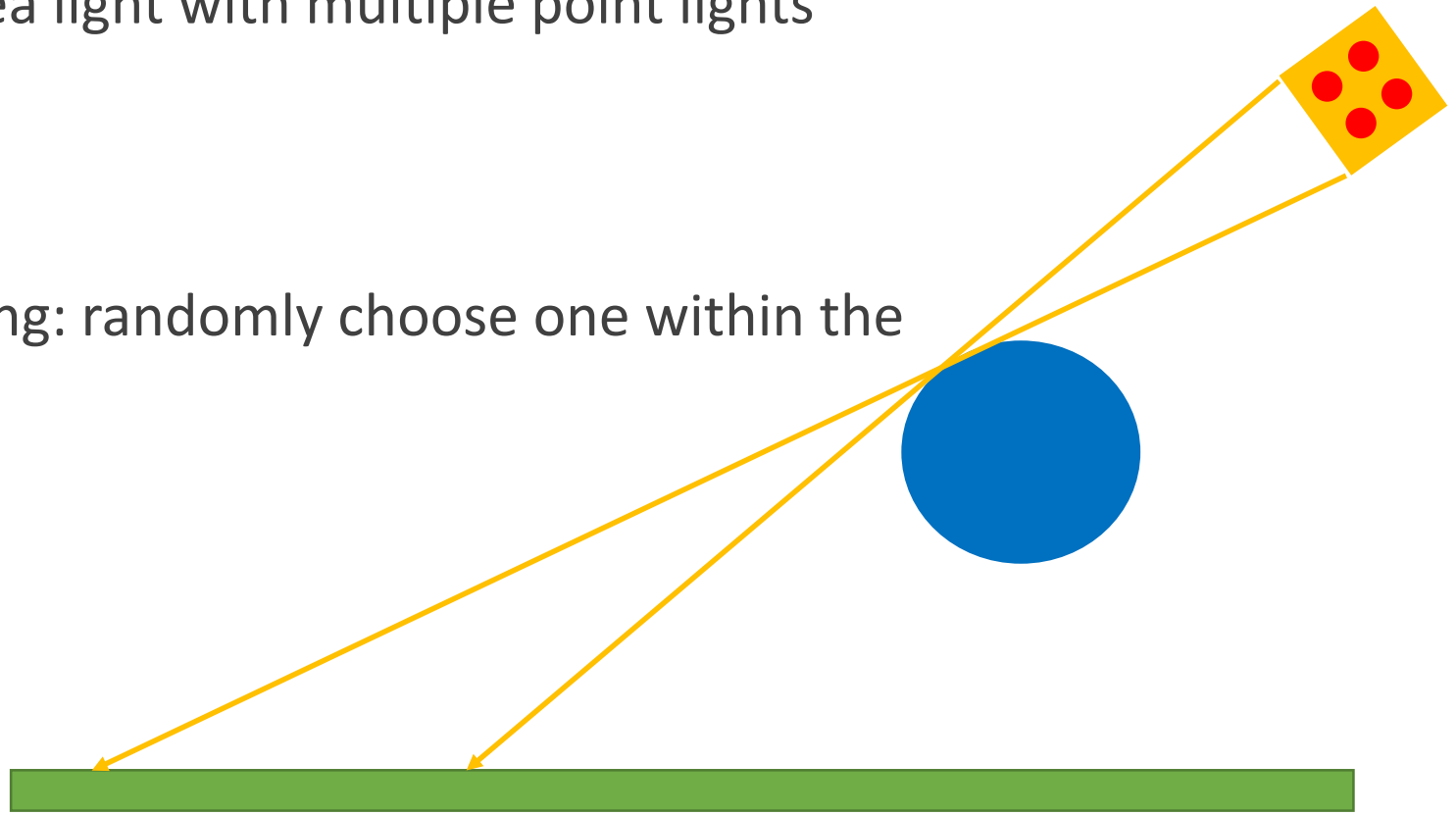
# Review: Soft Shadows

- An area light source introduces soft shadows
  - Visible
  - Partially visible (penumbra)
  - Invisible



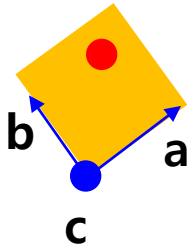
# Review: Soft Shadows

- How can we implement the soft shadows?
  - Approximate the area light with multiple point lights (regularly chosen)
    - Produce regular patterns
  - Distributed ray tracing: randomly choose one within the light shape

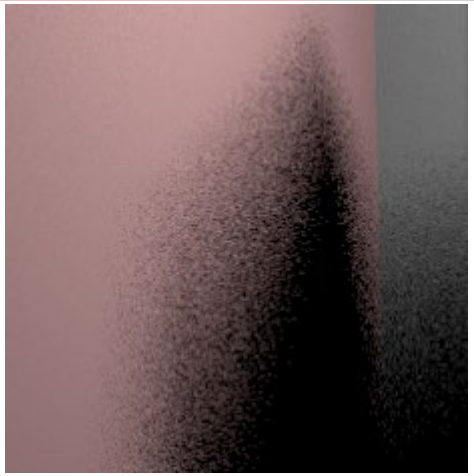


# Review: Soft Shadows

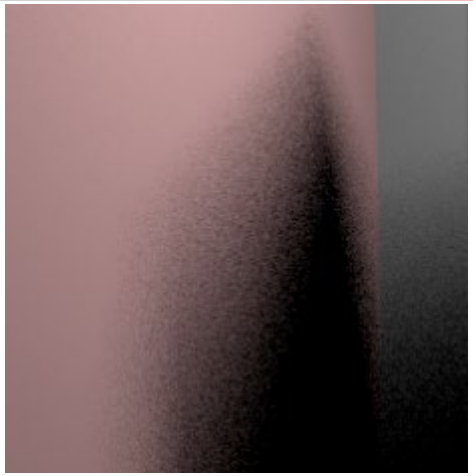
- e.g. area light defined as a parallelogram
  - Select a random point
    - $\boldsymbol{l_{pos}} = \boldsymbol{c} + \varepsilon_1 \boldsymbol{a} + \varepsilon_2 \boldsymbol{b}$
    - $\varepsilon_1, \varepsilon_2 \in [0,1)$
  - Generate a shadow ray from this point



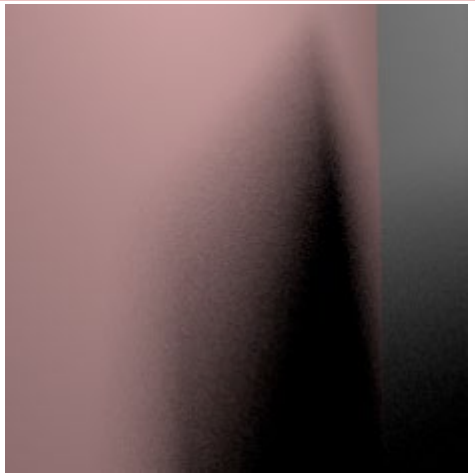
# Review: Soft Shadows



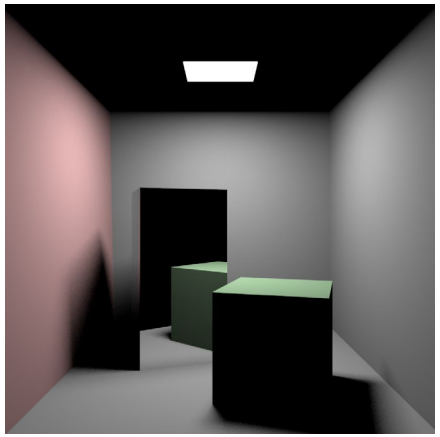
4 shadow rays / pixel



16 shadow rays / pixel

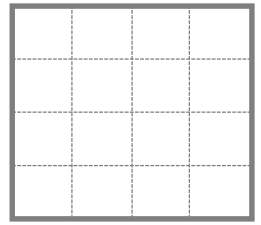


64 shadow rays / pixel

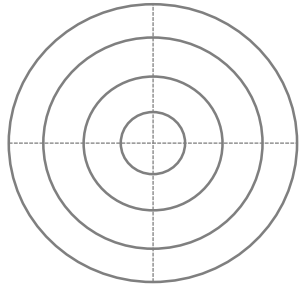


# Random Point Selection

- Problem: draw N samples from $[0,1]^2$
  - i.e., provide $(x_1, y_1), \dots, (x_N, y_N)$
  - Techniques
    - Random sampling
    - Jittering



- Problem: draw N samples from a disk shape (a camera lens)
  - Randomly select $u_i, v_i$ from uniform [0,1]
  - $\phi_i = 2\pi u_i$
  - $r_i = v_i R$
  - Note
    - This will cover all the area of a circle with a radius R
    - The downside is that the selected points will be distributed non-uniformly



# Inverse Transform Sampling

- 1D density function $f(x)$ with $x \in [x_{min}, x_{max}]$
- Q. Can we generate random numbers $\alpha_i$ so that they can have density $f(x)$ using a set of uniform random numbers $\xi_i \in [0,1]$?

- Cumulative probability distribution function $F(x)$:
  - $P(\alpha < x) = F(x) = \int_{x_{min}}^{x} f(x')d\mu$
- $\alpha_i = F^{-1}(\xi_i)$
- Example:
  - $y = x^2 \quad or\ f(x) = x^2 \quad (x > 0)$
  - $x = \sqrt{y} \quad or\ f^{-1}(x) = \sqrt{x}$

# Inverse Transform Sampling

- 1D density function $f(x)$ with $x \in [x_{min}, x_{max}]$
- Q. Can we generate random numbers $\alpha_i$ so that they can have density $f(x)$ using a set of uniform random numbers $\xi_i \in [0,1]$?
- Problem: generate random points $x_i$ that have the following density:
  - $f(x) = \frac{3x^2}{2}$ $on$ $[-1,1]$
  - 1. $F(x) = \frac{x^3+1}{2}$
  - 2. $F^{-1}(x) = \sqrt[3]{2x-1}$
  - 3. $(x_1, \dots, x_N) = \left(\sqrt[3]{2\xi_1 - 1}, \dots, \sqrt[3]{2\xi_N - 1}\right)$
  - Note
    - $\xi$ can be jittered samples

# Inverse Transform Sampling

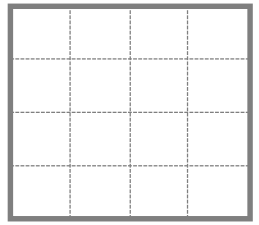
- 2D density function $f(x, y)$ with $(x, y) \in [x_{min}, x_{max}] \times [y_{min}, y_{max}]$

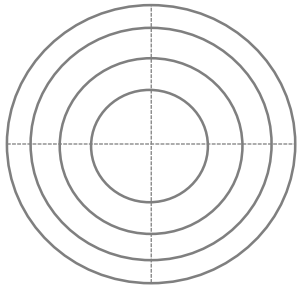
- $P(\alpha_x < x \; and \; \alpha_y < y) = F(x, y) = \int_{y_{min}}^{y} \int_{x_{min}}^{x} f(x', y') d\mu(x', y')$

- Steps.
  - Choose $x_i$ using a marginal distribution $F(x, y_{max})$
  - Choose $y_i$ according to $\frac{F(x_i, y)}{F(x_i, y_{max})}$

# Inverse Transform Sampling

- Example:
  - $P(r < r_0 \; and \; \phi < \phi_0) = F(r_0, \phi_0) = \int_0^{\phi_0} \int_0^{r_0} \frac{r dr d\phi}{\pi R^2} = \frac{\phi r^2}{2\pi R^2}$
- Procedure.
  - Generate two random numbers $\xi_1, \xi_2 \in [0,1]$
  - Choose $\phi_i$ using a marginal distribution $F(r_{max}, \phi)$
    - $F(r_{max}, \phi) = \frac{\phi R^2}{2\pi R^2}$
    - $\phi = 2\pi\xi_1$
  - Choose $r_i$ according to $\mathrm{F}(\mathrm{r}|\phi_i) = \frac{F(r,\phi_i)}{F(r_{max},\phi_i)}$
    - Similarly, $r = R\sqrt{\xi_2}$





# Rejection

- Choose some random points according to a distribution and reject some of them
- Example: draw uniform random points within the unit circle
  - Choose uniform random samples $(x, y) \in [-1,1]^2$
  - Reject the samples outside the circle
- Procedure
  - $i = 1$
  - $while\ (i < N)$
    - $x_i = -1 + 2 \times rand()$ // $rand()$ will return uniform random sample in [0, 1]
    - $y_i = -1 + 2 \times rand()$
    - $if\ (x_i^2 + y_i^2 < r)$
      - $i = i + 1$

# Rejection

- Choose some random points according to a distribution and reject some of them

- Pros
  - Very simple to code
- Cons
  - Can be very inefficient given complex scenarios