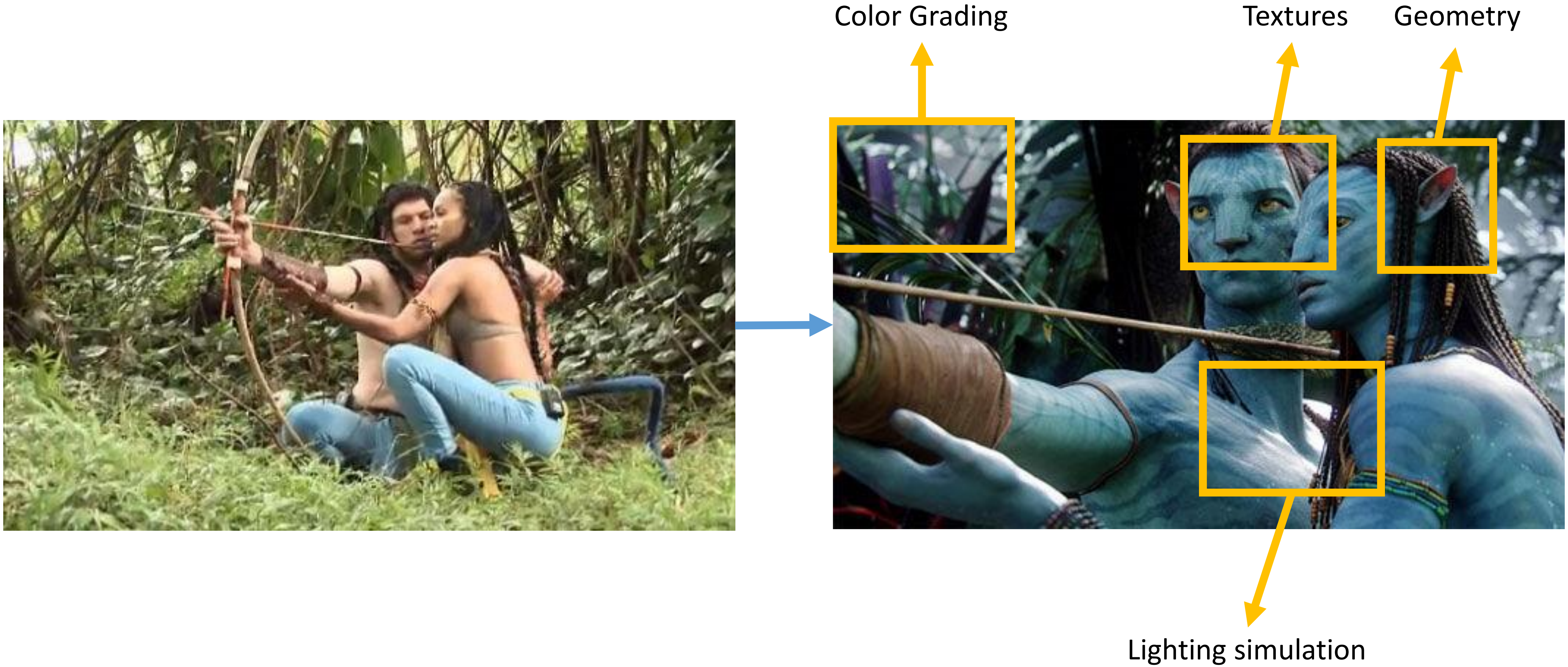
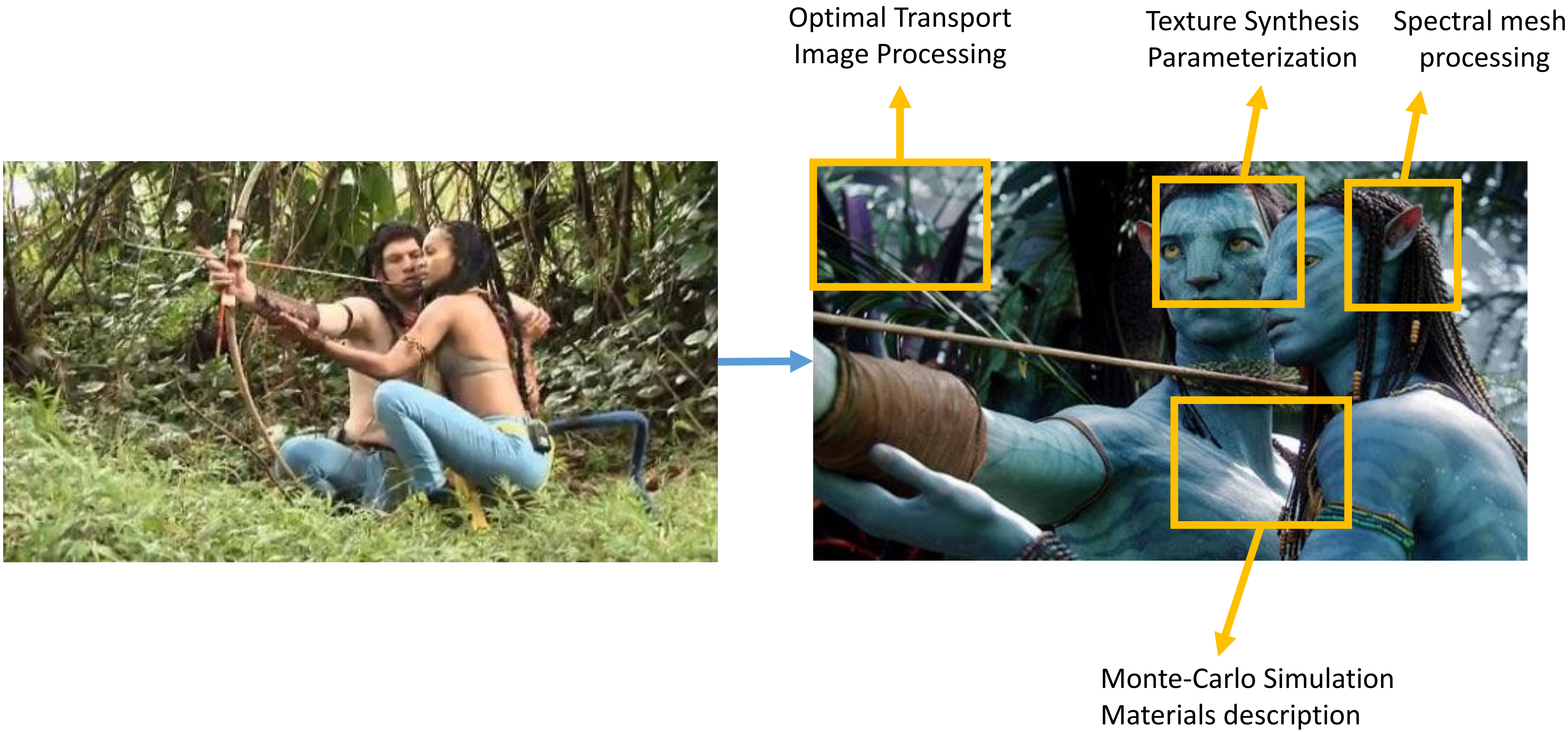
Mathematical methods for Image Synthesis

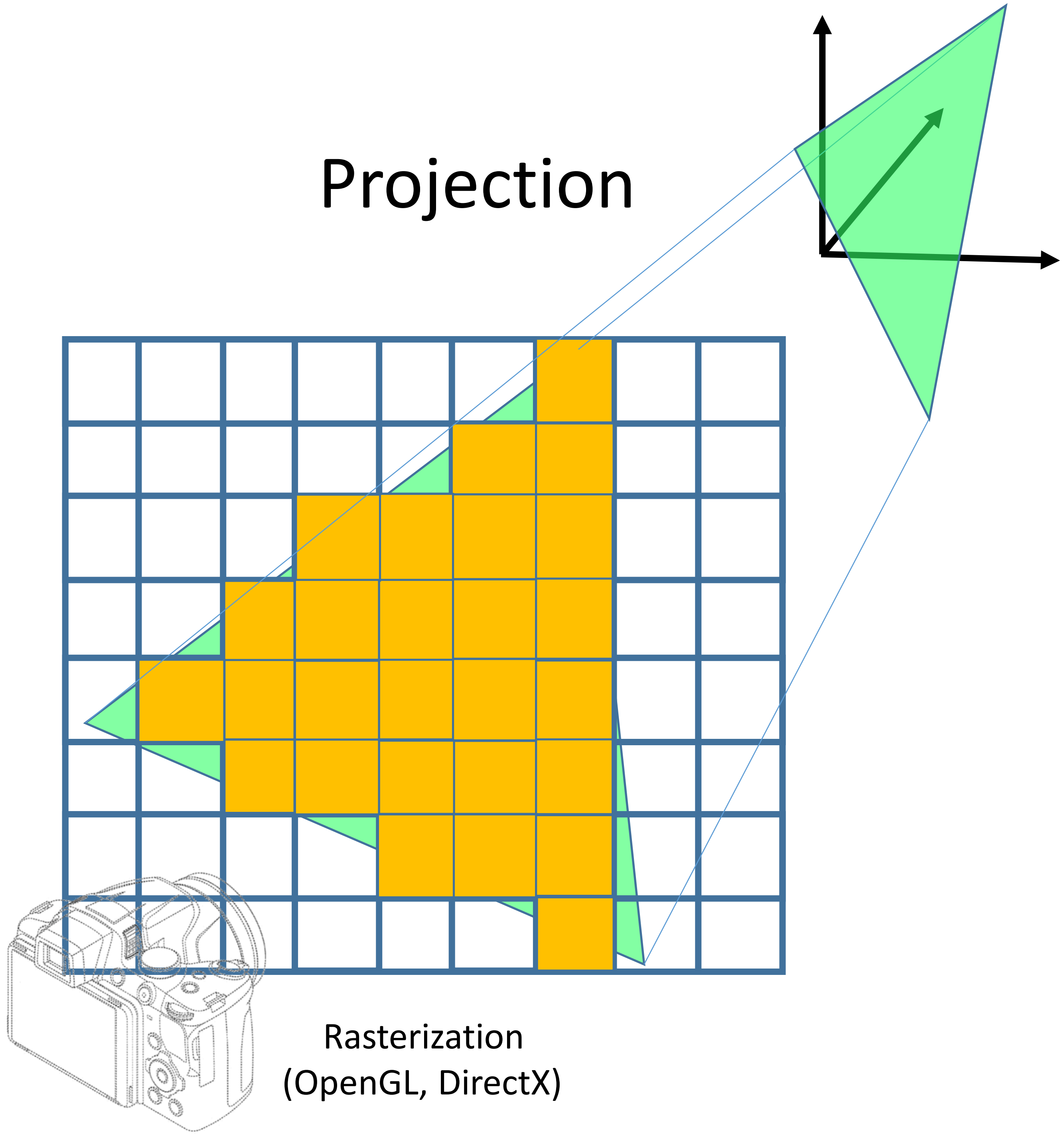
Nicolas Bonneel
Julie Digne



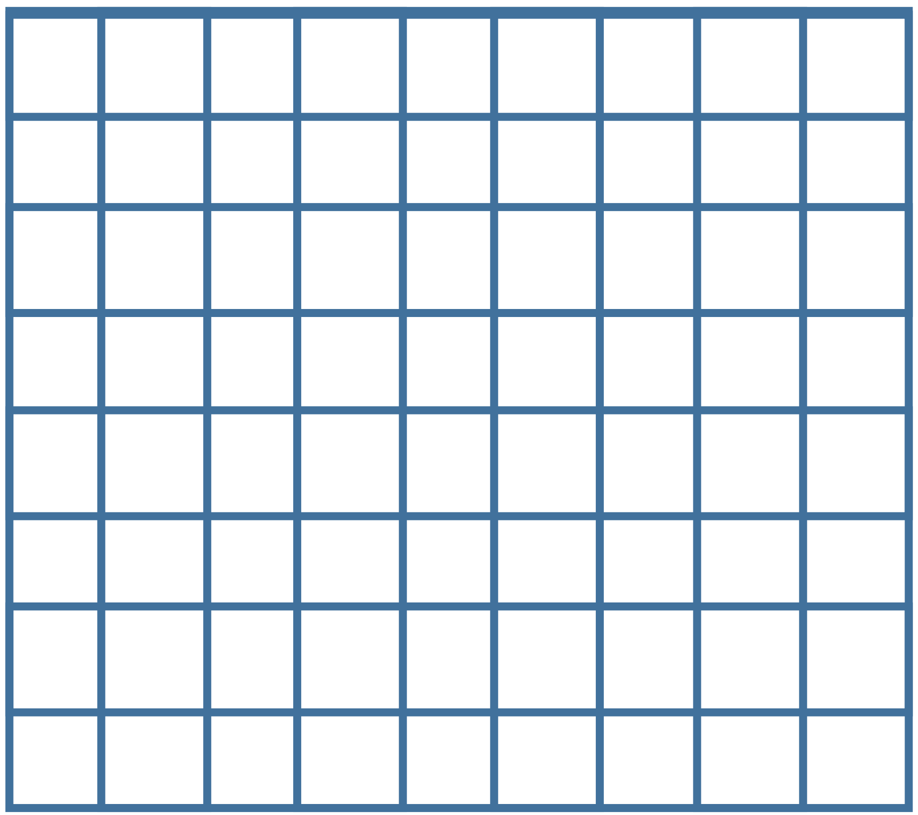
Color Grading
Textures
Geometry
Lighting simulation


Optimal Transport
Image Processing
Texture Synthesis
Parameterization
Spectral mesh
processing
Monte-Carlo Simulation
Materials description

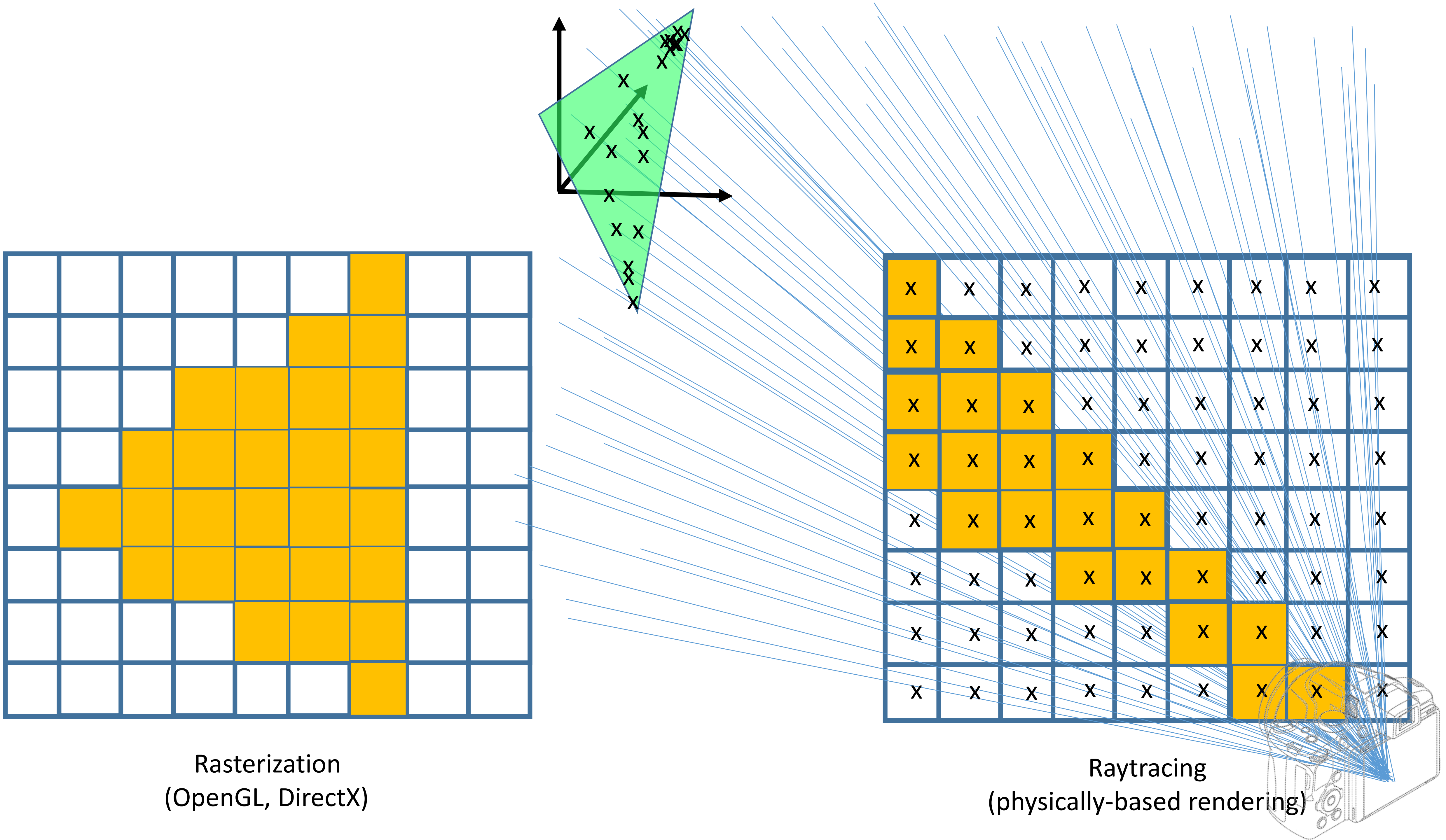
Monte-Carlo methods for 3d rendering



Rasterization
(OpenGL, DirectX)



Raytracing
(physically-based rendering)



Rasterization
(OpenGL, DirectX)

Raytracing
(physically-based rendering)

```
procedure drawTriangle(T)
    T' = Project(T)
    Rect = bounding_box(T')
    for each pixel p in Rect
        if p inside T'
            if depth(p) < z_buffer(p)
                p = color
                z_buffer(p) = depth(p)
End procedure
```

Rasterization
(OpenGL, DirectX)

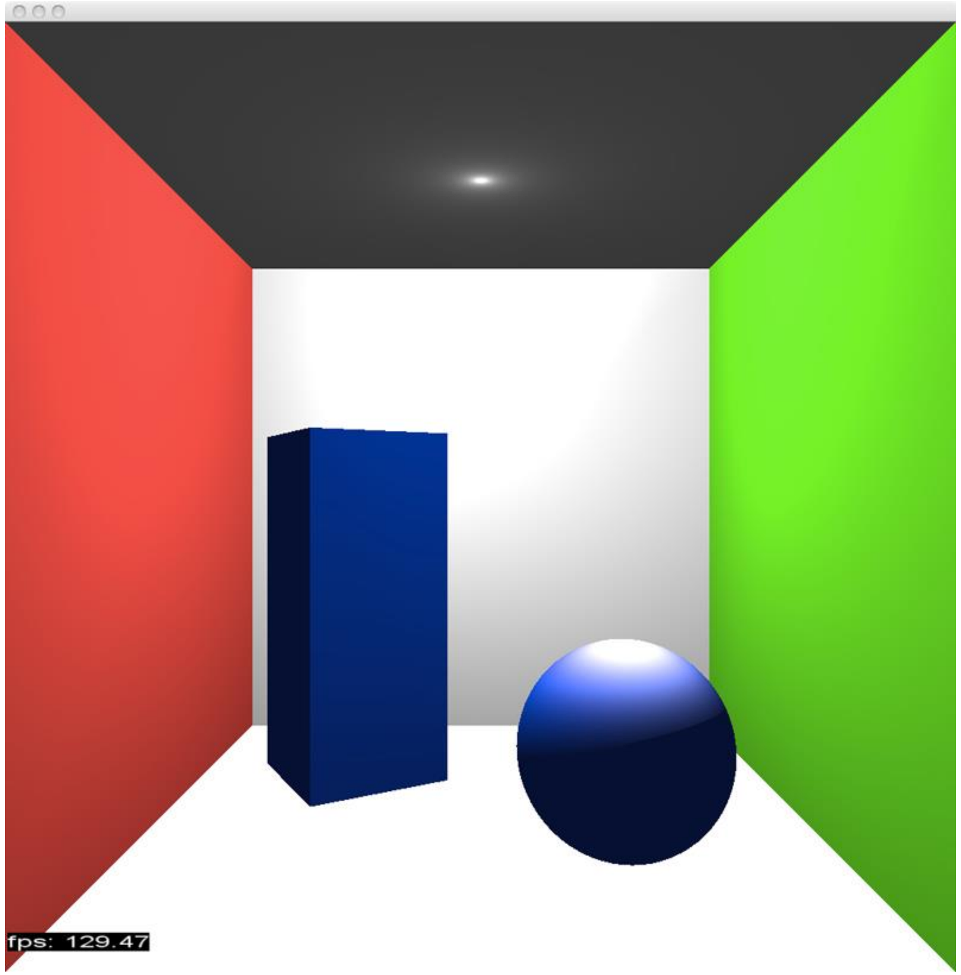
Raytracing
(physically-based rendering)

```
procedure drawTriangle(T)
    T' = Project(T)
    Rect = bounding_box(T')
    for each pixel p in Rect
        if p inside T'
            if depth(p) < z_buffer(p)
                p = color
                z_buffer(p) = depth(p)
End procedure
```

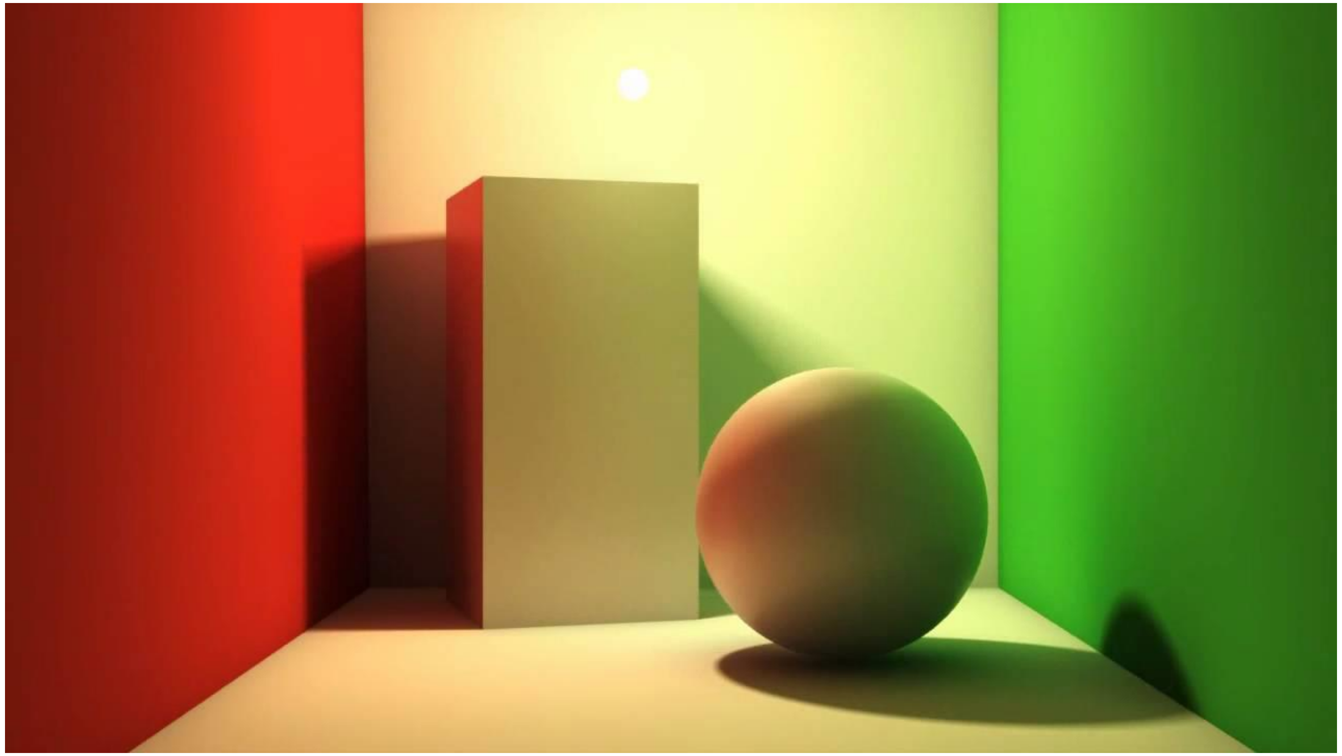
Rasterization
(OpenGL, DirectX)

```
procedure drawPixel(p)
L = Line(origin, p)
S = Ø
for each primitive T
    S = S ∪ intersect(L, T)
S = sort(S)
if S ≠ Ø
    p = color
End procedure
```

Raytracing
(physically-based rendering)

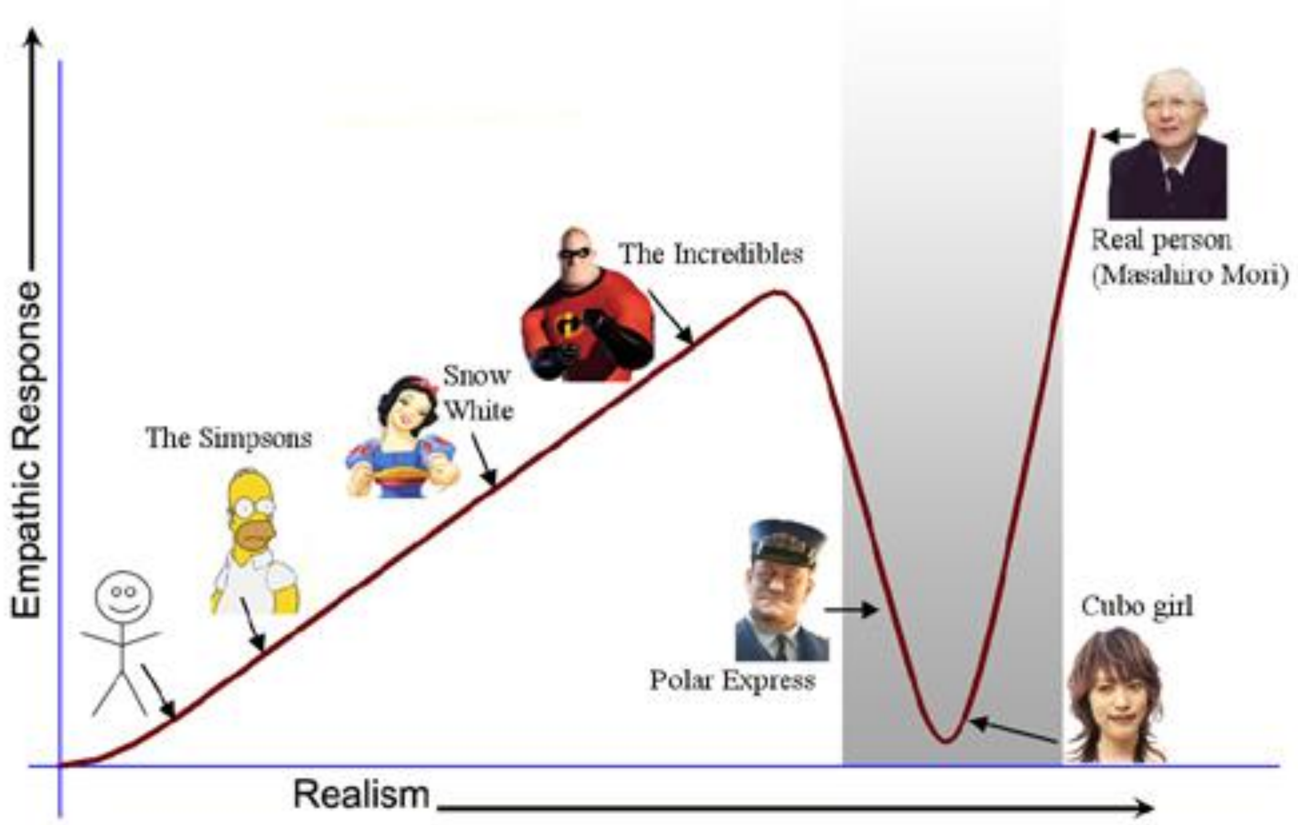


Rasterization
(OpenGL, DirectX)



Raytracing
(physically-based rendering)

Why realism can be important ?



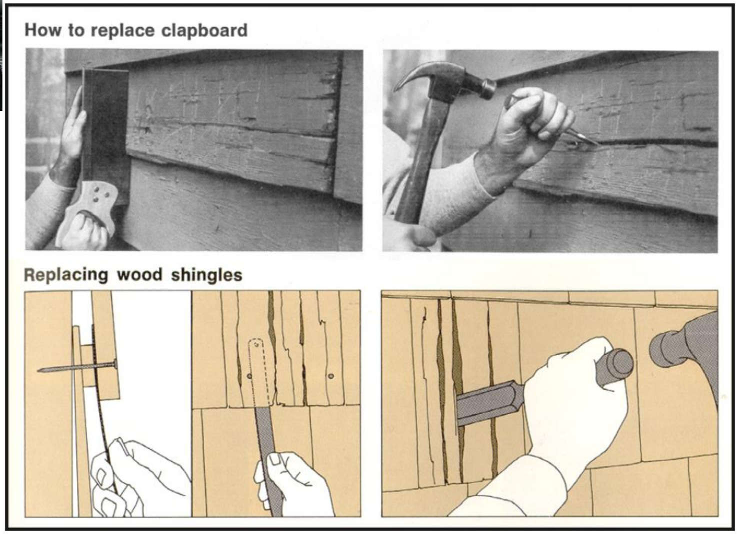
Uncanny Valley

Uncanny Valley

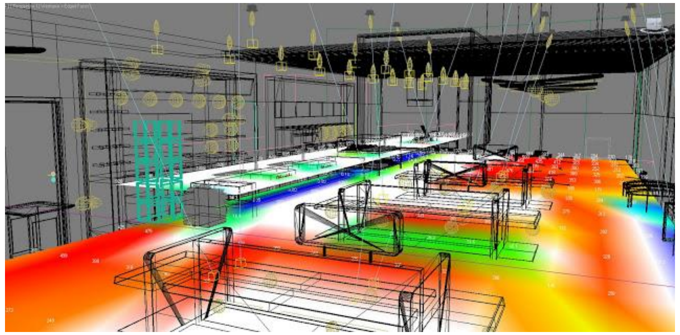
Polar Express

Tintin

What is realism ?



Functional Realism



Physical Realism
- full simulation
- useful in scientific computing
- e.g., architecture

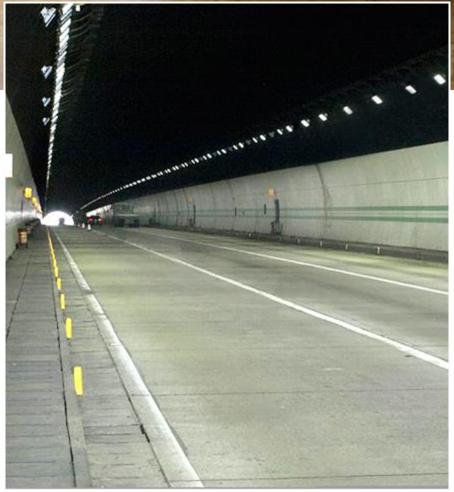


Photo-realism
- motivated by perception
- HDR, faster rendering

“Three Varieties of Realism in Computer Graphics”, Ferwerda

Limits of realism

Non-Photorealistic Rendering

Non physically realistic

Non realistic

The rendering equation

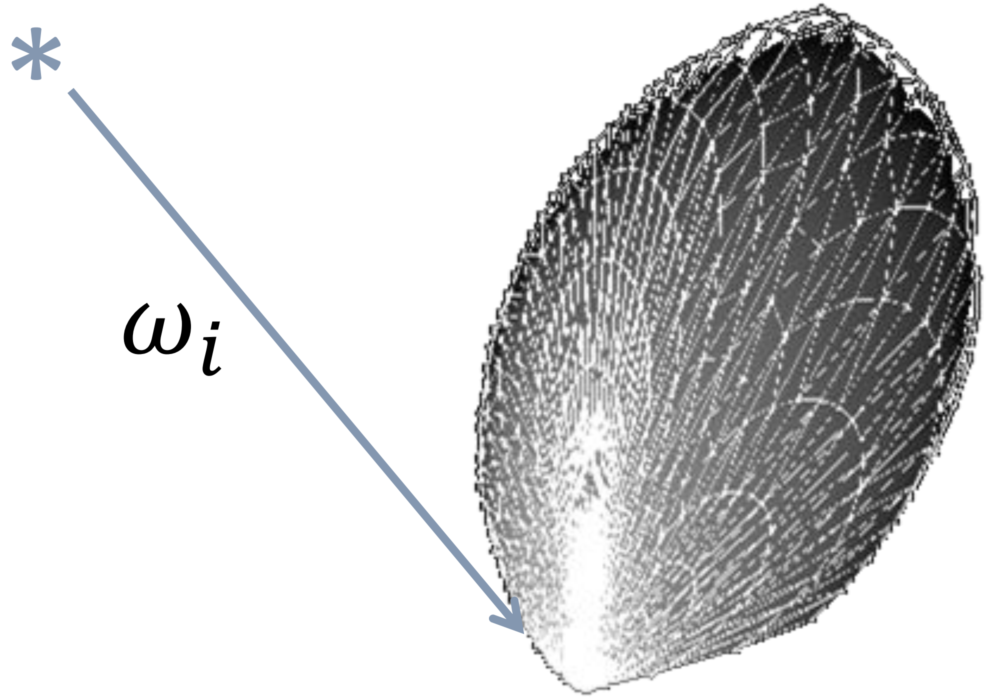
Bidirectional Reflectance Distribution Functions (BRDF)

Function $f : S^2 \times S^2 \ (\times \mathbb{R}) \to \mathbb{R}$

Describes the appearance of materials

Such that:

- $f(\omega_i, \omega_o) = f(\omega_o, \omega_i)$ (Helmoltz's reciprocity)
- $f(\omega_i, \omega_o) \geq 0$
- $\int_\Omega f(\omega_i, \omega_o)\, \omega_i^y \, d\omega_i \leq 1$



Bidirectional Reflectance Distribution Functions (BRDF)

Diffuse (Lambertian) BRDF.

$$f(\omega_i, \omega_o) = \frac{\rho}{\pi}$$

Specular BRDF.

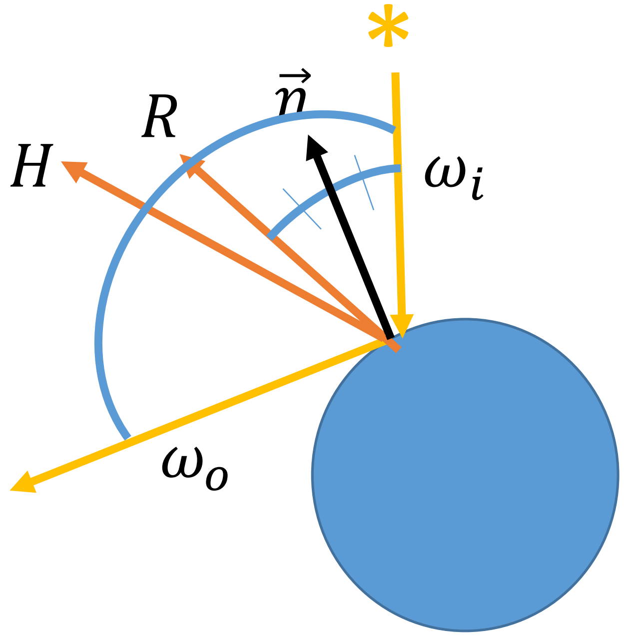
$$f(\omega_i, \omega_o) = \delta(R - \omega_o)$$

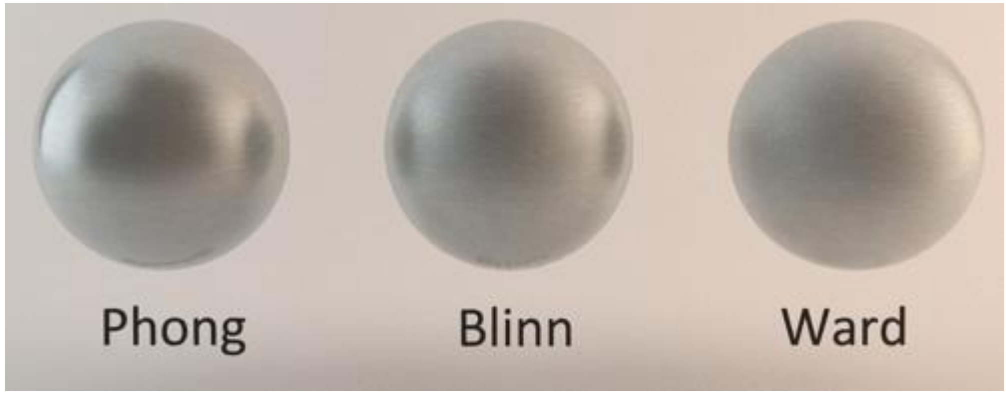
Bidirectional Reflectance Distribution Functions (BRDF)

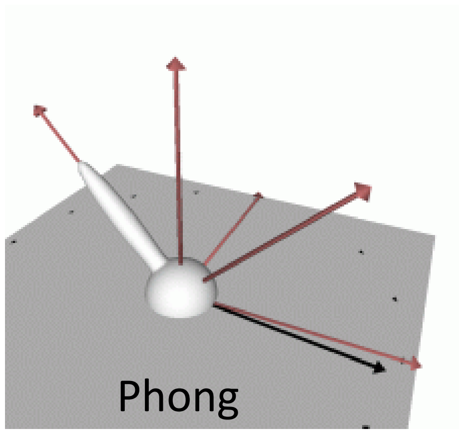
Glossy BRDF
$f(\omega_i, \omega_o) = \cdots$ many options

Bidirectional Reflectance Distribution Functions (BRDF)

- Analytical models
  - Phenomenological/Experimental models :
    - Phong $f(\omega_i, \omega_o) = \langle \omega_o, R \rangle^{\alpha}$
    - Blinn $f(\omega_i, \omega_o) = \langle \vec{n}, H \rangle^{\alpha}$
    - Ward $f(\omega_i, \omega_o) = \exp\left(-\tan^2 \frac{\langle \vec{n}, H \rangle}{\alpha^2}\right) \Big/ 4\pi\alpha^2 \sqrt{\langle \vec{n}, \omega_o \rangle \langle \vec{n}, \omega_i \rangle}$
    - Lafortune, Minnaert, Strauss, Lewis, Schlick, ...







Bidirectional Reflectance Distribution Functions (BRDF)

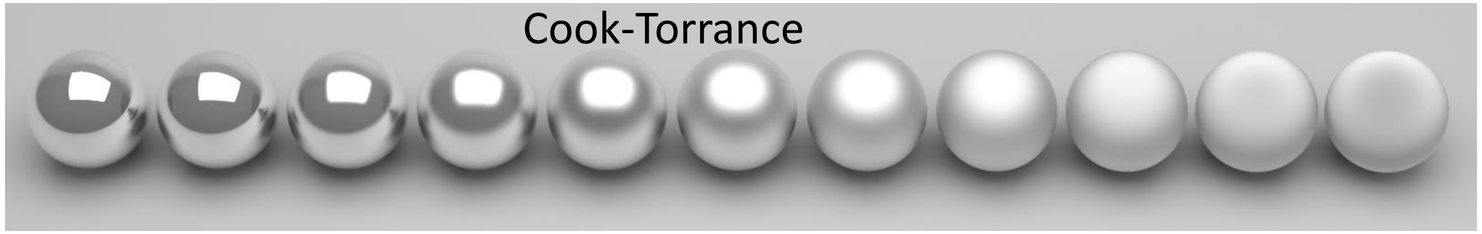
- Analytical models
  - Physical models (microfacets):
    - Ashikhmin-Shirley
    - Cook-Torrance
    - Poulin-Fournier, Torrance-Sparrow, Oren-Nayar, Kajiya, …

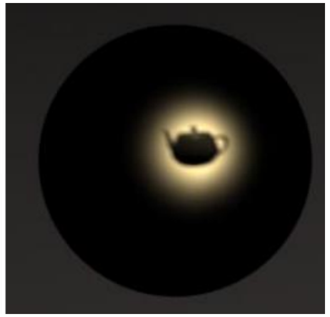
Distribution of normals (~Gaussian)

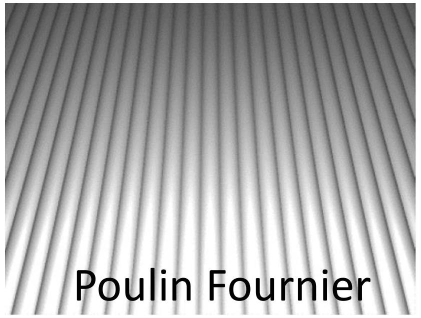
Fresnel

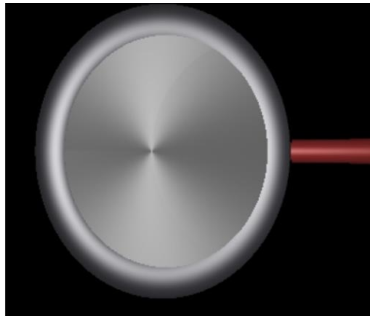
Shadowing and masking
(can be related to D)

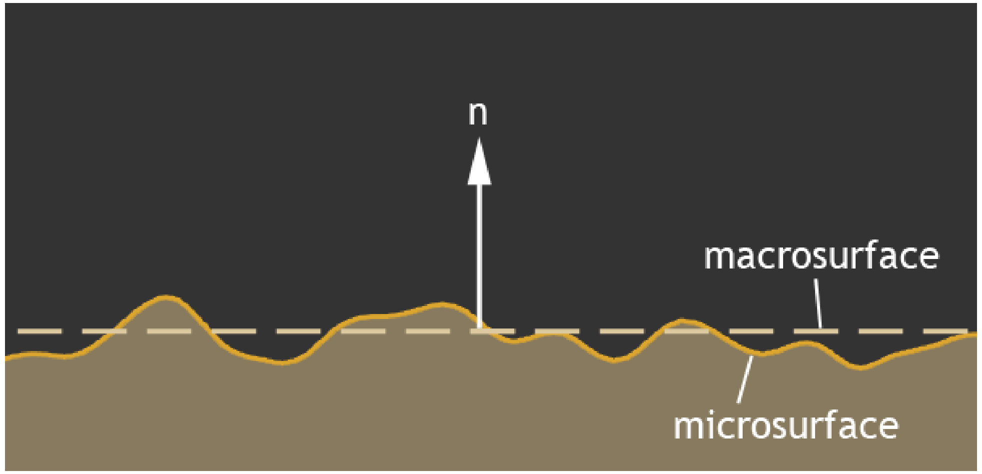
$$f(\omega_i, \omega_o) = \frac{F(\beta)}{\pi} \frac{D(H)G(\omega_i, \omega_o)}{\cos(\omega_o)\ \cos(\omega_i)}$$





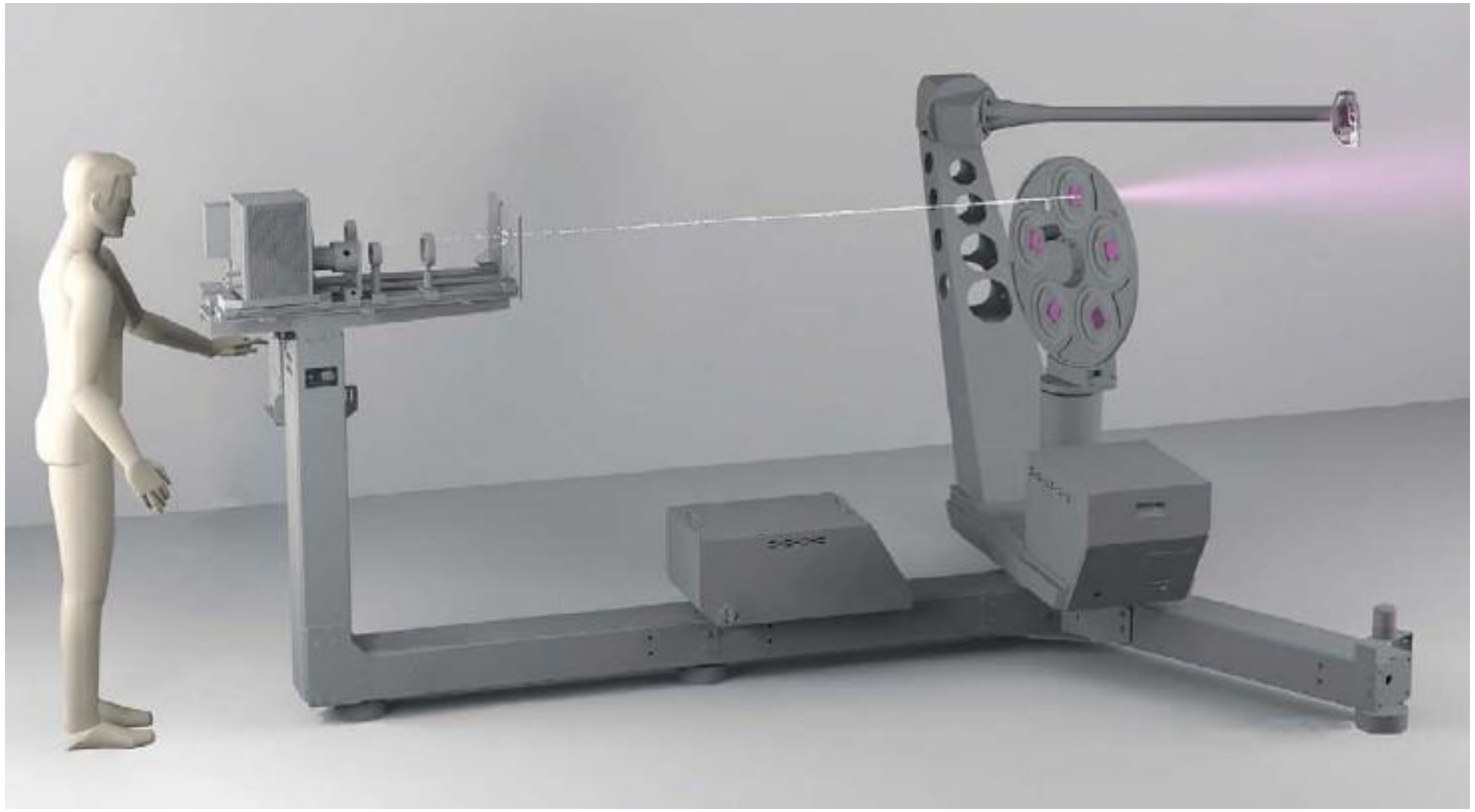






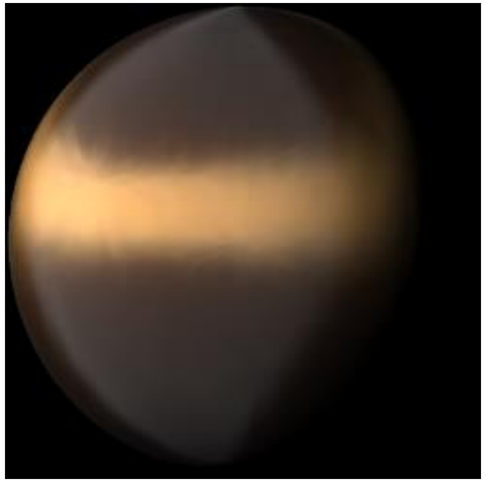
Bidirectional Reflectance Distribution Functions (BRDF)

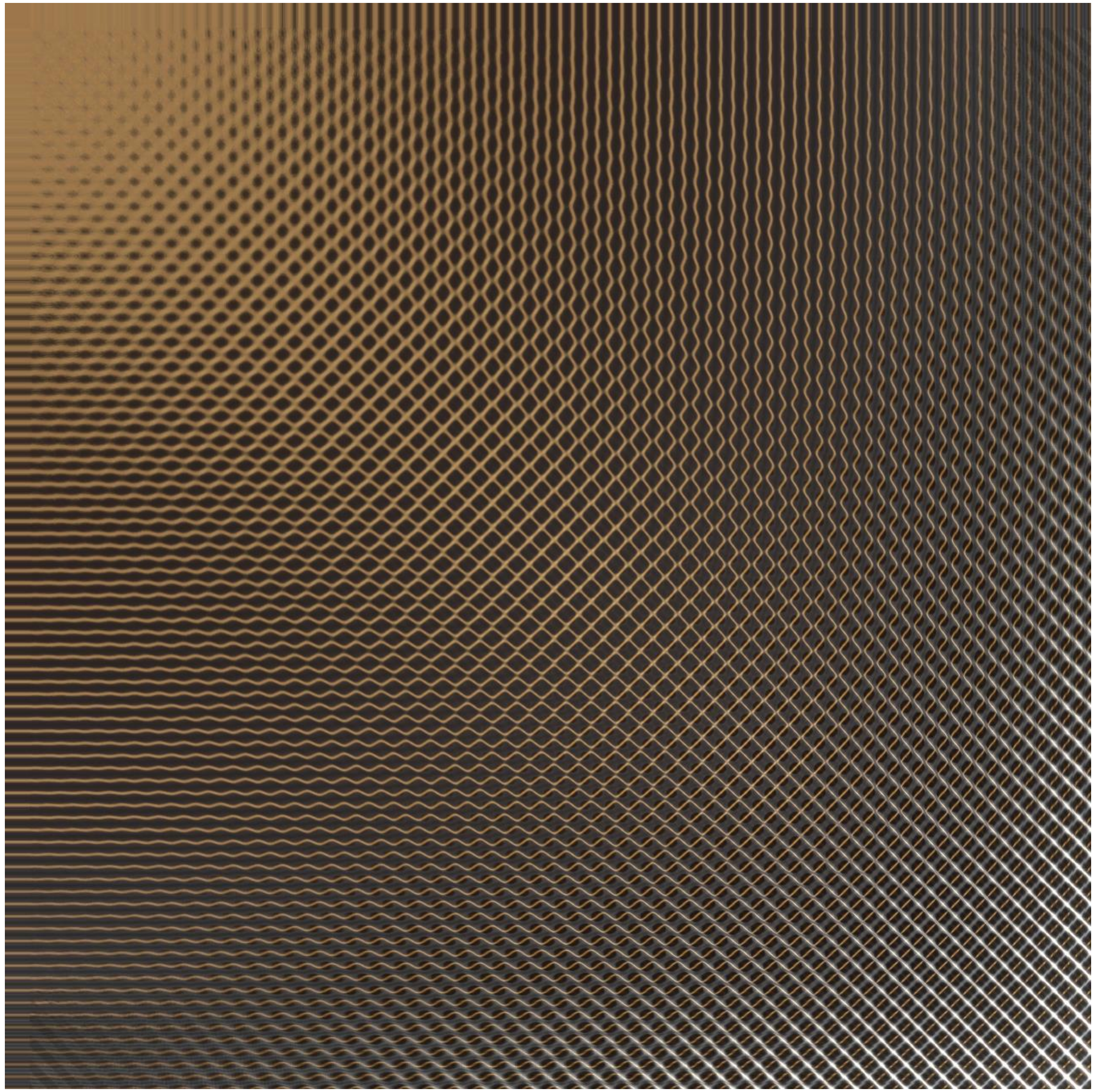
- Empirical models



Bidirectional Reflectance Distribution Functions

Empirical models

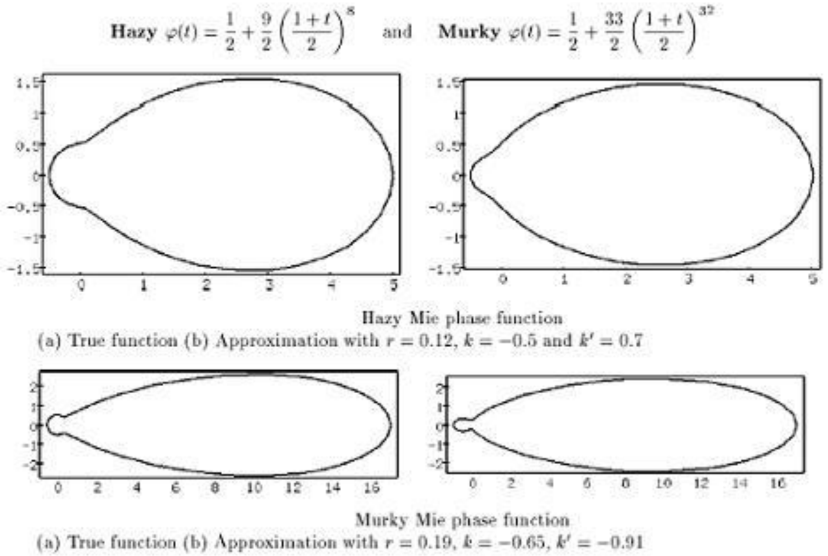


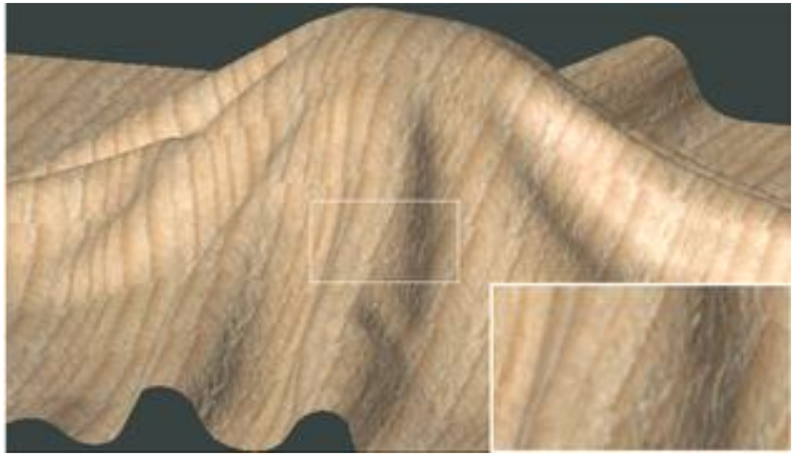


Bidirectional Reflectance Distribution Functions

Bidirectional Reflectance Distribution Functions (BRDF)

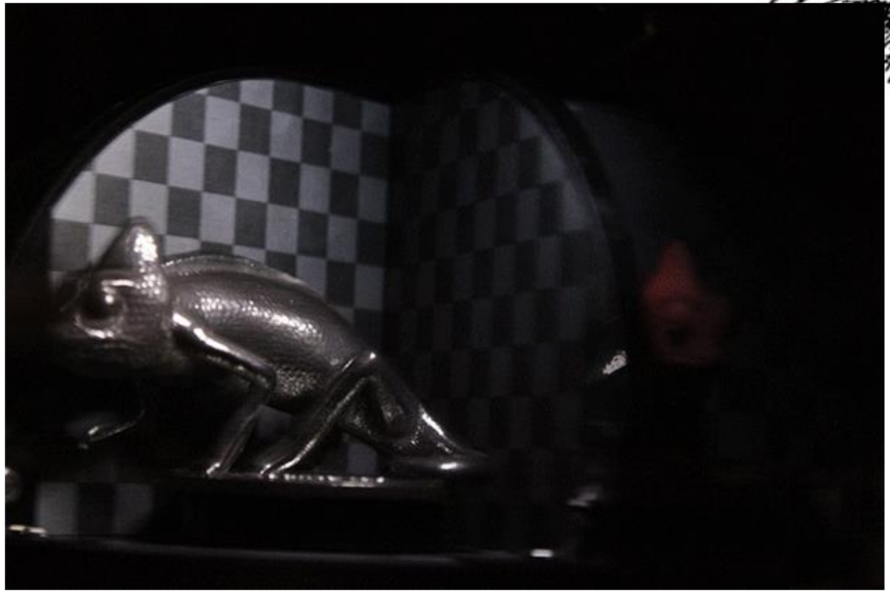
- Other / generalizations
  - BSDF
  - Scattering / Phase functions
  - BSSRDF
  - SVBRDF

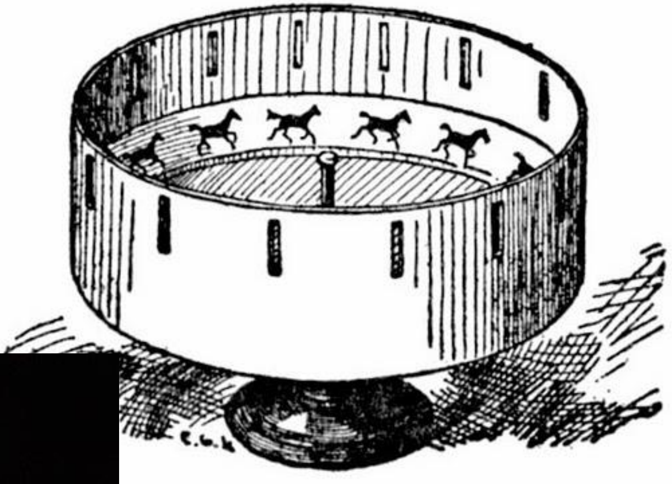




Bidirectional Reflectance Distribution Functions (BRDF)

- Parenthesis: BRDF printing
  - Zoematrope





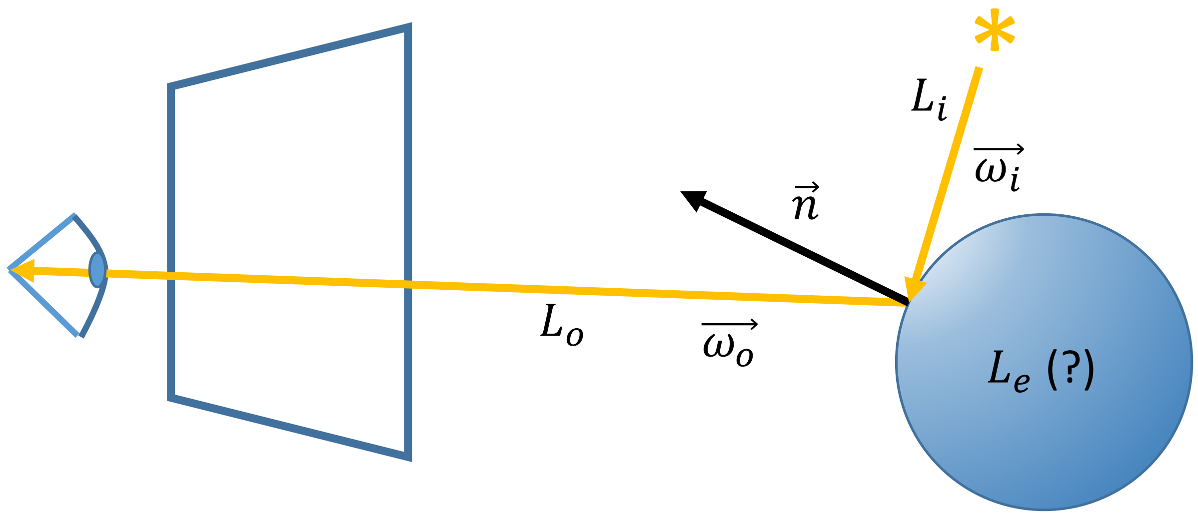
THE ZOOTROPE.

L. Miyashita, K. Ishihara, Y. Watanabe and M. Ishikawa
ZoeMatrope: A System for Physical Material Design (SIGGRAPH 2016)

The rendering equation

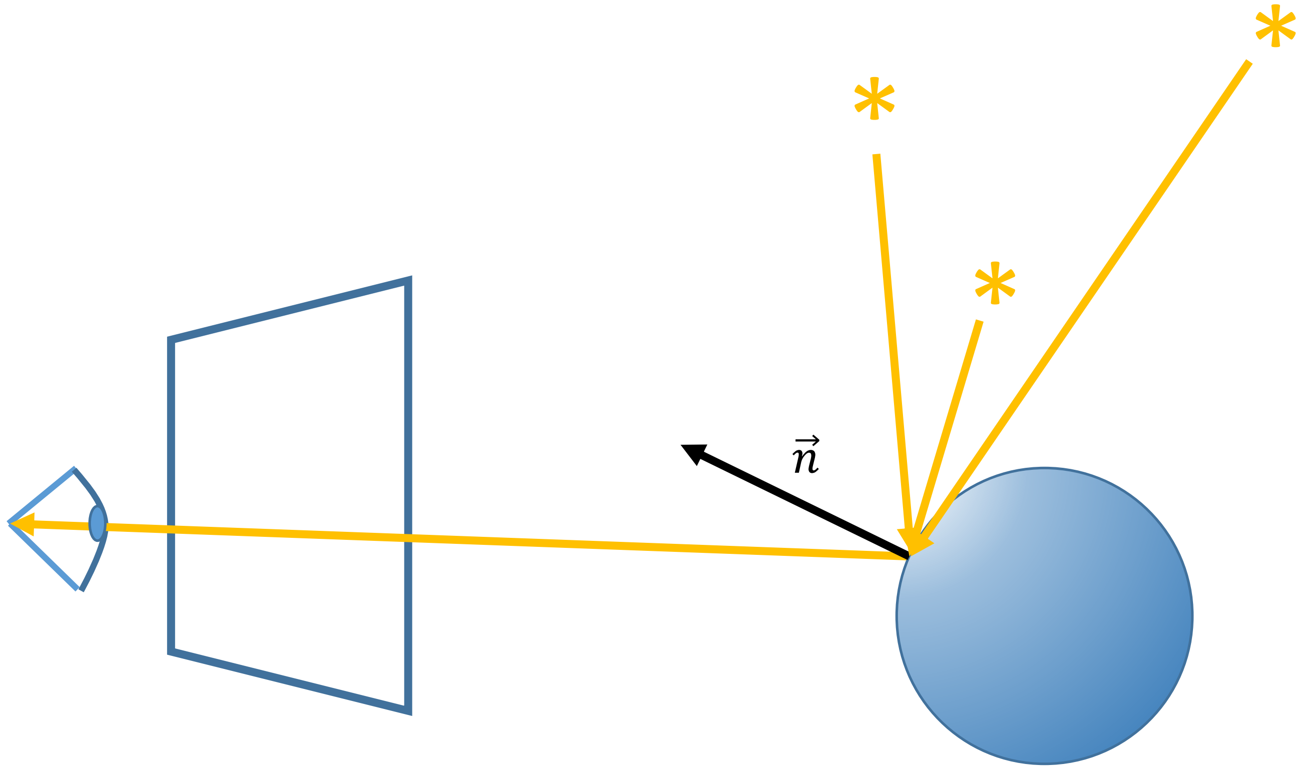
$$L_o(x,\overrightarrow{\omega_o}) = L_e(x,\overrightarrow{\omega_o}) + \int_\Omega f(\overrightarrow{\omega_i},\overrightarrow{\omega_o}) L_i(x,\overrightarrow{\omega_i}) \langle \overrightarrow{\omega_i},\vec{n}\rangle d\overrightarrow{\omega_i}$$

- Assumptions:
  - Geometric optics
  - No subsurface scattering, fluorescence, transparency, polarization



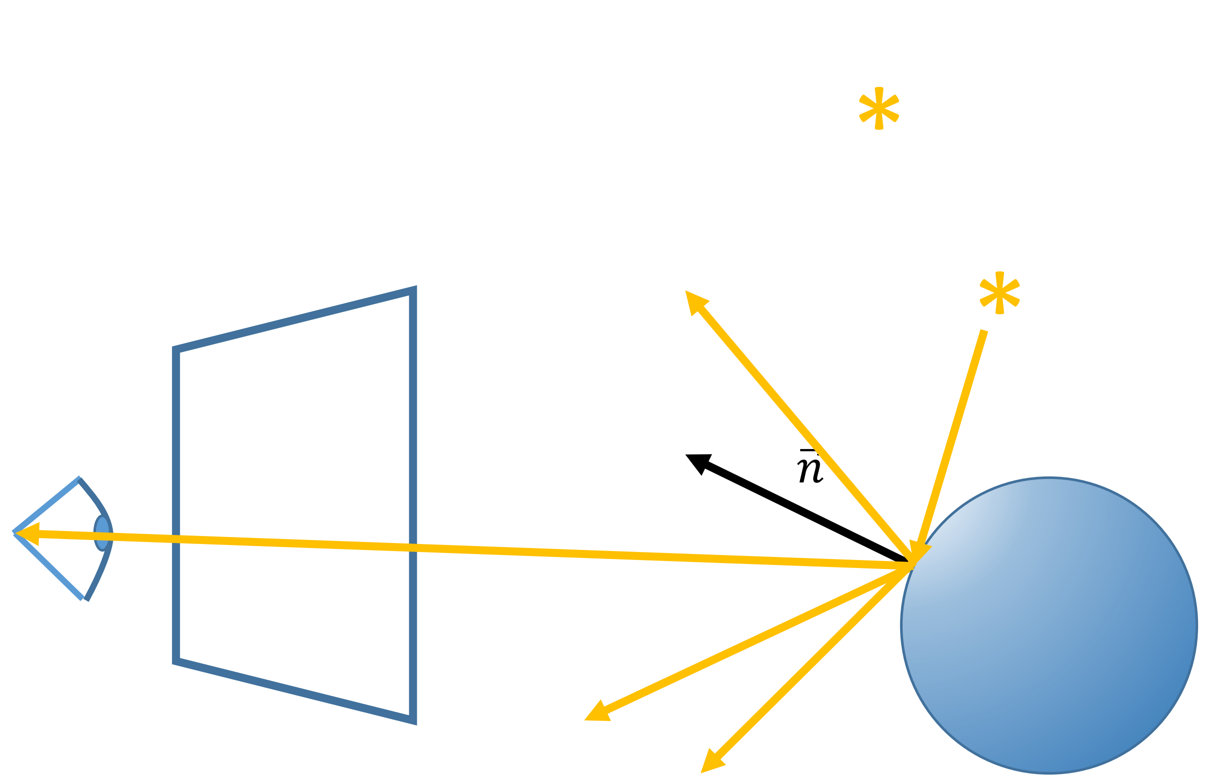
The rendering equation

$$L_o(x,\overrightarrow{\omega_o}) = L_e(x,\overrightarrow{\omega_o}) + \int_\Omega f(\overrightarrow{\omega_i},\overrightarrow{\omega_o}) L_i(x,\overrightarrow{\omega_i}) \langle \overrightarrow{\omega_i},\vec{n}\rangle d\overrightarrow{\omega_i}$$



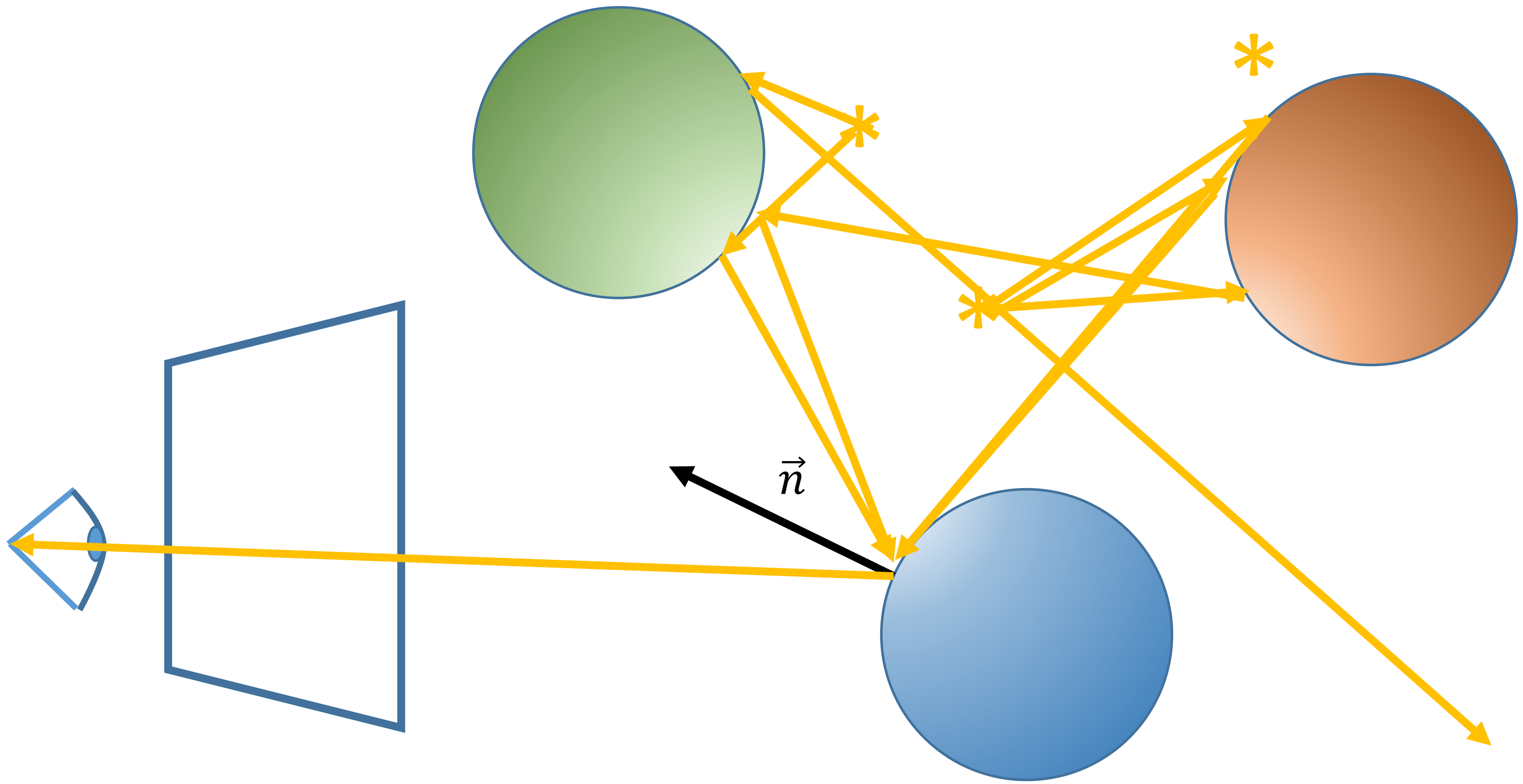
The rendering equation

$$L_o(x,\overrightarrow{\omega_o}) = L_e(x,\overrightarrow{\omega_o}) + \int_{\Omega} f(\overrightarrow{\omega_i},\overrightarrow{\omega_o}) L_i(x,\overrightarrow{\omega_i}) \langle \overrightarrow{\omega_i},\vec{n} \rangle d\overrightarrow{\omega_i}$$

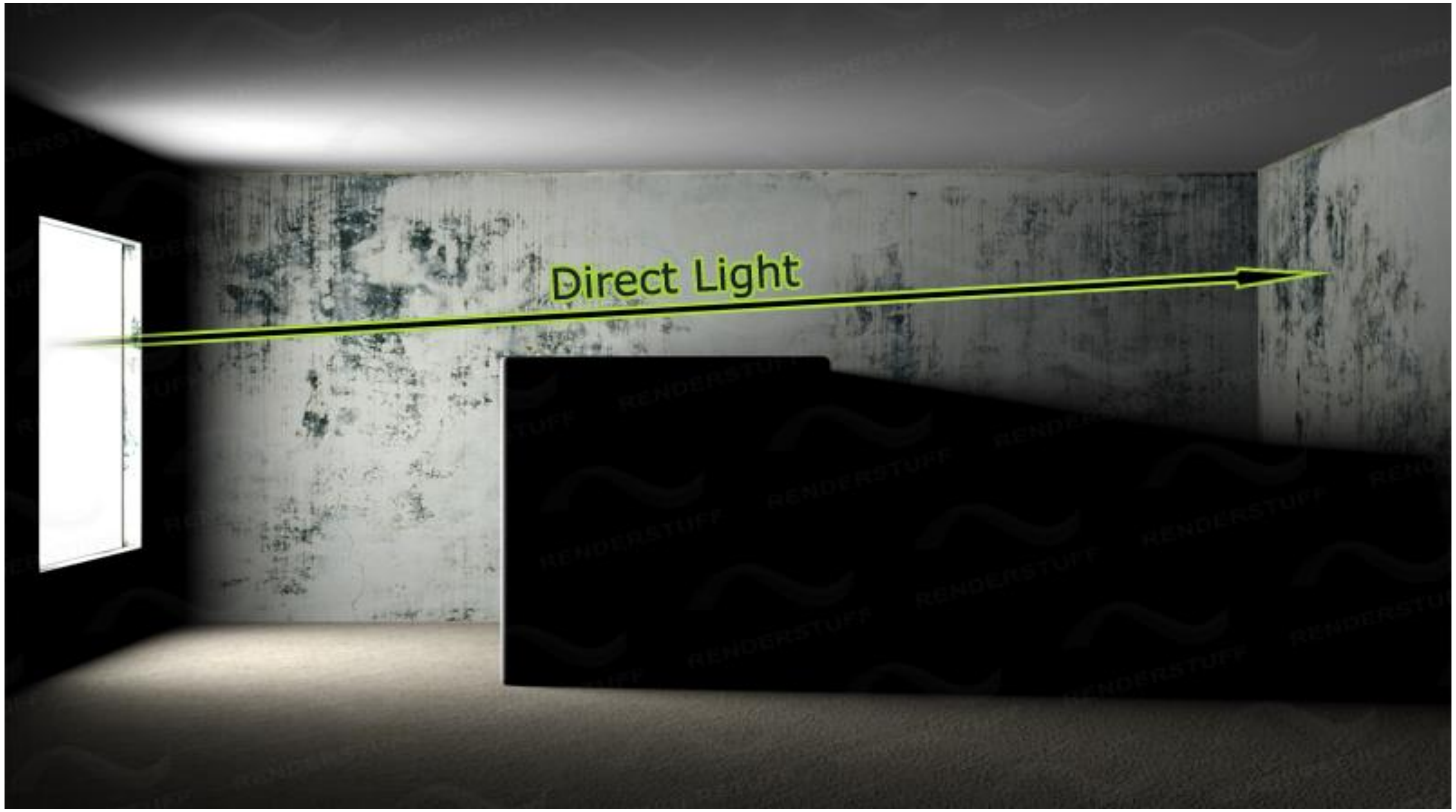


The rendering equation

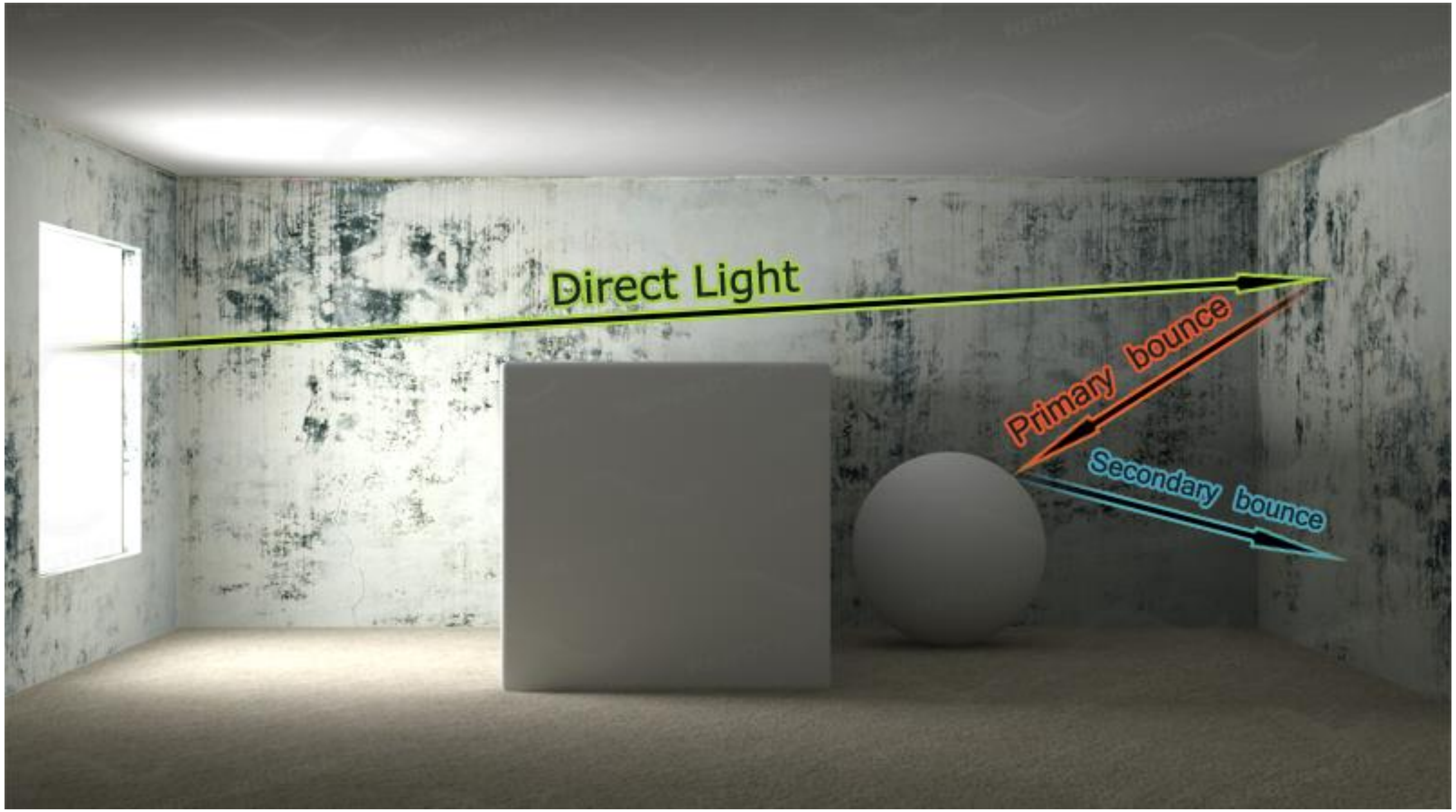
$$L_o(x,\overrightarrow{\omega_o}) = L_e(x,\overrightarrow{\omega_o}) + \int_{\Omega} f(\overrightarrow{\omega_i},\overrightarrow{\omega_o})L_i(x,\overrightarrow{\omega_i})\langle\overrightarrow{\omega_i},\vec{n}\rangle d\overrightarrow{\omega_i}$$



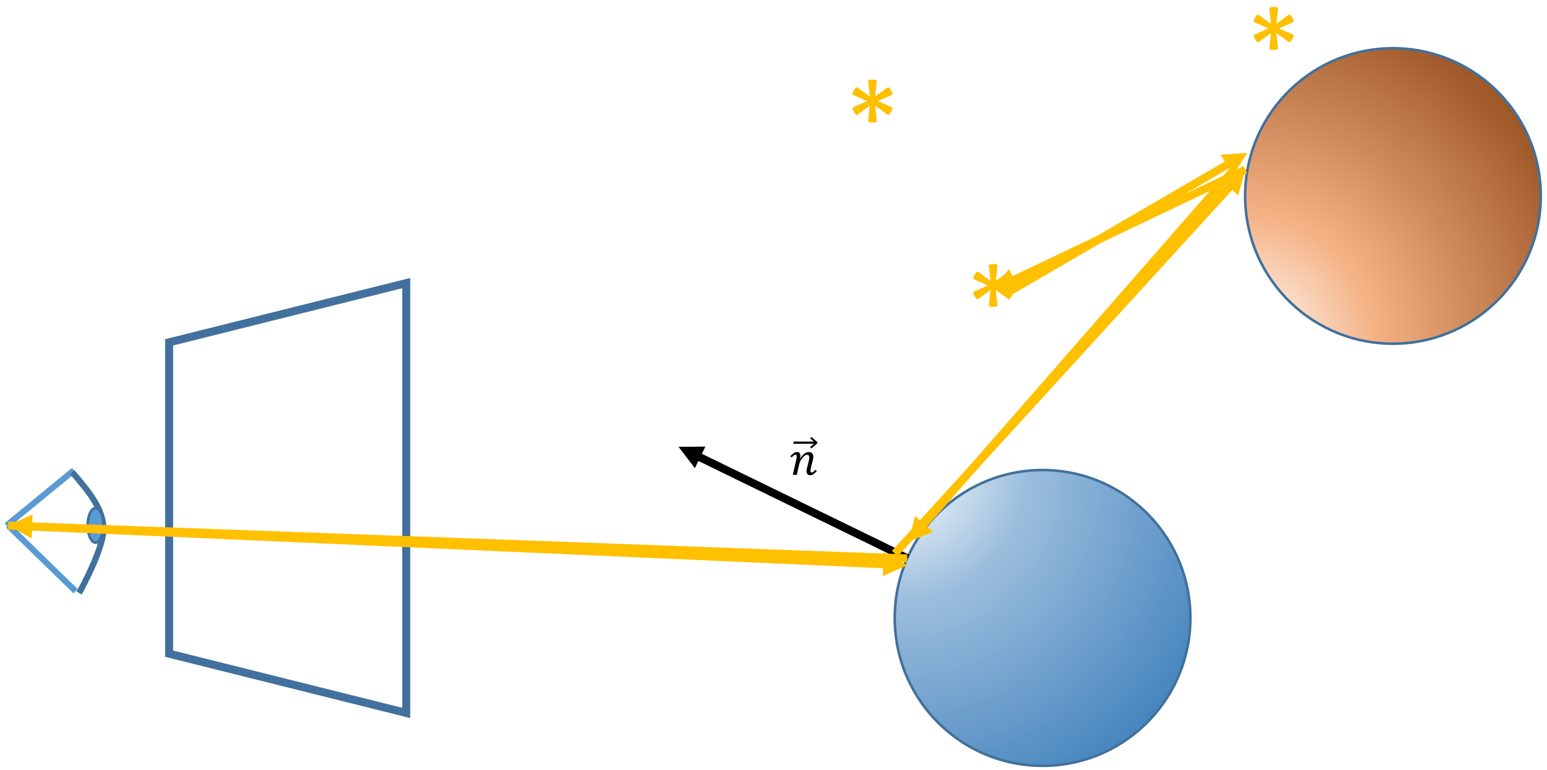
The rendering equation



The rendering equation



First idea: Using Helmoltz's reciprocity



Idea behind Backward Raytracing / path tracing

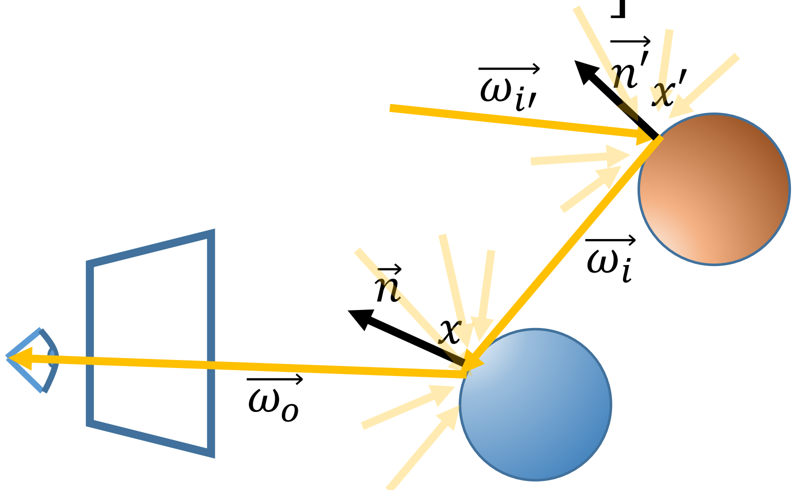
The rendering equation

$$L_o(x,\overrightarrow{\omega_o}) = L_e(x,\overrightarrow{\omega_o}) + \int_\Omega f(\overrightarrow{\omega_i},\overrightarrow{\omega_o})L_i(x,\overrightarrow{\omega_i})\langle\overrightarrow{\omega_i},\vec{n}\rangle d\overrightarrow{\omega_i}$$

At each bounce, new integral

$$L_o(x,\overrightarrow{\omega_o}) = L_e(x,\overrightarrow{\omega_o}) + \int_\Omega f(\overrightarrow{\omega_i},\overrightarrow{\omega_o})\left[L_e(x',\overrightarrow{\omega_i}) + \int_\Omega f(\overrightarrow{\omega_{i'}},\overrightarrow{\omega_i})L_i(x',\overrightarrow{\omega_{i'}})\left\langle\overrightarrow{\omega_{i'}},\overrightarrow{n'}\right\rangle d\overrightarrow{\omega_{i'}}\right]\langle\overrightarrow{\omega_i},\vec{n}\rangle d\overrightarrow{\omega_i}$$

Fredholm equation of the second kind

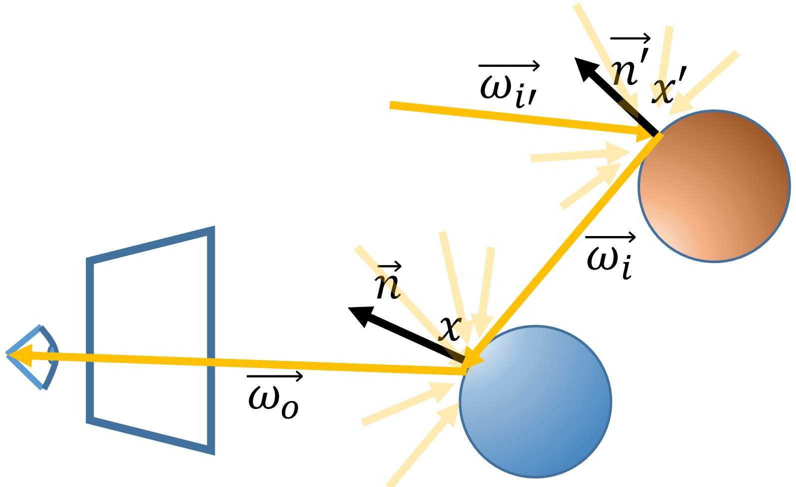


The rendering equation

$$L_o(x,\overrightarrow{\omega_o}) = L_e(x,\overrightarrow{\omega_o}) + \int_\Omega f(\overrightarrow{\omega_i},\overrightarrow{\omega_o}) L_i(x,\overrightarrow{\omega_i}) \langle \overrightarrow{\omega_i},\vec{n}\rangle d\overrightarrow{\omega_i}$$

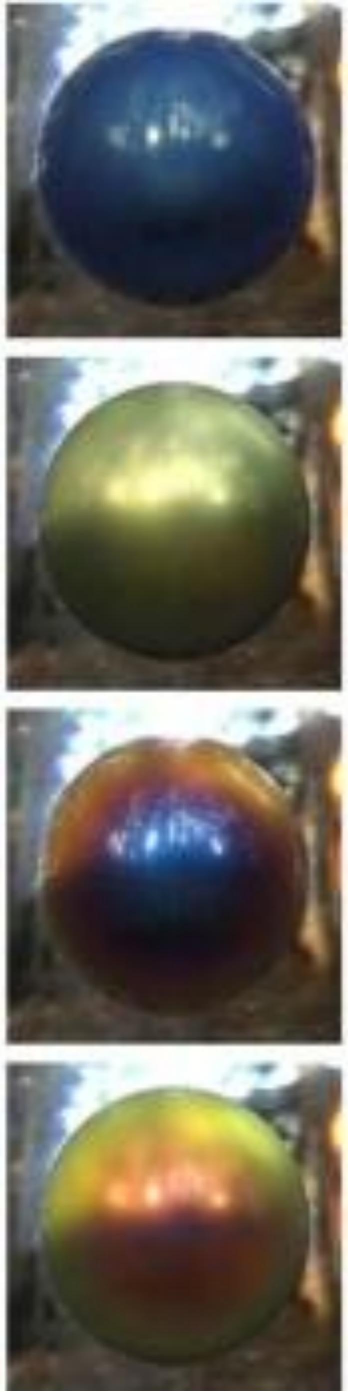
At each bounce, new integral

$$L_o(x,\overrightarrow{\omega_o}) = L_e + \int_\Omega \int_\Omega \int_\Omega \int_\Omega \dots \int_\Omega F(x,x',x'',x''',\dots,x'''',\omega_i,\omega_i',\dots,\omega_i'''',\omega_o,\omega_o',\dots,\omega_o'''')d\omega d\omega' \dots d\omega''''$$

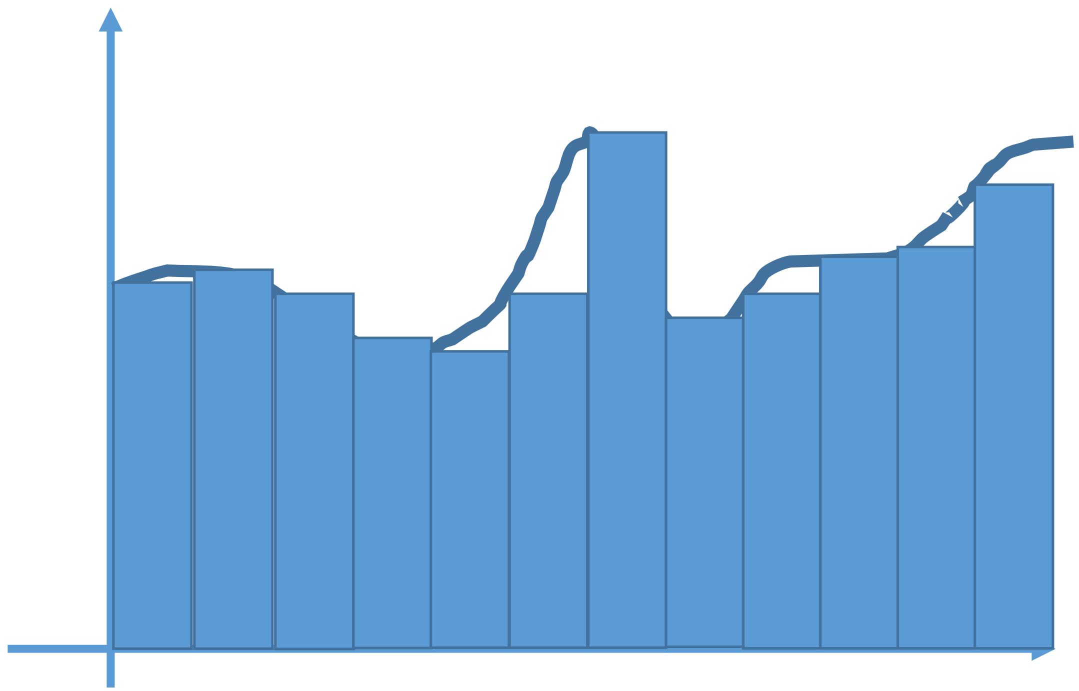


The rendering equation

- In fact, additional dimensions for the camera model
  - Anti-Aliasing (+2D)
  - Depth-of-Field (+2D)
  - Motion blur (+1D)
  - Multispectral (+1D)



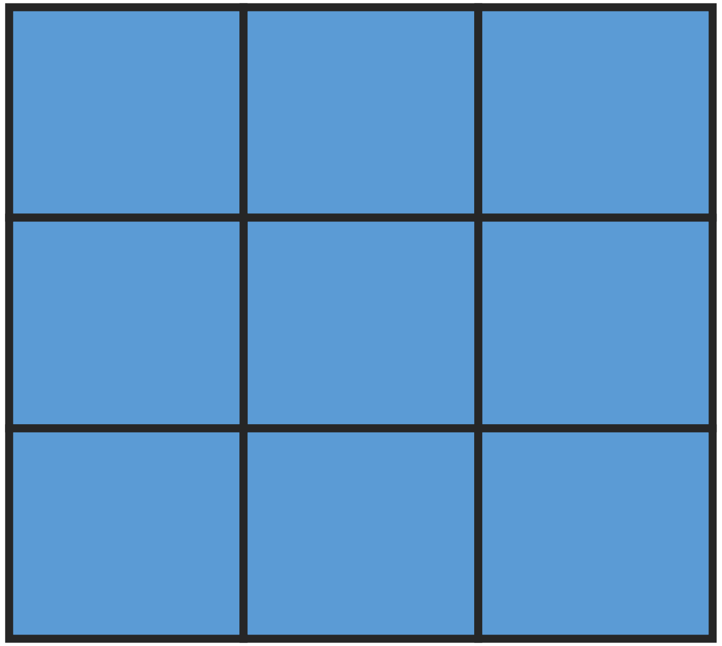
Second idea: integration



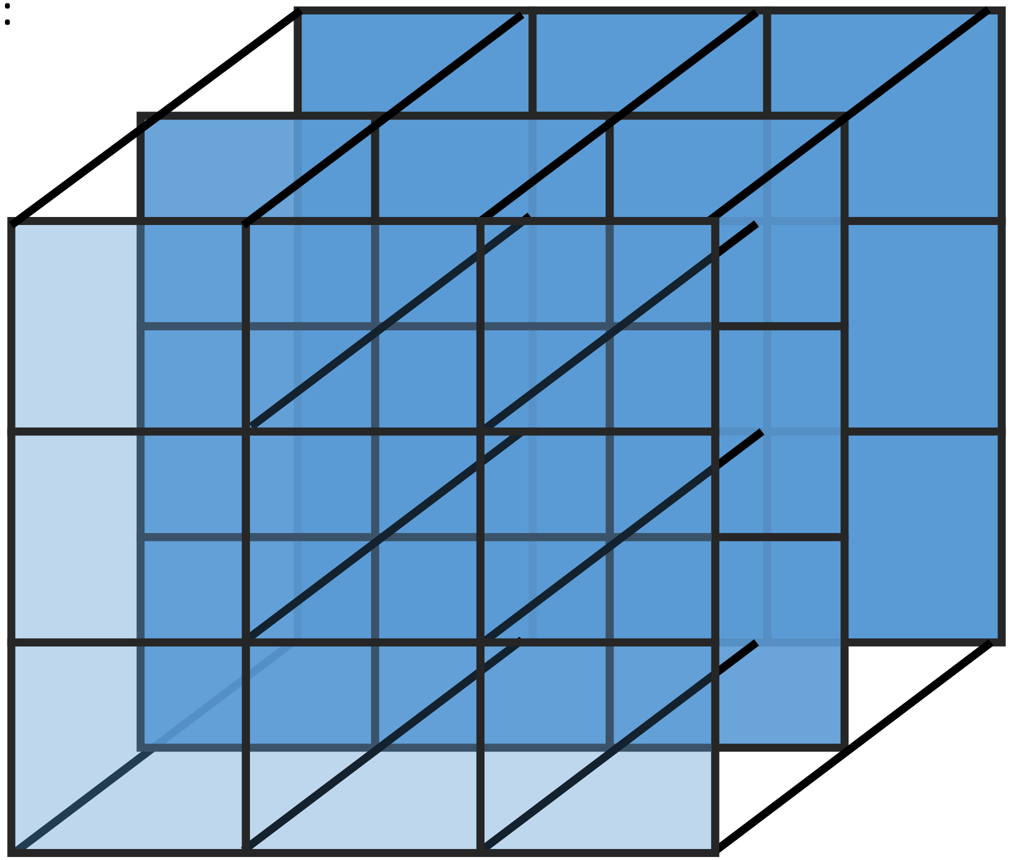
Method of rectangles, error in 1/N
($1/N^2$ for mid-point method – and better schemes exist)

Second idea: integration

In 2D:



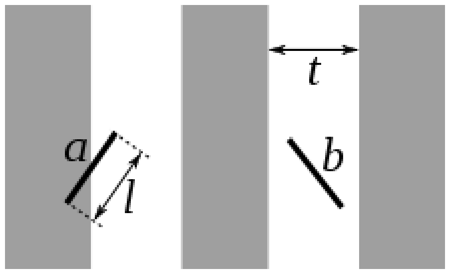
In 3D:



Second idea: integration

- Instead, Monte-Carlo integration

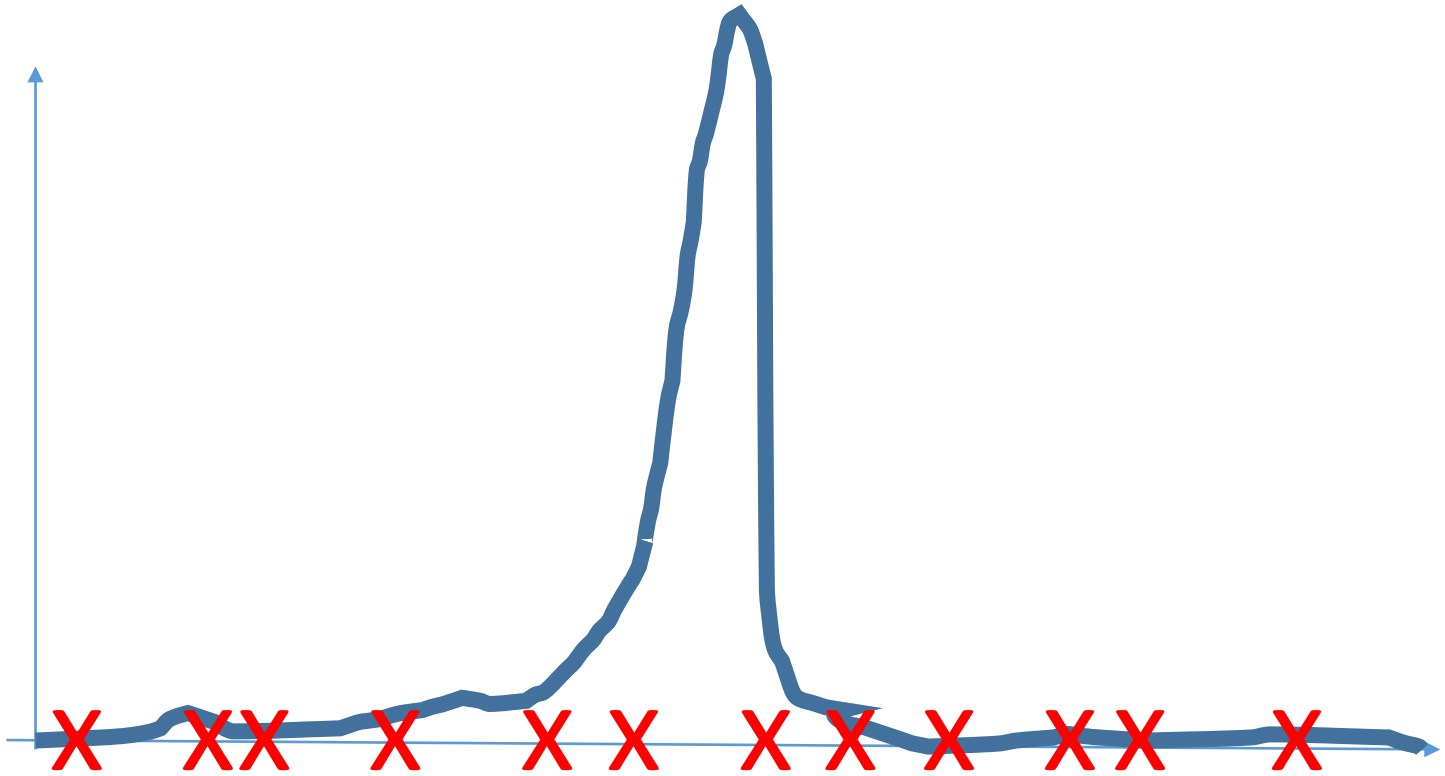
- Example: Buffon's needle to estimate Pi
  - Probability for a needle to cross a line: $\mathrm{P} = \frac{2l}{t\pi}$



  - So, $\pi = \frac{2l}{tP}$

Monte-Carlo Integration

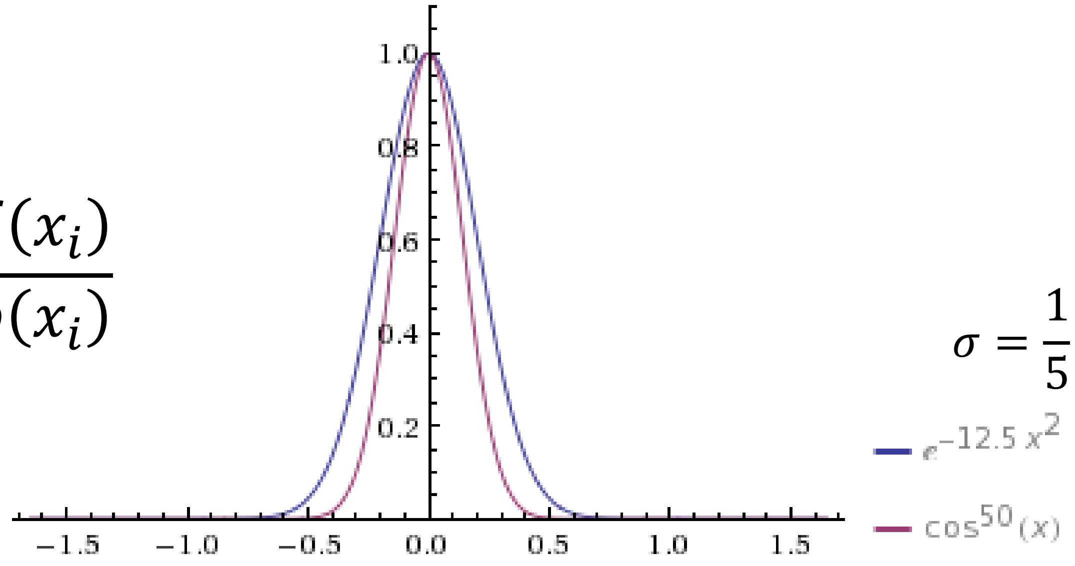
- Why not using uniform random variables ?



Problem for specular or glossy BRDFs

Monte-Carlo Integration

Plot:



$$\sigma = \frac{1}{5}$$

- General formula:

$$\int f(x)dx \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}$$

Where $x_i \sim \mathcal{L}$

- Example

$$\int_{x=-\frac{\pi}{2}}^{+\frac{\pi}{2}} \cos^{50} x\, dx \approx \frac{1}{N}\sum_{i=1}^{N}\frac{\cos^{50} x_i}{\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{x_i^2}{2\sigma^2}\right)}$$

Where $x_i \sim \mathcal{N}(0, \sigma^2)$

Monte Carlo Integration

- Other example

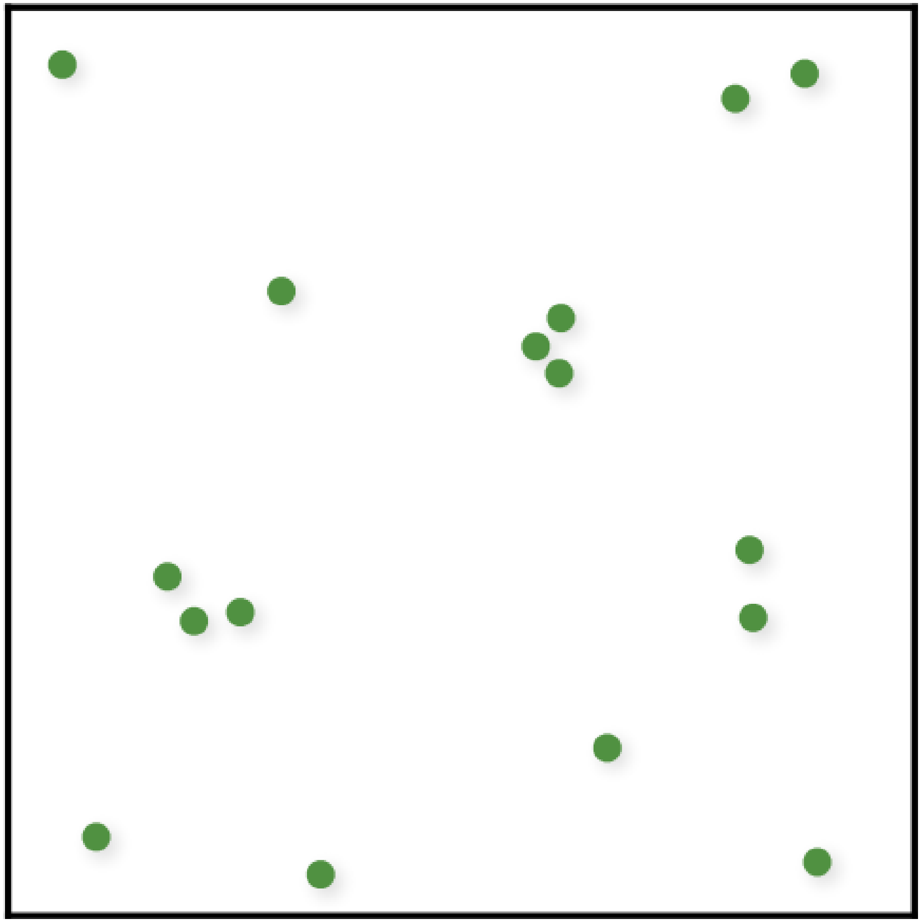
$$\int_{x=-\frac{\pi}{2}}^{+\frac{\pi}{2}}\int_{y=-\frac{\pi}{2}}^{+\frac{\pi}{2}}\int_{z=-\frac{\pi}{2}}^{+\frac{\pi}{2}}\int_{w=-\frac{\pi}{2}}^{+\frac{\pi}{2}} \cos^{50}(x*y*z*w) \quad dw\, dz\, dy\, dx \approx \frac{1}{N}\sum_{i=1}^{N} \frac{\cos^{50}(x_i * y_i * z_i * w_i)}{\frac{1}{\left(\sigma\sqrt{2\pi}\right)^4}\exp\left(-\frac{(x_i^2+y_i^2+z_i^2+w_i^2)}{2\sigma^2}\right)}$$

Where $x_i, y_i, z_i, w_i \sim \mathcal{N}(0, \sigma^2)$

  - Unlike rectangle integration, STILL a single sum

- Property
  - If $\mathrm{f}(x) = \alpha\, p(x)$, we have $\int f(x)dx \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)} = \frac{1}{N}\sum_{i=1}^{N}\alpha = \alpha$ for all N
  - ... So there is equality, and you can take a single sample: $N = 1$, or even 0 sample!
  - ...so $\int f(x)dx = \alpha$
  - ....but if you know α, this means you already knew how to integrate f
  - So this NEVER happens: you always have p “similar” to f but not equal

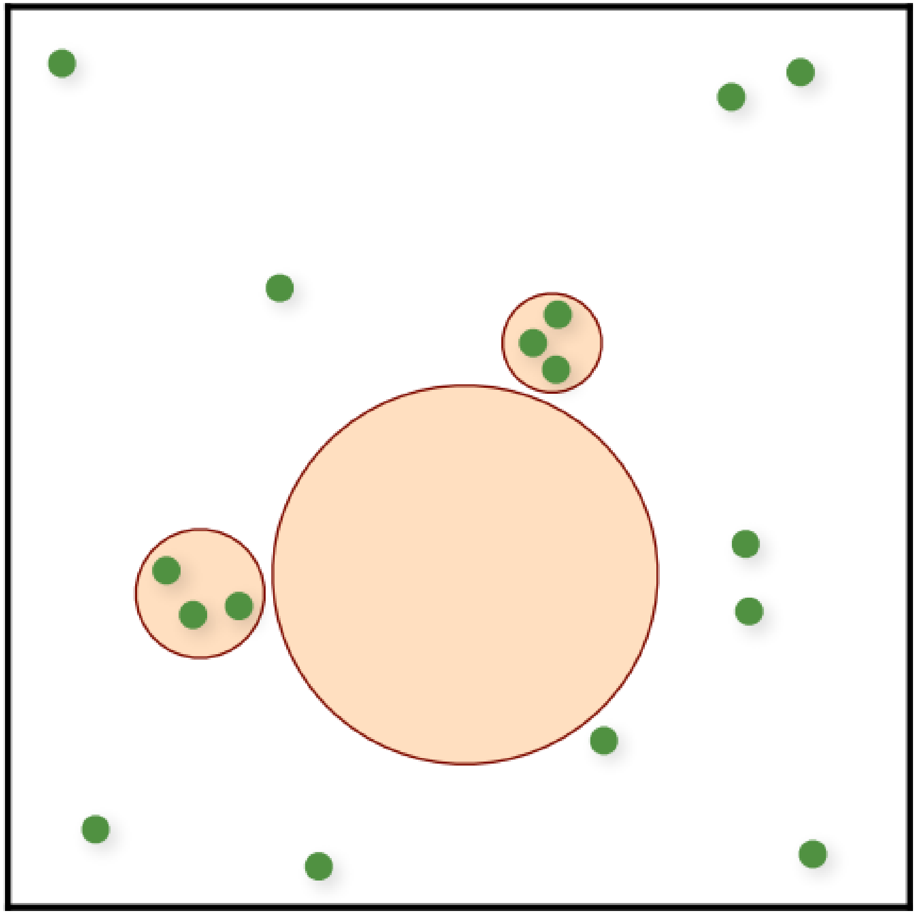
Sampling



Independent random sampling

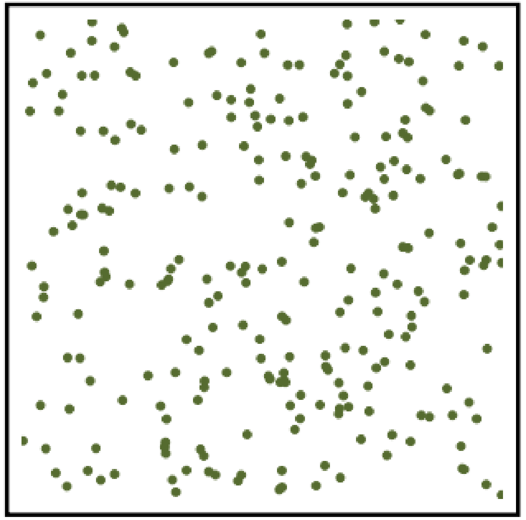
Following material from:
Fourier Analysis of Numerical Integration in Monte Carlo Rendering: Theory and Practice
Kartic Subr, Gurprit Singh, Wojciech Jarosz

Sampling

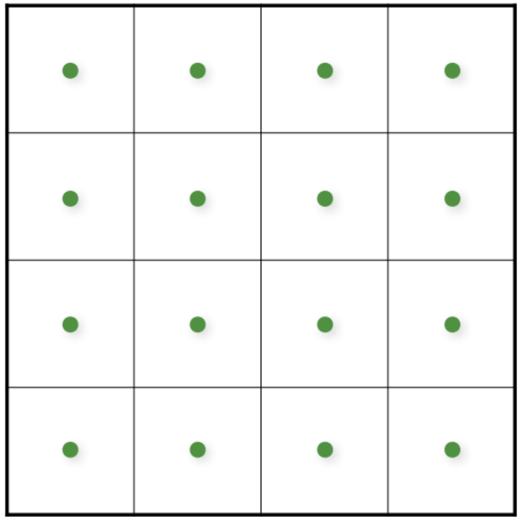


Independent random sampling

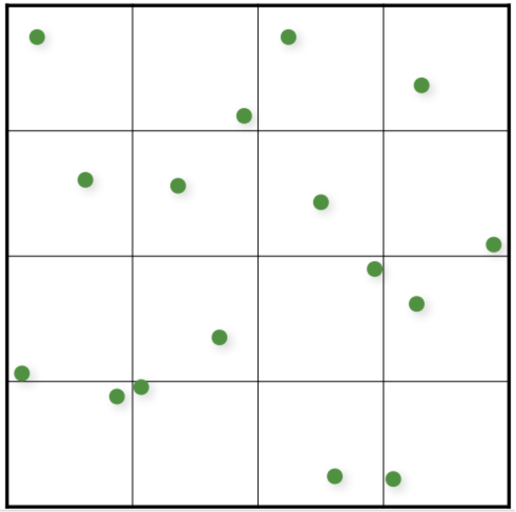
Sampling



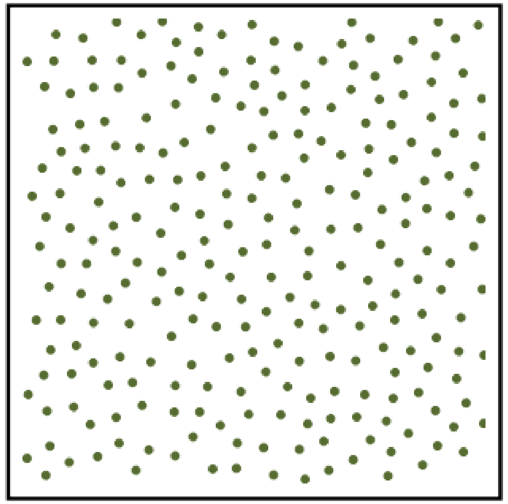
Independent random sampling



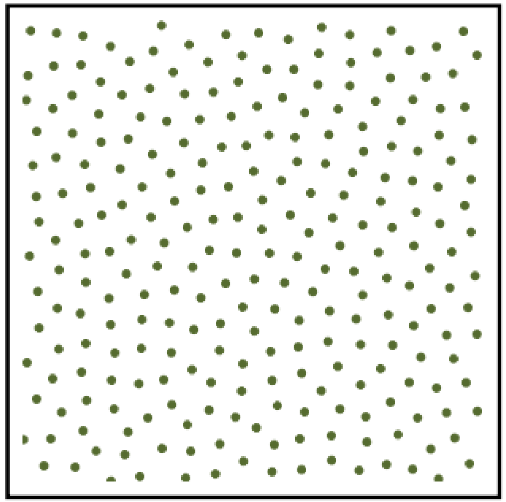
Regular sampling



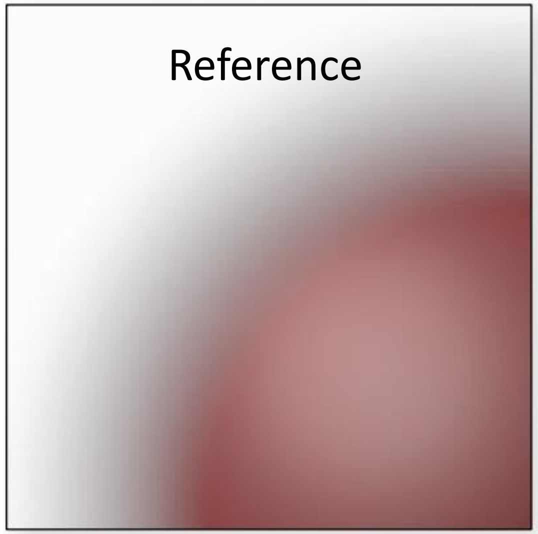
Jittered/stratified sampling

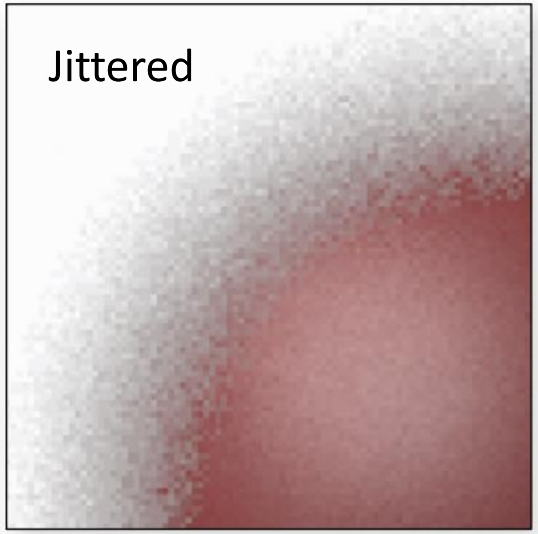


Poisson Disk



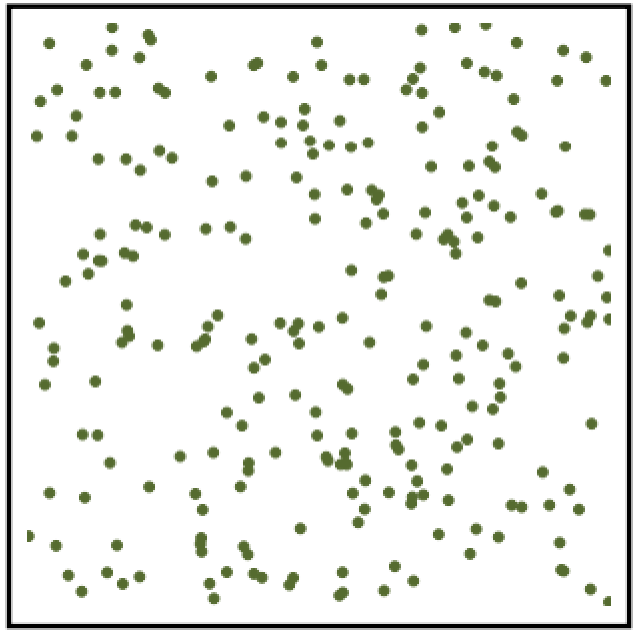
CCVT

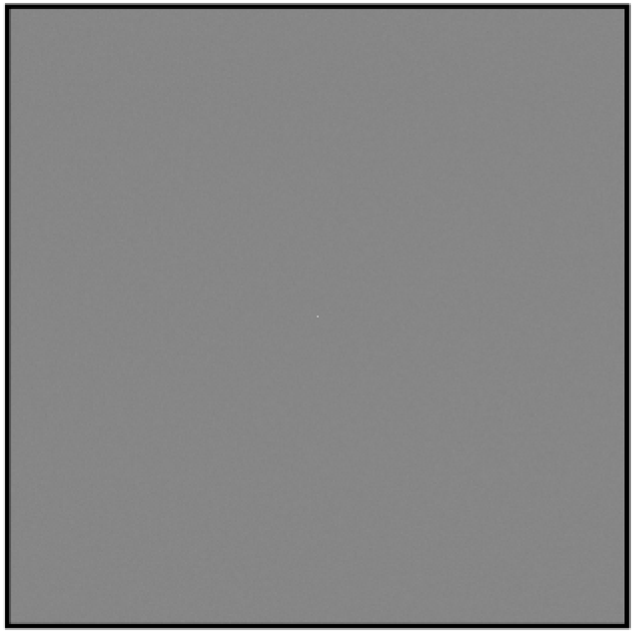


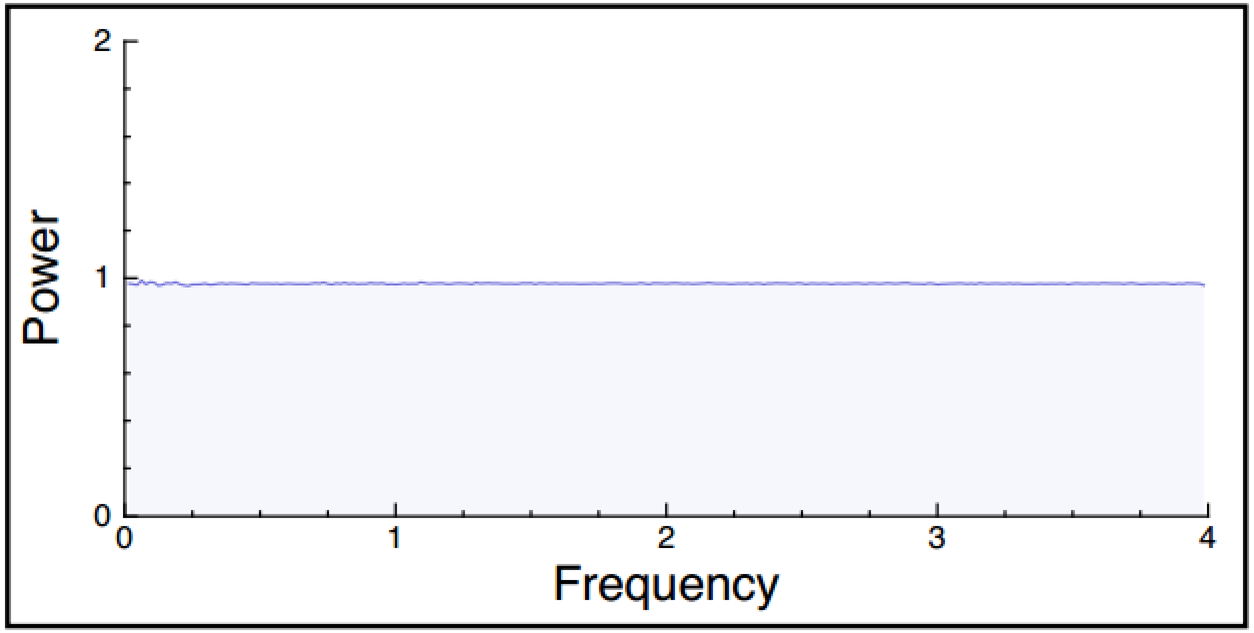


Frequency analysis

Samples | Expected power spectrum | Radial mean



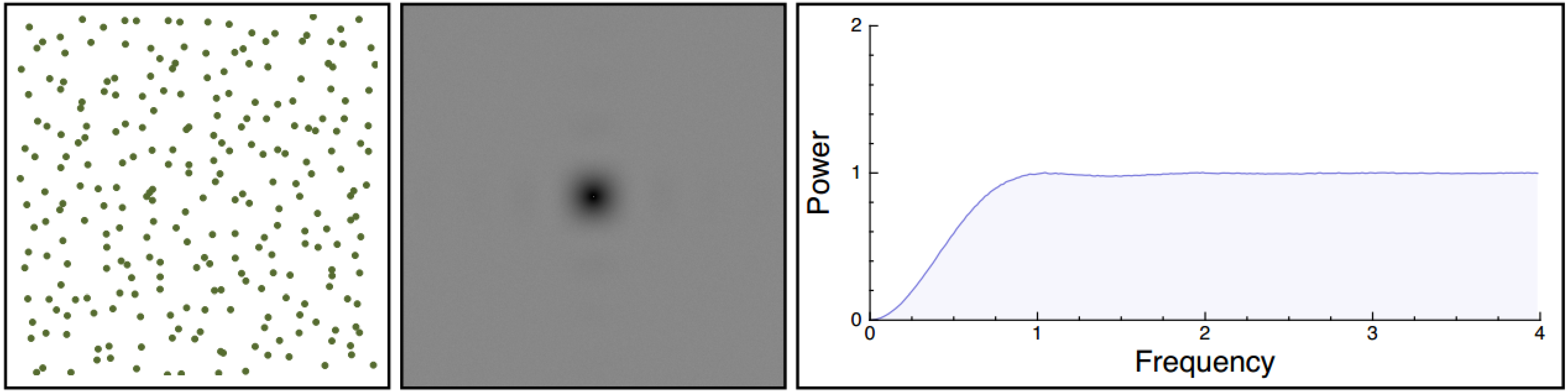




Uniform sampling

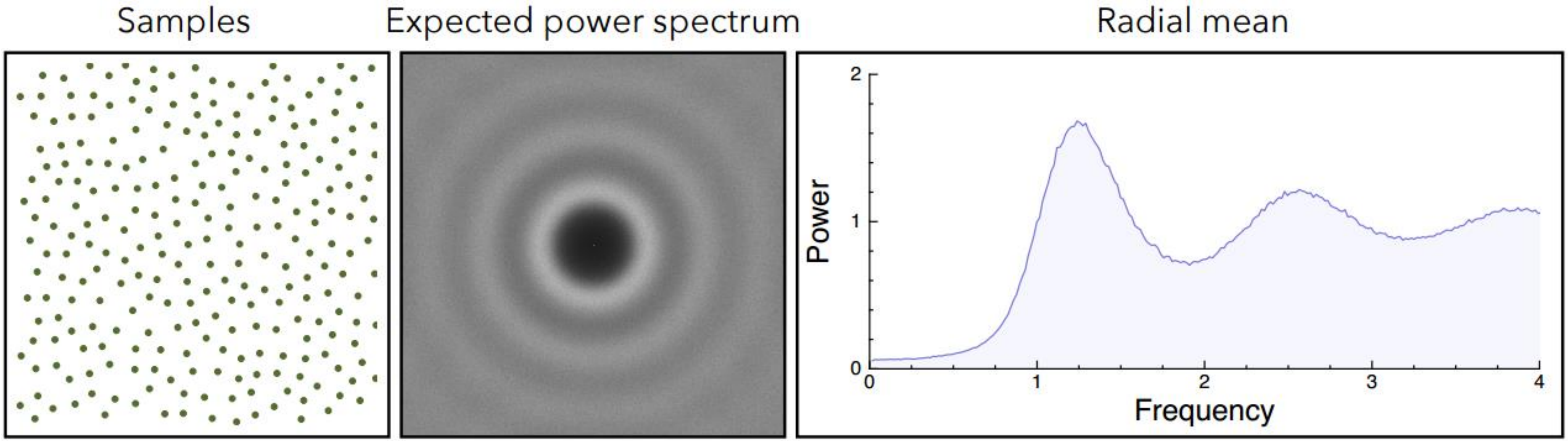
Frequency analysis

Samples | Expected power spectrum | Radial mean



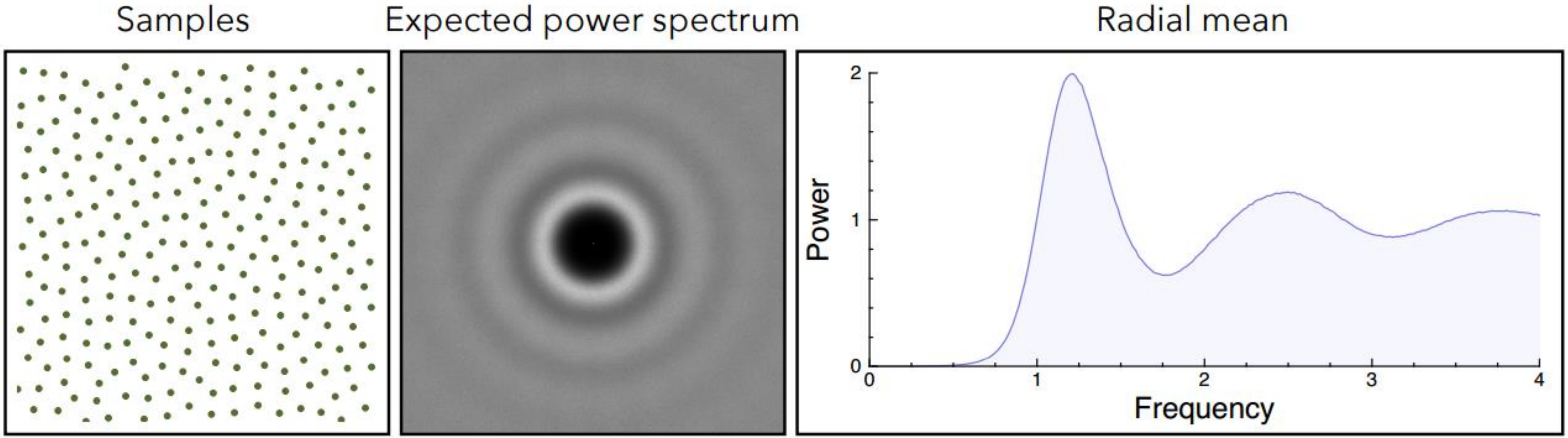
Jittered sampling

Frequency analysis



Poisson Disk

Frequency analysis



CCVT

Monte Carlo Integration

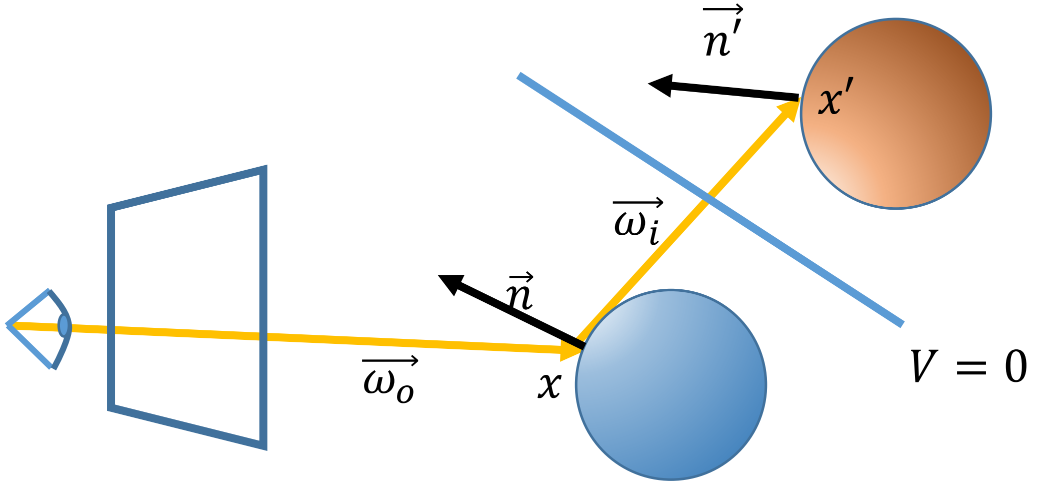
- Convergence rate : Depends on frequency of samples
  - Random: variance in $1/N$
  - Jitter : variance in $1/N^{1.5}$
  - Poisson Disk : variance in $1/N$
  - CCVT: variance in $1/N^{1.5}$
- With $\text{error} = \sqrt{\text{variance}}$
- So, using standard random numbers, 4x more samples for 2x accuracy
- using jittered sampling, 2.5x more samples for 2x accuracy
- $Var(I_N) = \frac{\mu(T^d)^2 \mu(S^{d-1})^2}{N} \int_0^\infty \rho^{d-1} \breve{P}_S(\rho) \breve{P}_F(\rho) \mathrm{d}\rho$
  - With $\mu(S^{d-1}) = \frac{2\sqrt{\pi^d}}{\Gamma\left(\frac{d}{2}\right)}$ and $\mu(T^d) = 1$

Third idea: Change of variables

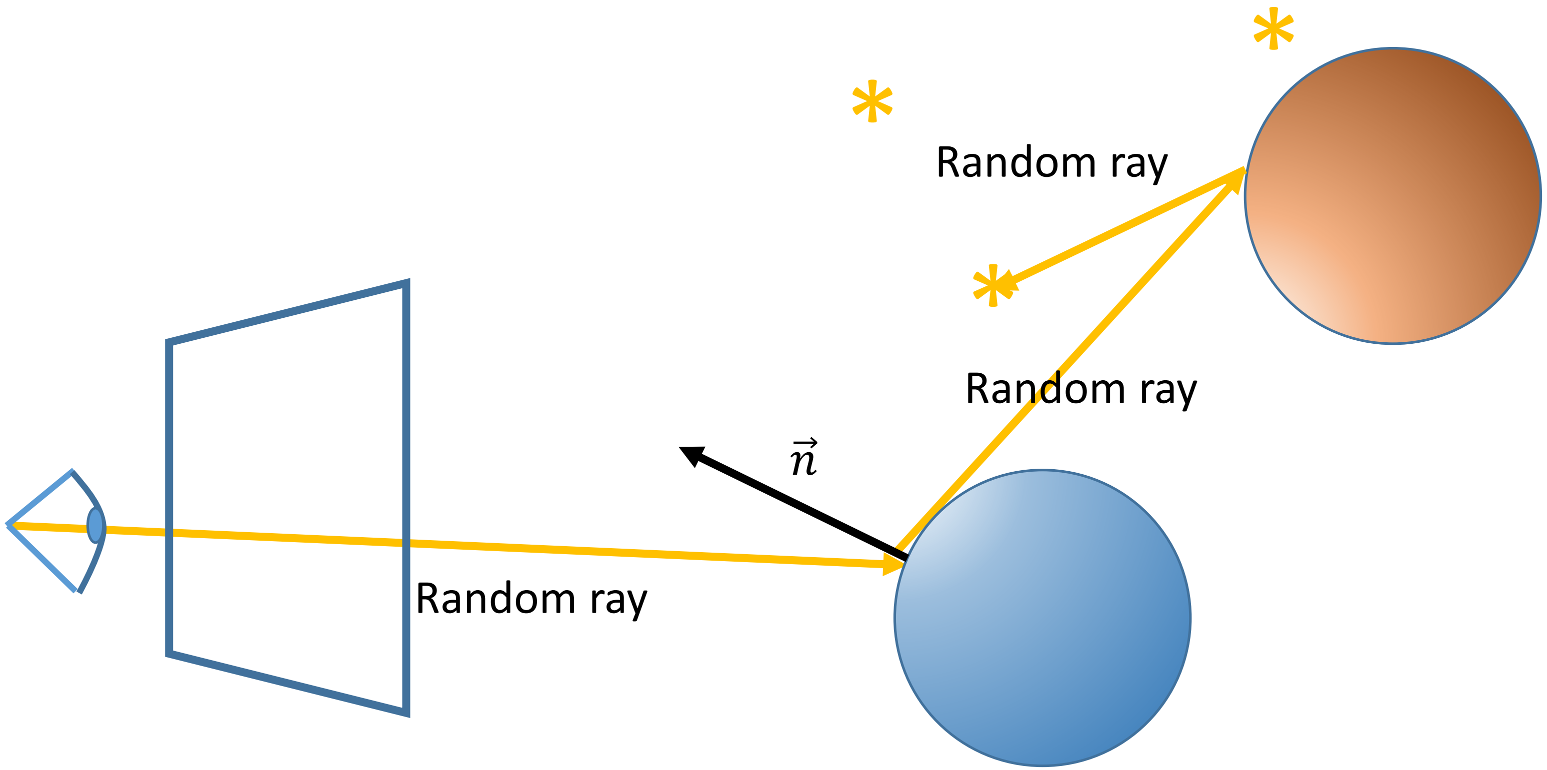
$$L_o(x, \overrightarrow{\omega_o}) = L_e(x, \overrightarrow{\omega_o}) + \int_{\Omega} f(\overrightarrow{\omega_i}, \overrightarrow{\omega_o}) L_i(x, \overrightarrow{\omega_i}) \langle \overrightarrow{\omega_i}, \vec{n} \rangle d\overrightarrow{\omega_i}$$

$$\Leftrightarrow$$

$$L_o(x, \overrightarrow{\omega_o}) = L_e(x, \overrightarrow{\omega_o}) + \int_{\mathrm{P}} f(\overrightarrow{\omega_i}, \overrightarrow{\omega_o}) L_i(x, \overrightarrow{\omega_i}) \langle \overrightarrow{\omega_i}, \vec{n} \rangle \frac{V(x, x') \left| \langle \overrightarrow{\omega_i}, \vec{n}' \rangle \right|}{\|x - x'\|^2} dP$$



Path Tracing



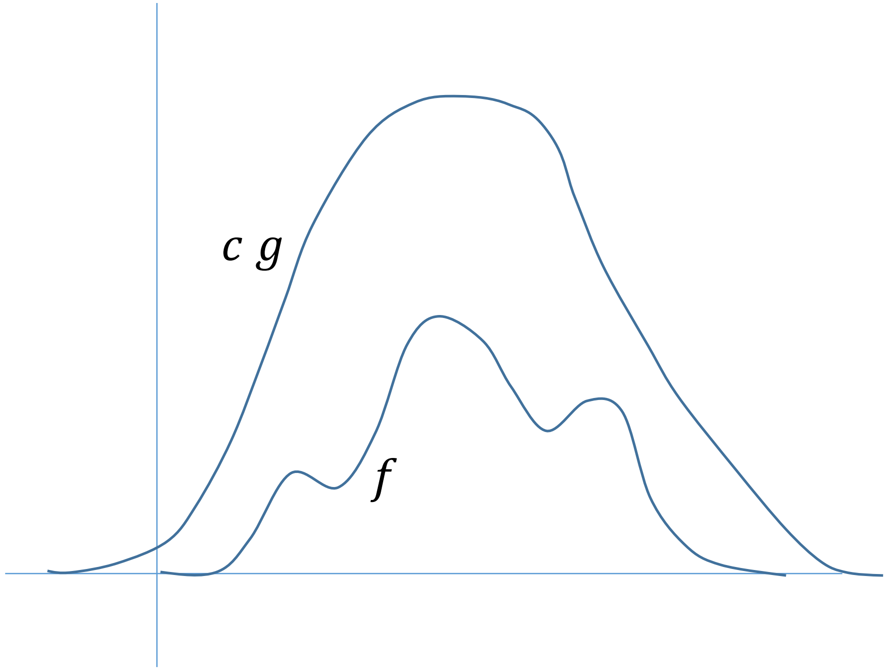
Path Tracing

- How to generate a random ray
  - With a probability similar to BRDF ?
    - Mostly easy: for diffuse BRDFs and some BRDF models
      e.g., diffuse: $p(\theta) = \frac{\cos\theta}{\pi}$ and $(x, y, z) = \left(\cos 2\pi r_1 \sqrt{1 - r_2}, \sin 2\, \pi r_1 \sqrt{1 - r2}, \sqrt{r_2}\right)$
    - See Global Illumination Compendium (Philip Dutré)
  - With a probability similar to incoming Light ?
    - More difficult in general
    - Easy for point lights
  - Both ?
    - “Multiple Importance Sampling”

How to generate samples according to a distribution f ?

- Rejection method
  - Suppose you know how to sample g, with $\frac{f}{g} \leq c$
  - Do
    - Sample y with density g
    - Generate a uniform random number $u \sim U(0,1)$
  - While $u > \frac{f(y)}{c\, g(y)}$
  - Keep y

  Average number of iteration = c



How to generate samples according to a distribution f ?

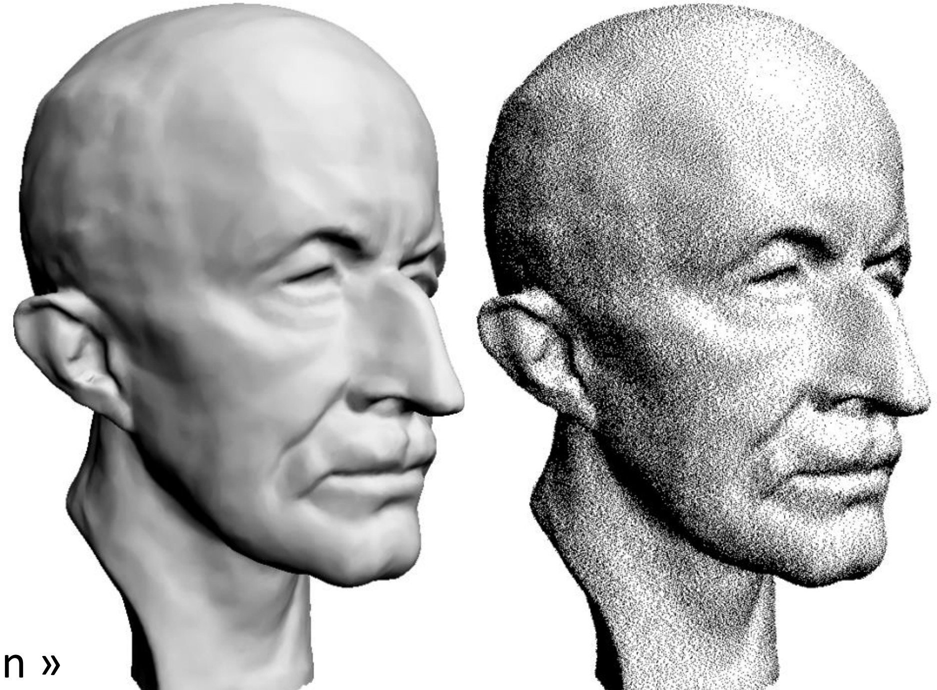
- Inverse transform sampling (1-d)
  - Compute the inverse cumulative distribution function F^{-1}
    - $F(x) = \int_{-\infty}^{x} f(t)dt$ (may use numerical integration)
    - $F^{-1}(y) = \min \{x \mid y = f(x)\}$ (may use numerical solvers)
  - Take a random uniform $u \sim U(0,1)$
  - Use $F^{-1}(u)$
- Proof
  - $P(F^{-1}(u) \leq x) = P\big(u \leq F(x)\big) = F(x)$

How to generate samples according to a distribution f ?

- Example
  - $f(x) = \lambda e^{-\lambda x} \qquad x \geq 0$
  - $F(x) = 1 - e^{-\lambda x}$
  - $F^{-1}(y) = -\frac{\log(1-y)}{\lambda}$
  - We take $u \sim U(0,1)$ and compute $v = -\frac{\log(u)}{\lambda}$

How to generate samples according to a distribution f ?

- Inverse transform sampling (images)
  - First option: concatenate image rows => 1d case
  - Second option:
    - Compute $m(x) = \int_0^1 f(x,y)dy$ the marginal density function of f
    - Then $M(y) = \int_0^y m(x)dx$ -> allows to determine y
    - Then use the conditional $c(x \mid y) = \frac{f(x,y)}{M(y)}$ and its cumulative C to determine x



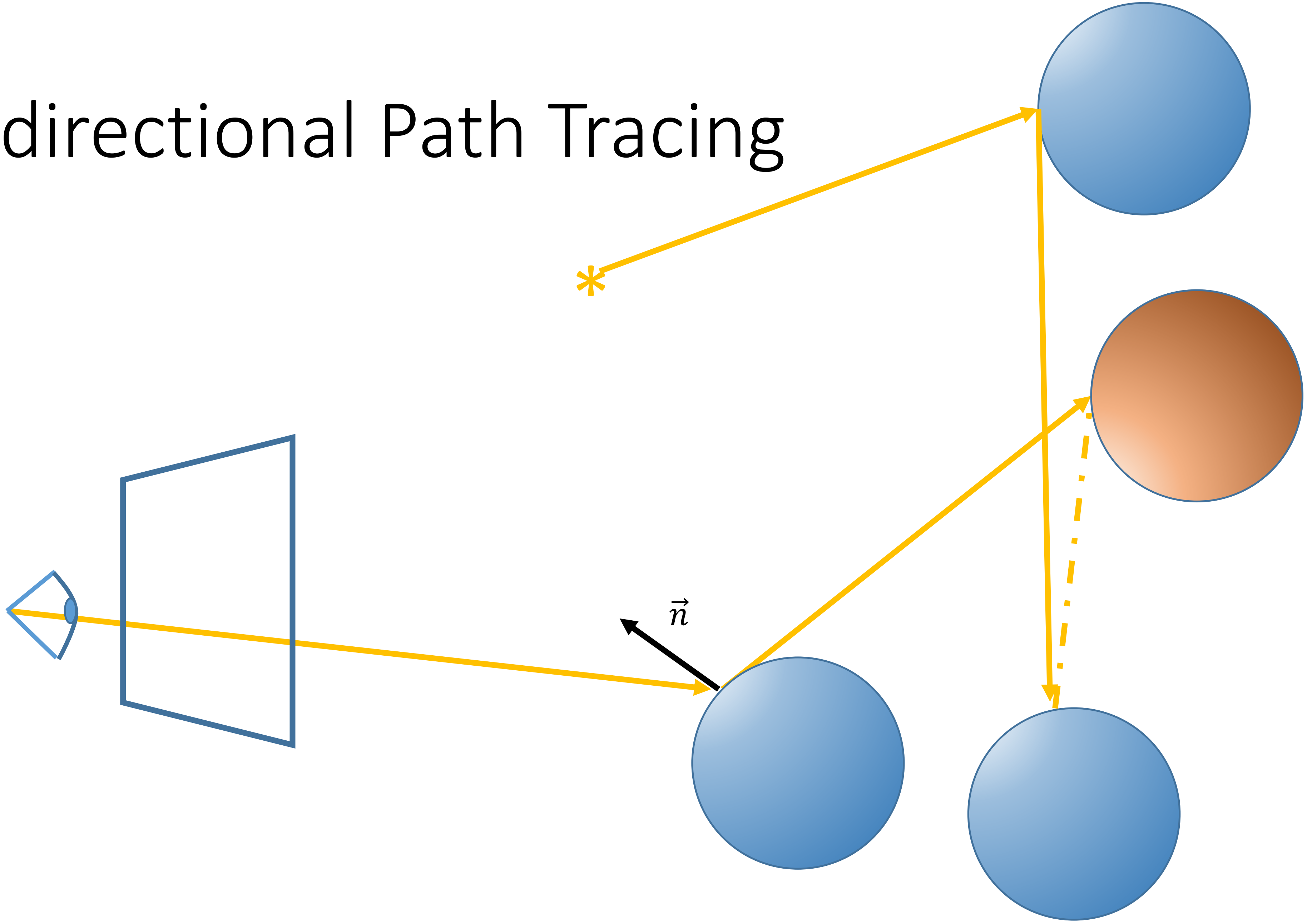
[Secord et al. 2002] "Fast Primitive Distribution for Illustration »

Other tricks

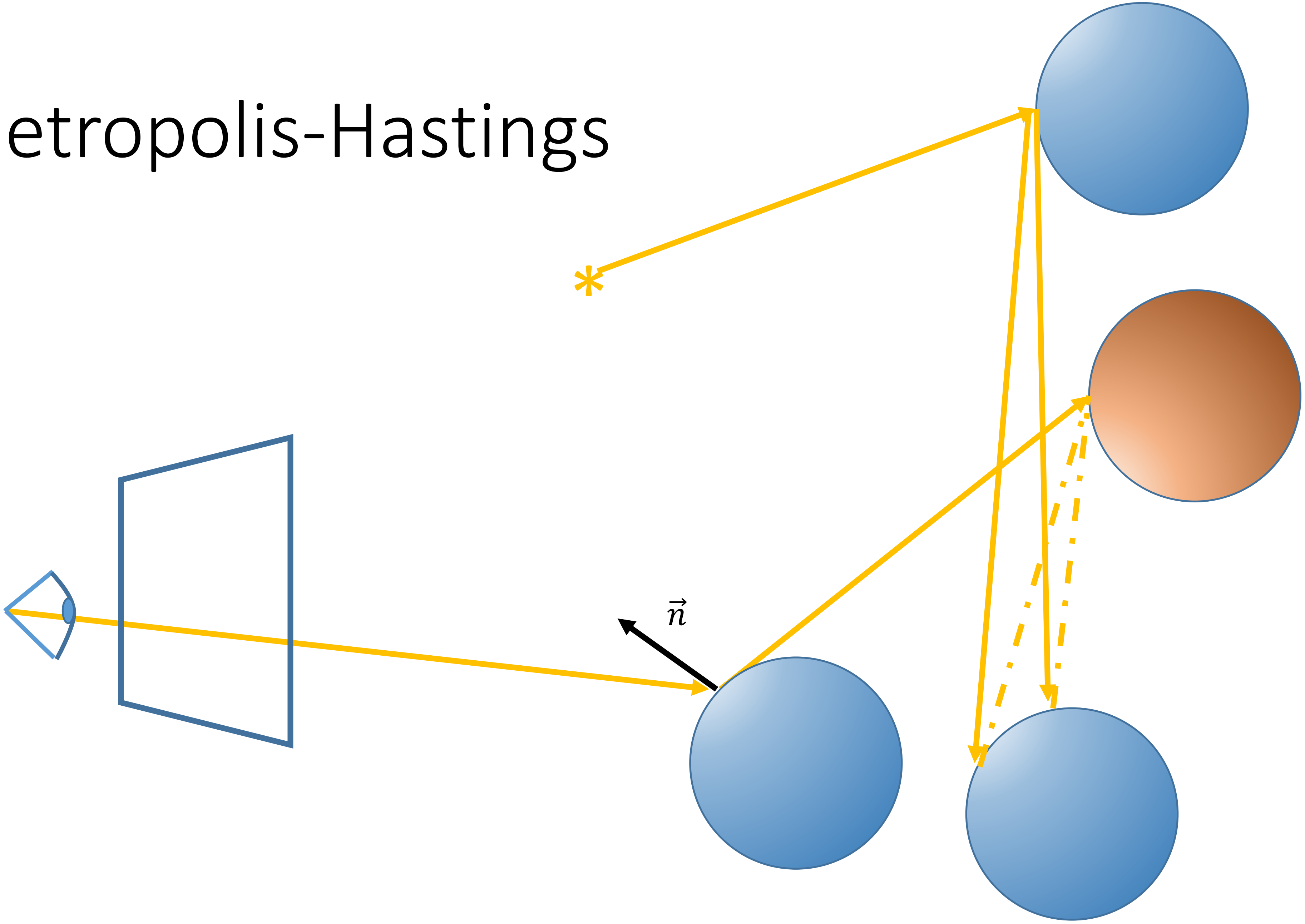
- Multiple Importance Sampling
  - Used for integrating with multiple strategies
  - Given estimates $\{I_k\}_{k=1..n}$ of an integral $\int f(x)dx$ using pdf $\{p_k\}_{k=1..n}$
    - $I = \sum_{k=1}^{n} w_k I_k$ with $\sum_{k=1}^{n} w_k = 1$ (naïve)
    - $I = \sum_{i=1}^{n} \sum_{j=1}^{n_i} \frac{f(x_{ij})}{\sum_{k=1}^{n} n_k \, p_k(x_{ij})}$ (balanced heuristic) : optimal*
- Control Variates
  - Used for integrating when the integral H of a proxy h is known
  - $I \approx \frac{1}{n} \sum_i \left(f(x_i) - h(x_i) \right) + H \quad \Rightarrow \quad Var(I) = \frac{1}{n} Var(f - h)$
  - Better : $I \approx \frac{1}{n} \sum_i \left(f(x_i) - \beta h(x_i) \right) + \beta H$ with $\beta = \frac{Cov(f,h)}{Var(h)}$

*Assuming a convex sum. See [Kondapaneni et al. 2019] “Optimal Multiple Importance Sampling” for better, possibly negative, weights

Bidirectional Path Tracing



Metropolis-Hastings

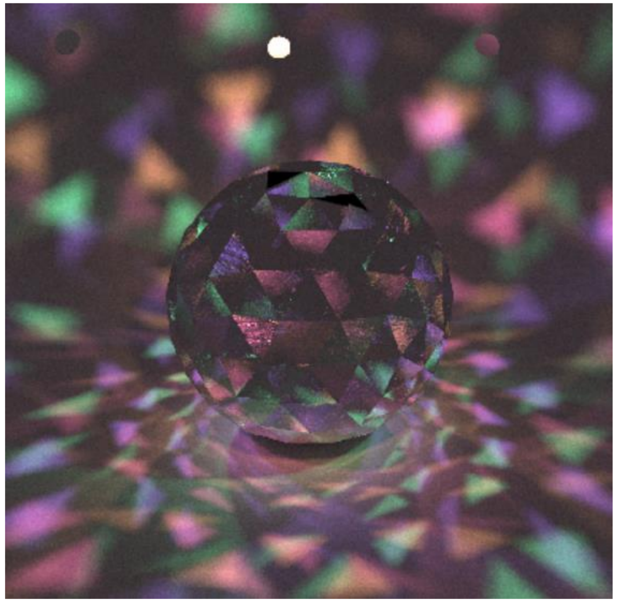


Metropolis-Hastings

- Probability P of accepting a new path p'
  - $P = \min(1, \frac{\pi(p')}{\pi(p)})$

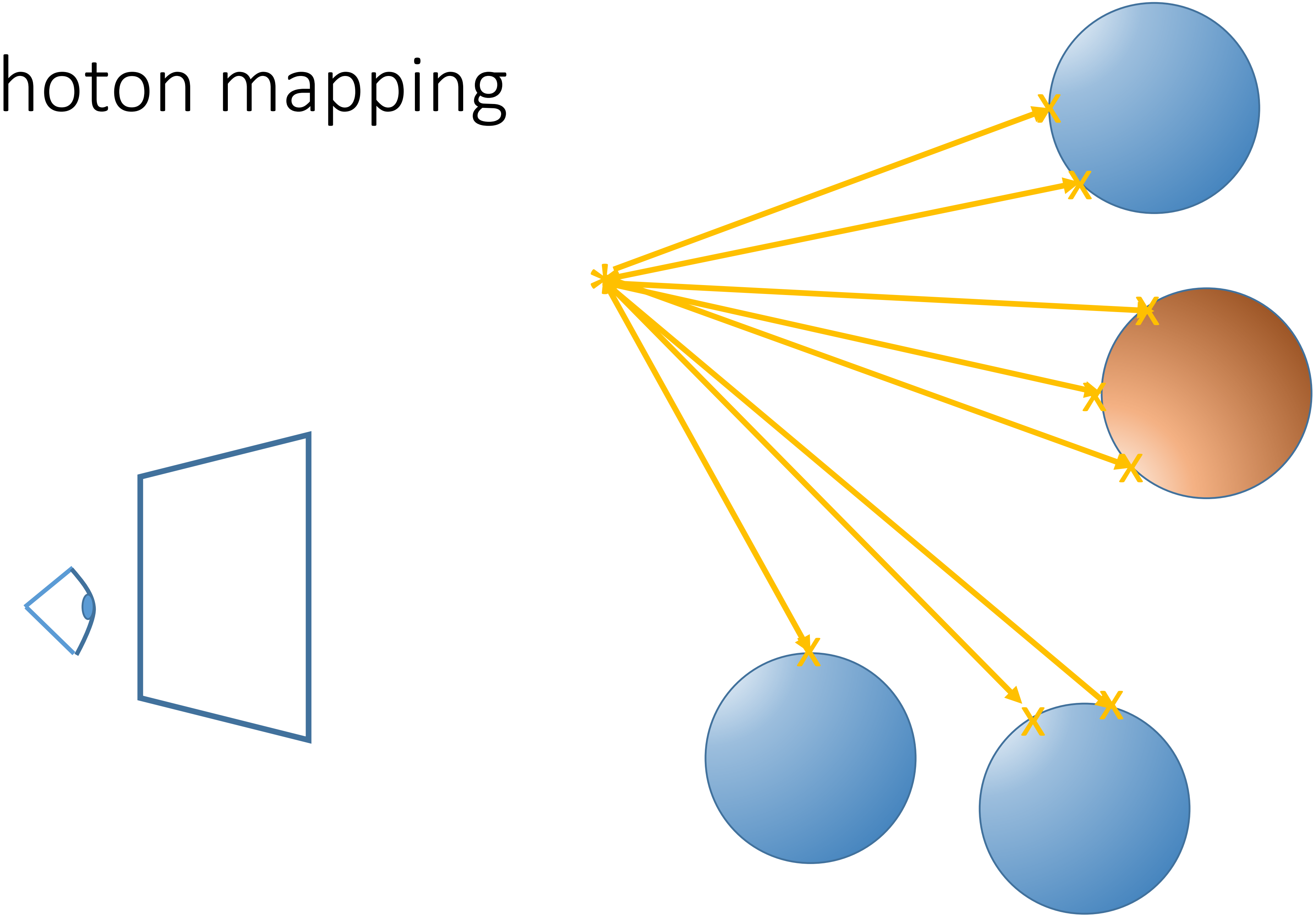
- Results in a sequence of paths following the distribution π

- Ideal when paths are hard to find
  - Specular paths
  - Refraction / caustics
  - Small holes letting light pass



A Simple and Robust Mutation Strategy for the Metropolis Light Transport

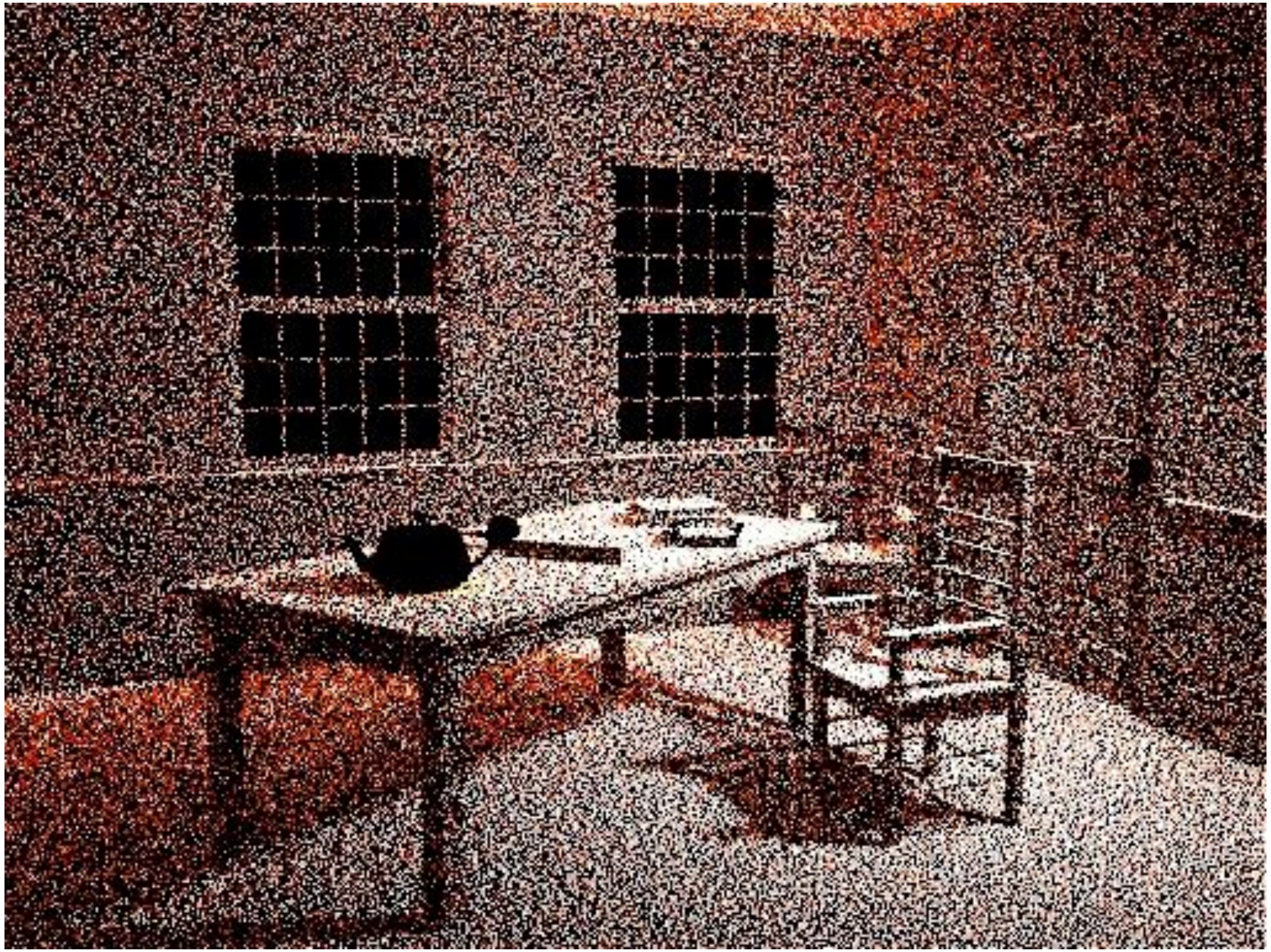
Photon mapping



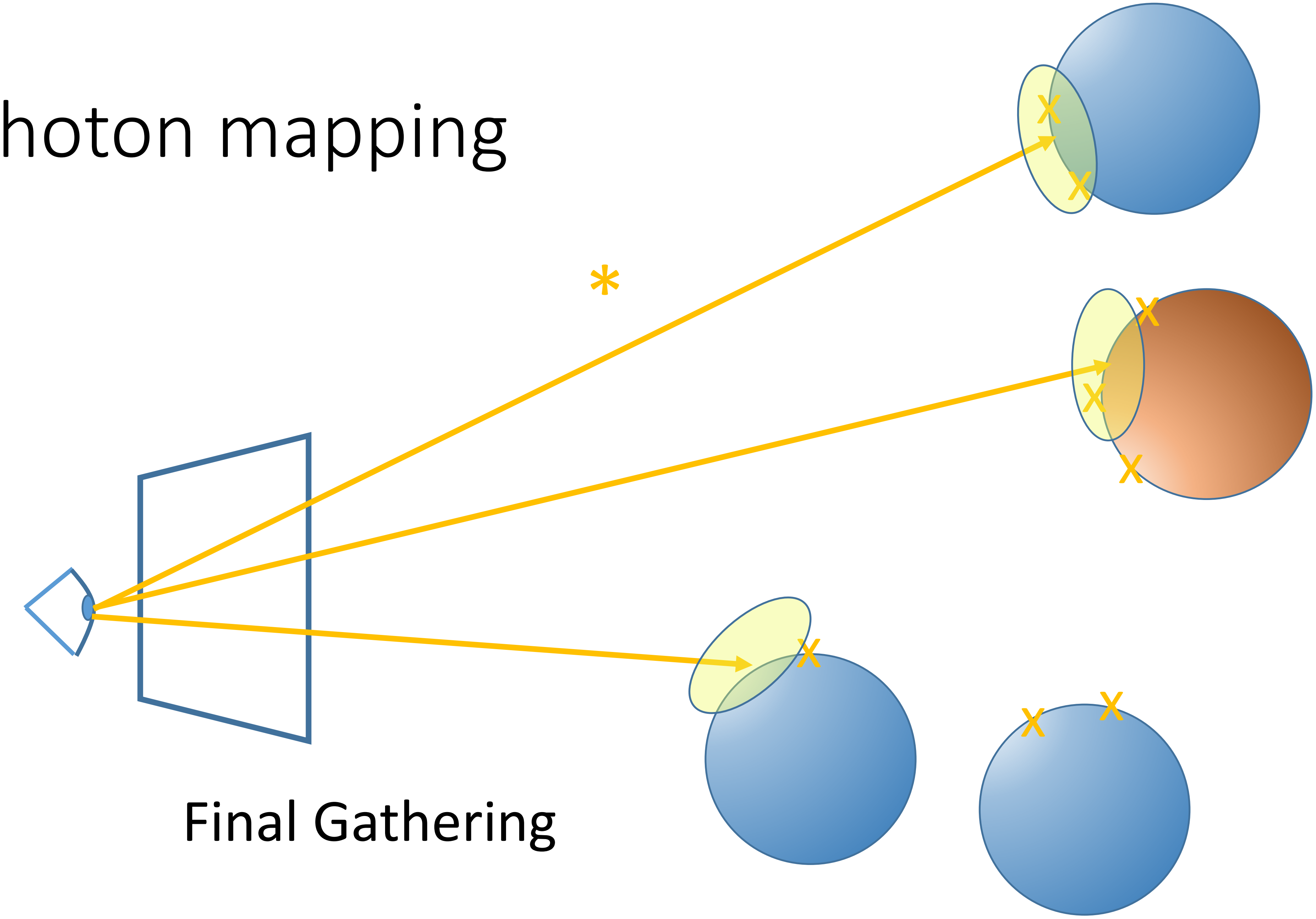
Faster Photon Map Global Illumination, PH Christensen

(only non purely specular objects receive photons)

Photon mapping



Photon mapping



Photon mapping

Final gathering step
(irradiance)

Photon mapping

- Density estimation
  - Fix a radius, count the number of photons inside
  - Fix a number of photons to get, look at the radius
  - Goal: obtain a number of photon per unit area
  - Can weigh photons with distance
- Bottleneck : Retrieving nearest neighbors
  - Acceleration structures (kd-trees, octrees...)
- Biased estimate
  - For a finite # photons, the expected value is not the true value
  - Due to the gathering of nearby (and incorrect) values
  - E.g., next to a shadow, a pixel is systematically darker

Photon mapping

Final gathering step
(radiance)

Noisy. Idea:
do it just for indirect

Photon mapping

Purely direct lighting

Photon mapping

Sum

Precomputed Radiance Transfer

$$L_o(x, \overrightarrow{\omega_o}) = L_e(x, \overrightarrow{\omega_o}) + \int_\Omega f(\overrightarrow{\omega_i}, \overrightarrow{\omega_o}) L_i(x, \overrightarrow{\omega_i}) \langle \overrightarrow{\omega_i}, \vec{n} \rangle d\overrightarrow{\omega_i}$$

- Decompose $f(\overrightarrow{\omega_i}, \overrightarrow{\omega_o})$ $\langle \overrightarrow{\omega_i}, \vec{n} \rangle$ and $L_i(\overrightarrow{\omega_i})$ on orthonormal bases
- Use a scalar product $\langle f, g \rangle = \int f(x) g(x) dx$
- If $f(x) = \sum \alpha_i S_i(x)$ and $g(x) = \sum \beta_j S_j(x)$ and $\{S_i\}$ is orthonormal

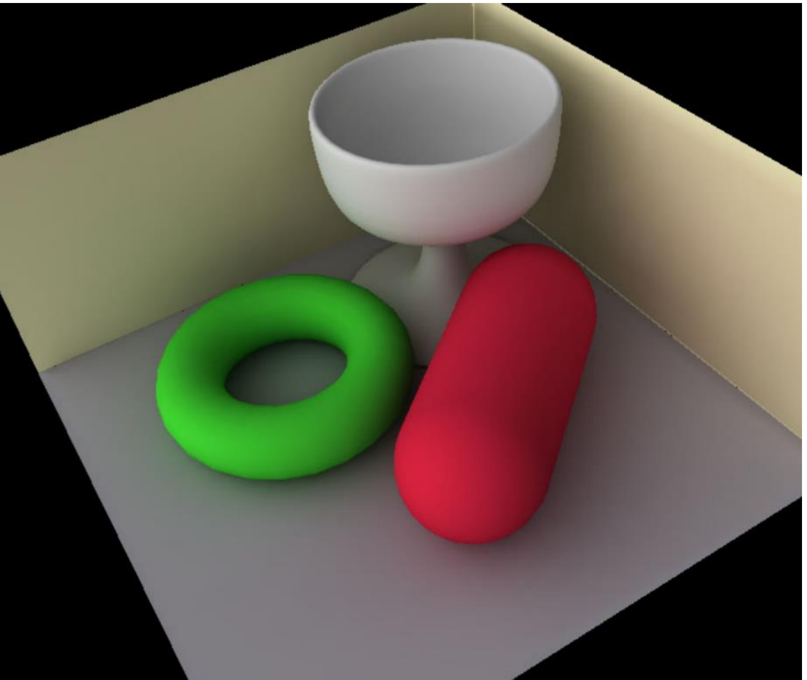
$$\langle f, g \rangle = \langle \alpha, \beta \rangle$$

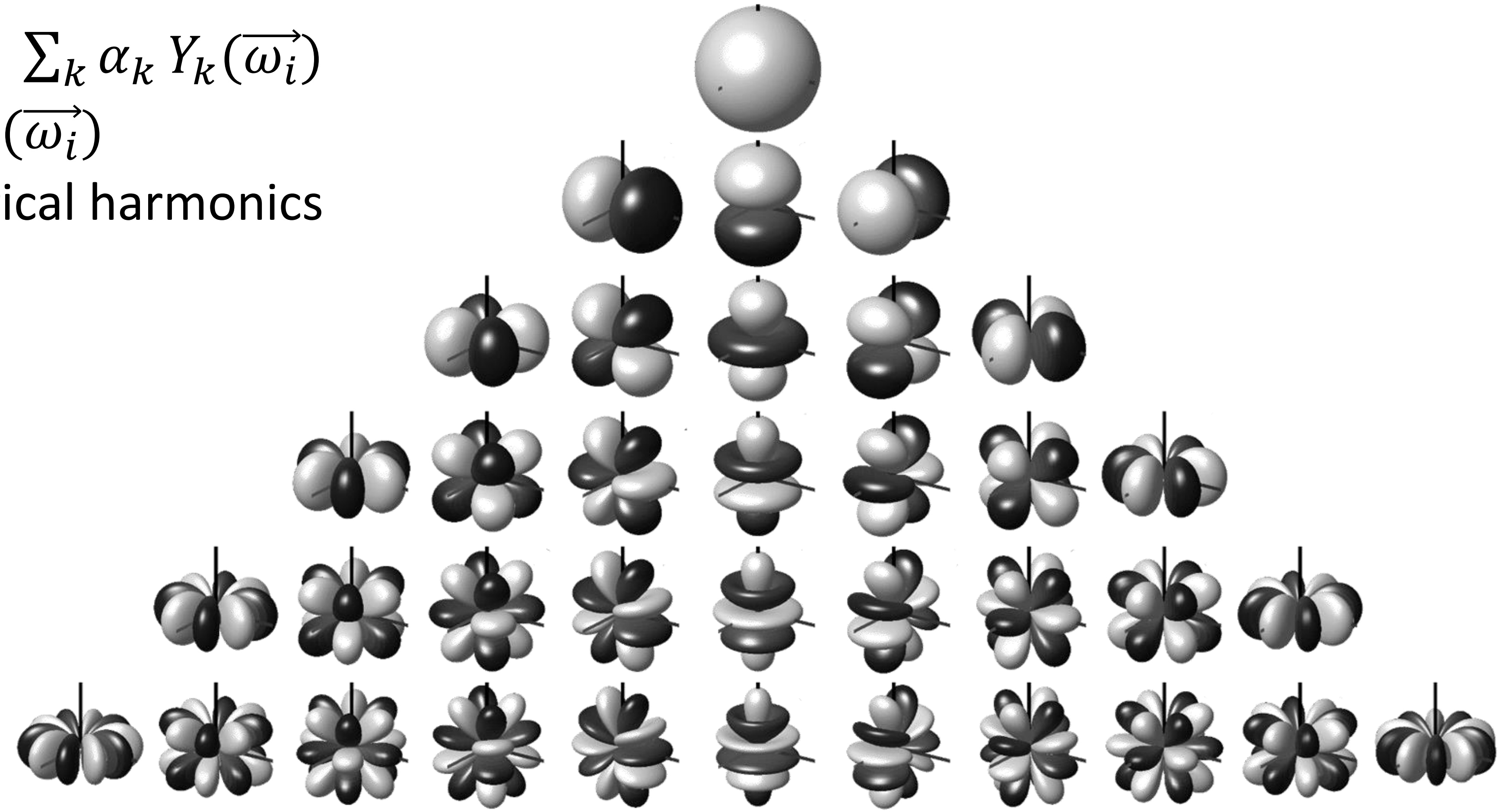
(proof by bilinearity of scalar product)

Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments, PP Sloan et al.

Precomputed Radiance Transfer

- For instance:
  - $f_{\overrightarrow{\omega_o}}(\overrightarrow{\omega_i})\,\langle\overrightarrow{\omega_i},\vec{n}\rangle = \sum_k \alpha_k\, Y_k(\overrightarrow{\omega_i})$
  - $L_i(\overrightarrow{\omega_i}) = \sum_k \beta_k\, Y_k(\overrightarrow{\omega_i})$
  - With $Y_k(\overrightarrow{\omega_i})$ spherical harmonics





Precomputed Radiance Transfer

- $Y_k(\overrightarrow{\omega_i}) = \alpha\, P_l^m(\cos\theta)e^{im\phi}$
  - With $P_l^m(x)$ the associated Legendre polynomial of order m (related to the m'th derivative of P_l)
  - Eigenfunctions of the Δ operator on the sphere
  - Equivalent to Fourier basis on the sphere
  - Fast computation via 2D Fast Fourier Transforms

- The frequency content of the result is bounded by the minimum frequency content between the BRDF and illumination !
  - E.g. : a diffuse object under high frequency lighting looks the same as a metal ball under diffuse (constant) lighting

- Other bases have been used: Spherical Wavelets, Zonal Harmonics, ...

Wavelets

- Haar wavelet
  - $\psi_{n,k}(t) = 2^{n/2}\,\psi(2^n t - k)$

    with $\psi(t) = 1_{t\in[0,\frac{1}{2}]} - 1_{t\in\left[\frac{1}{2},1\right]}$

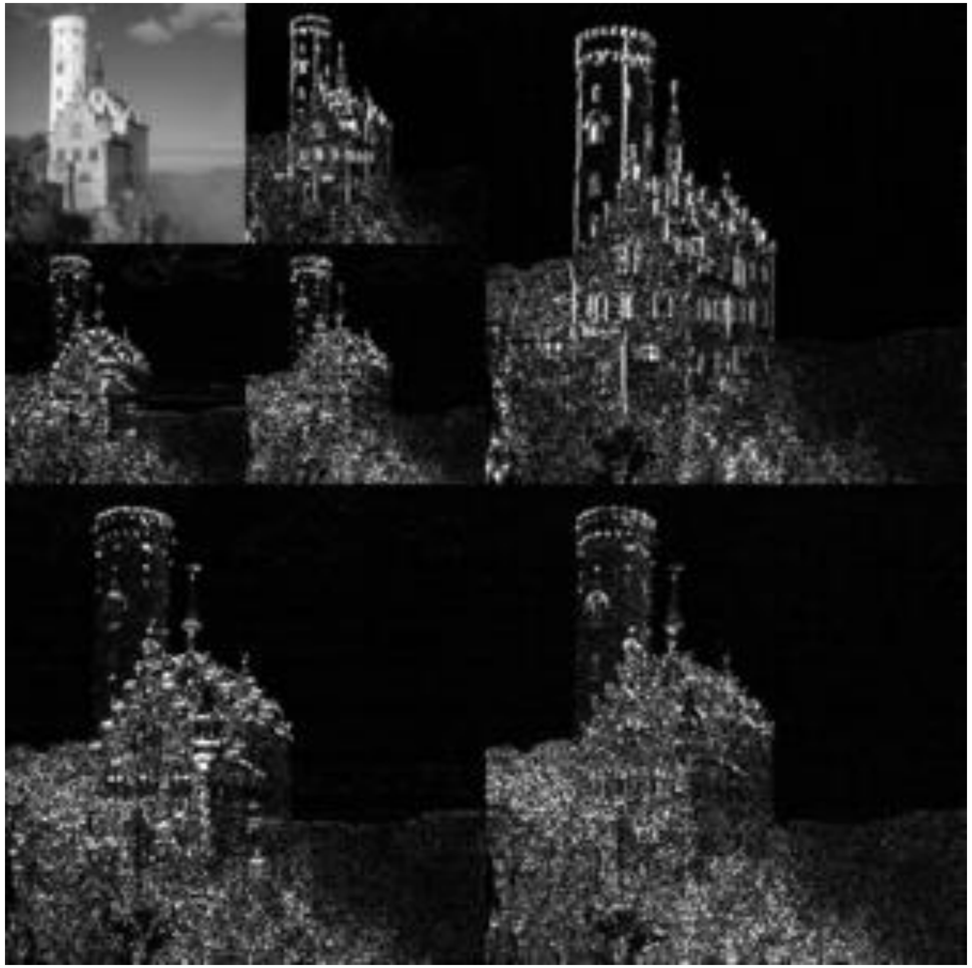
    and $k, n \in Z$
- Orthogonal:
  - $\int_R \psi_{k_1,n_1}(t)\,\psi_{k_2,n_2}(t)dt = \delta_{k_1,k_2}\delta_{n_1,n_2}$

Wavelets

- Wavelet transform
  - $X(k,n) = \int_R \psi_{k,n}(t)\, x(t)dt$
  - Various translations k => convolution
  - In fact:

$$X_n(k) = \; 2^{n/2} \int_R \psi(2^n t - k)\;\; x(t)dt$$

- Haar scaling function
  - $\phi(t) = 1_{t\in[0,1]}$
  - Expresses residual low frequencies



Wavelets

- In fact, recursive formulation of Haar wavelets:
  - Scaling function : $\phi(t) = \phi(2t) + \phi(2t-1)$
  - Wavelet: $\psi(t) = \phi(2t) - \phi(2t-1)$ (no typo!)

  - Given $\chi(k,n) = 2^{n/2} \int_R x(t)\phi(2^n t - k)dt$

  and $X(k,n) = 2^{n/2} \int_R x(t)\,\psi(2^n t - k)\,dt$

  We recursively obtain:

$$\chi(k,n) = 2^{-\frac{1}{2}}(\chi(2k,n+1) + \chi(2k+1,n+1))$$

Box filter

$$X(k,n) = 2^{-\frac{1}{2}}(\chi(2k,n+1) - \chi(2k+1,n+1))$$

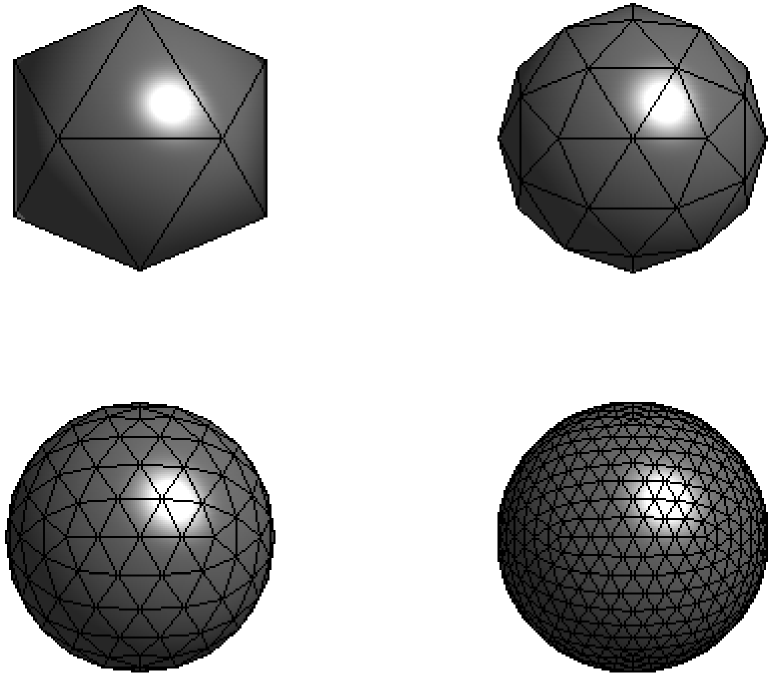
Finite difference

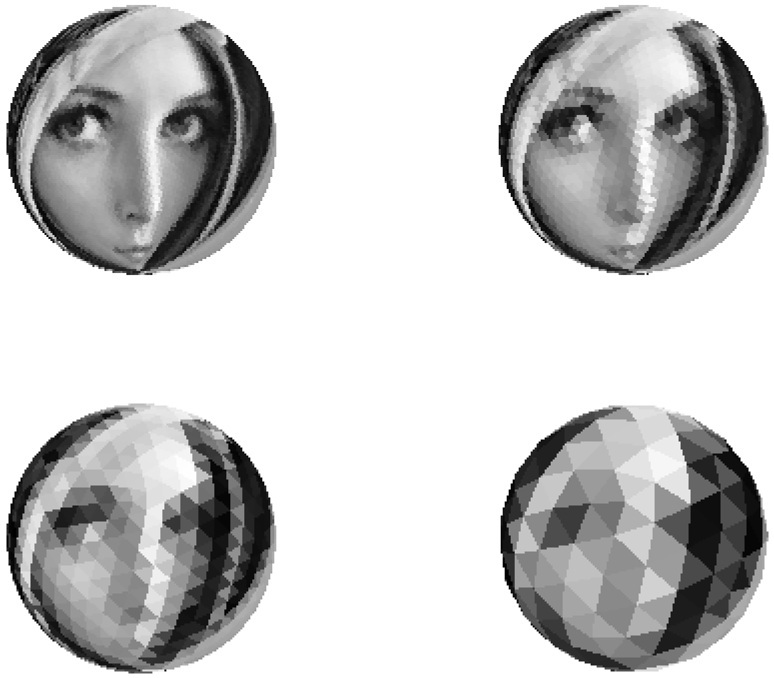
Only depends on scaling function!

Example at:
https://www.eecis.udel.edu/~amer/CISC651/Haar.wavelets.paper.by.Mulcahy.pdf

Spherical Wavelets

- Same concept on the sphere
- Scaling function defined as piecewise constant on tessellated spheres



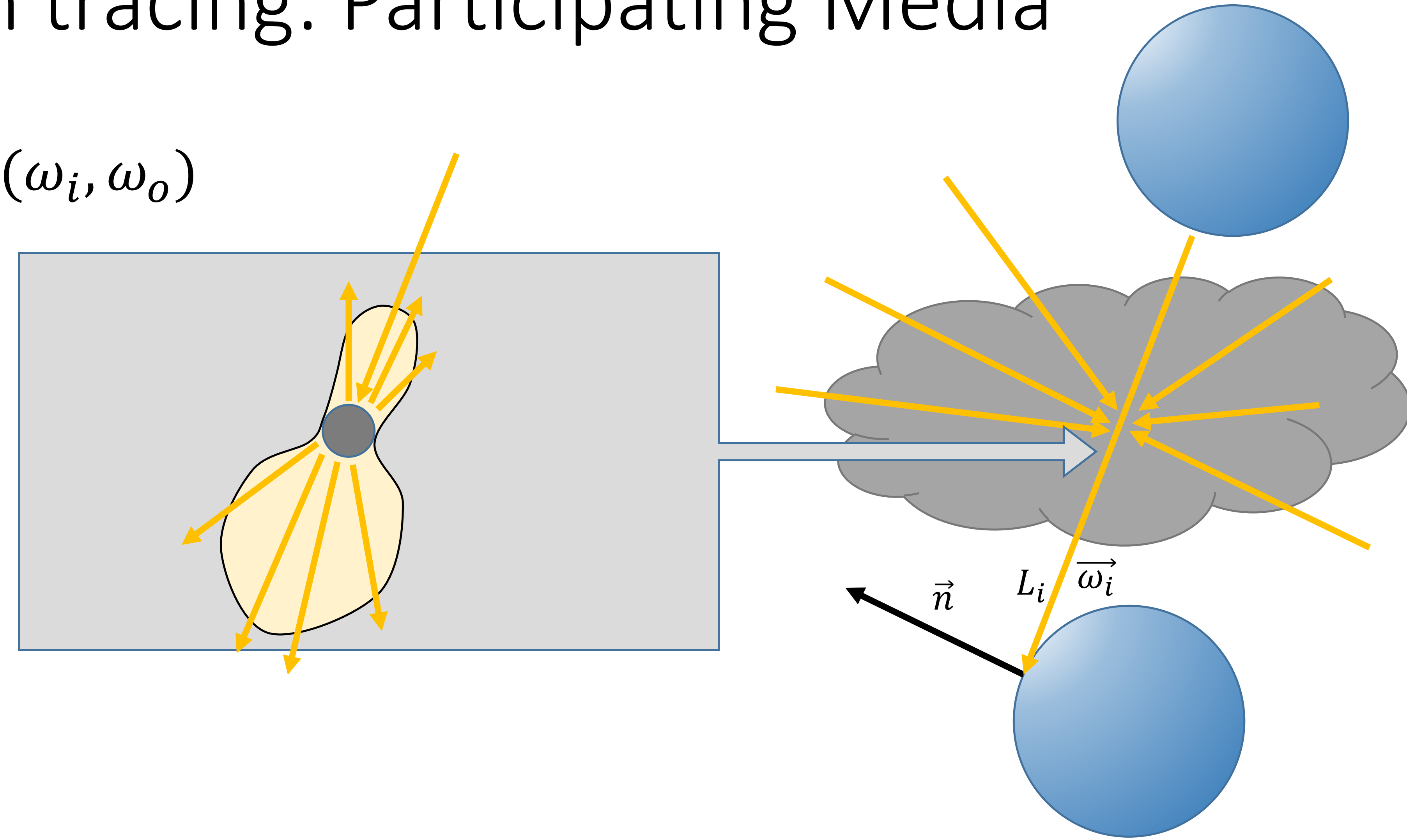


(low passed only)

Images from Gabriel Peyre’s “Numerical Tours” : check these tours out!

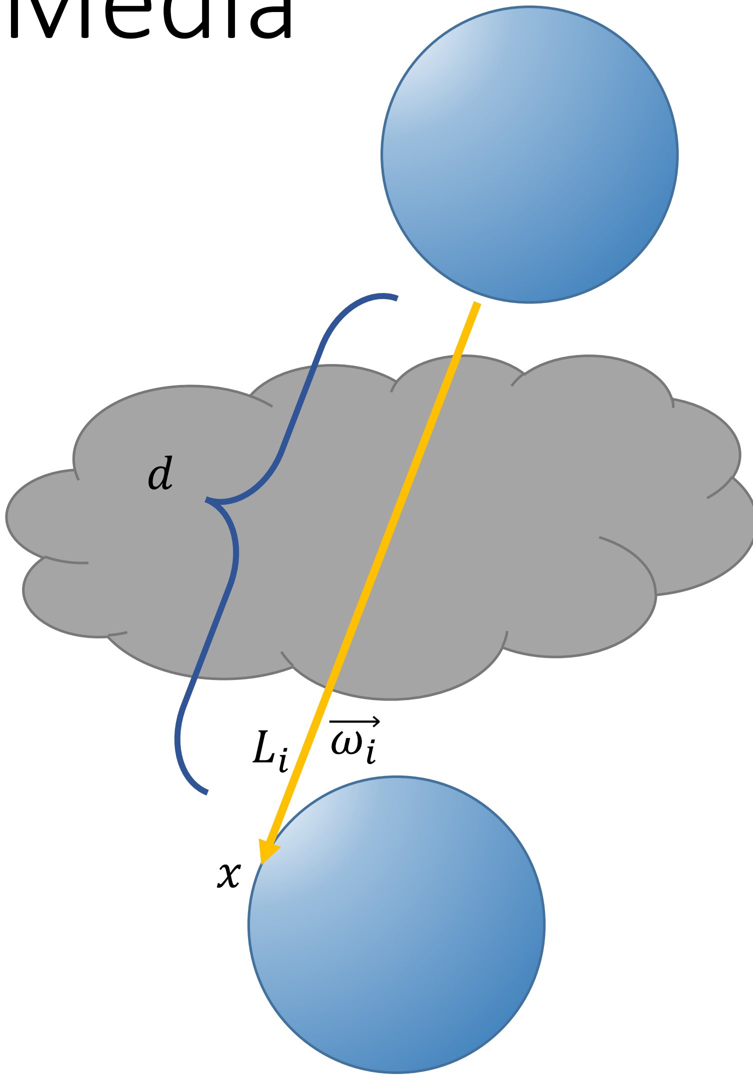
Back to path tracing: Participating Media

- Phase function $f(\omega_i, \omega_o)$



Back to path tracing: Participating Media

- Extinction coefficient σ_t
  - density of the medium
  - Optical depth: $\tau(d) = \int_0^d \sigma_t(x - t\omega_i)\, dt$
  - Transmittance: $\mathrm{T}(\mathrm{d}) = \exp(-\tau(d))$
    defines how much light is absorbed or scattered out



Back to path tracing: Participating Media

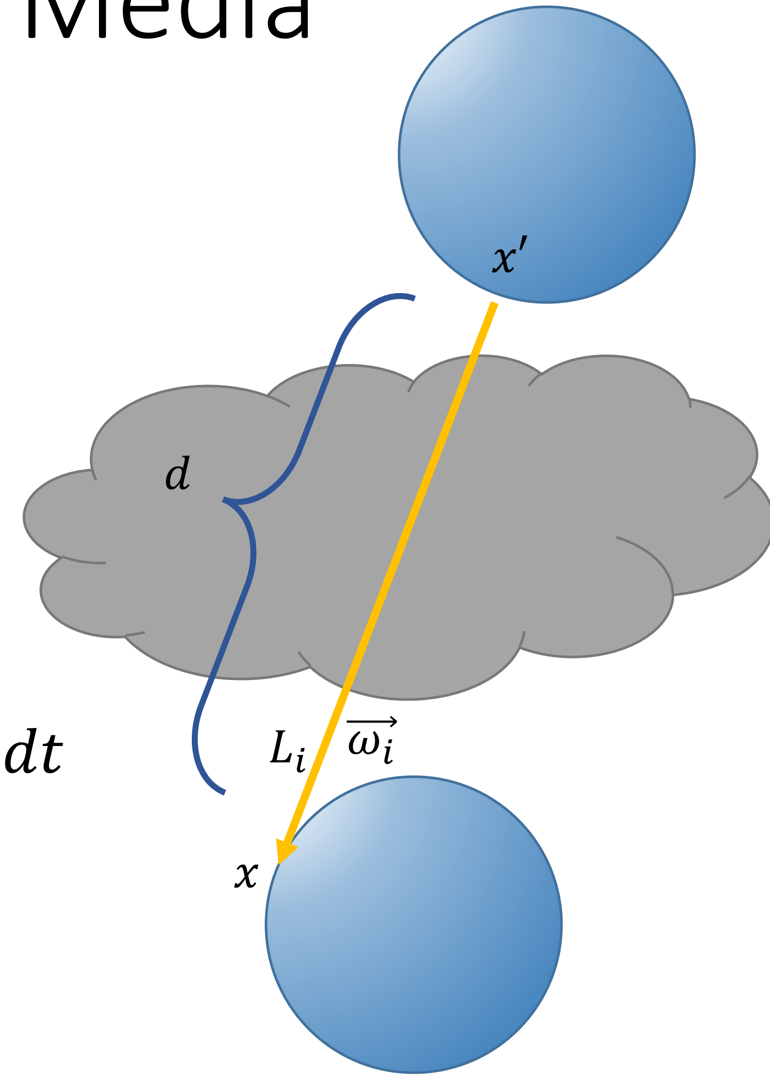
- Scattering coefficient σ_s
- Absorption coefficient σ_a
- $\sigma_t = \sigma_s + \sigma_a$

- $L_i(x, \omega_i) = L_i(x', \omega_i).\mathrm{T(d)}$

$$+ \int_0^d \int_{\Omega^+} T(t) f(\omega_i, \omega_o)\, L_i(x_t, \omega_o) \sigma_s(x_t) d\omega_o dt$$

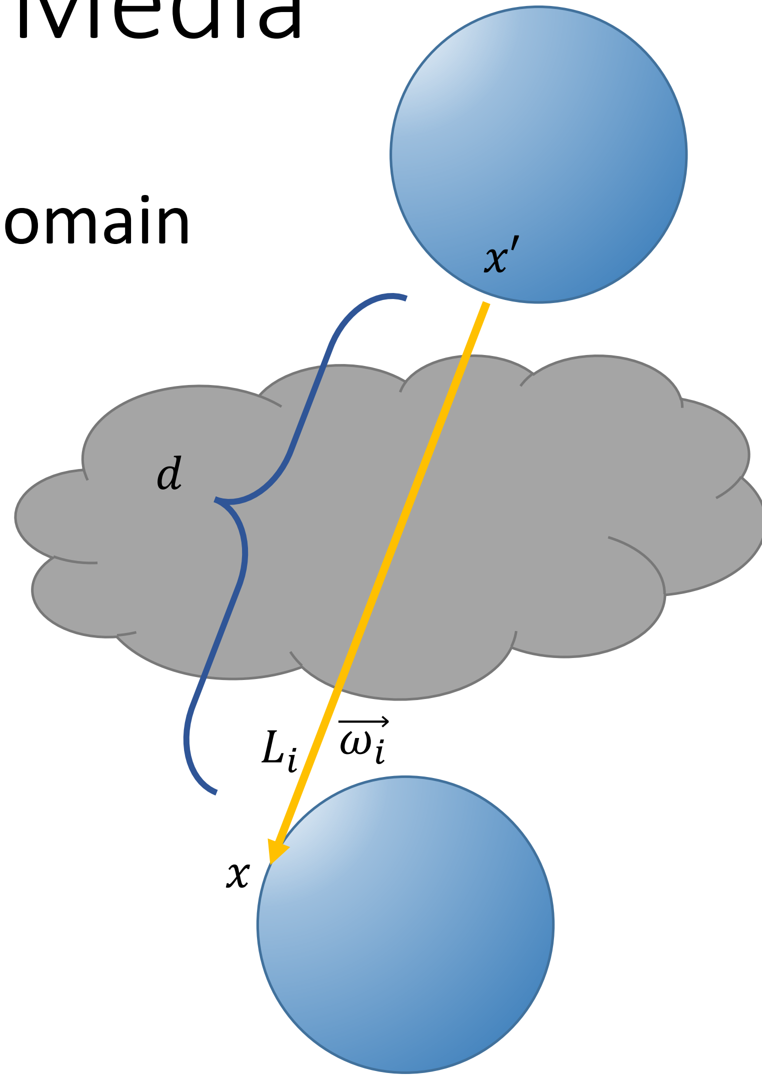
With $x_t = x - \omega_o t$

(loose notations)



Back to path tracing: Participating Media

- We merely added 3 dimensions to the integration domain
  - Absorb the incoming light
  - Add in-scattered radiance by
    - Sampling one position
    - Sampling one direction
    - Adding the contribution $T(t)f(\omega_i, \omega_o)L_i(x_t, \omega_o)\sigma_s(x_t)$



Back to path tracing: Participating Media

Tutorial : http://liris.cnrs.fr/~nbonneel/teaching.html

Radiosity

Form factor:

$$G(x, x')$$

$$L_o(x, \overrightarrow{\omega_o}) = L_e(x, \overrightarrow{\omega_o}) + \int_{\mathrm{P}} f(\overrightarrow{\omega_i}, \overrightarrow{\omega_o}) L_i(x, \overrightarrow{\omega_i}) \boxed{\langle \overrightarrow{\omega_i}, \vec{n} \rangle \frac{V(x, x') \left| \langle \overrightarrow{\omega_i}, \vec{n}' \rangle \right|}{\| x - x' \|^2}} dP$$

- Under diffuse reflectance, we have $f(\overrightarrow{\omega_i}, \overrightarrow{\omega_o}) = \frac{\rho}{\pi}$
- And omnidirectional emissivity
- So :

$$L_o(x) = L_e(x) + \frac{\rho(x)}{\pi} \int_{\mathrm{P}} L_i(x, \overrightarrow{\omega_i}) G(x, x') dP$$

Radiosity

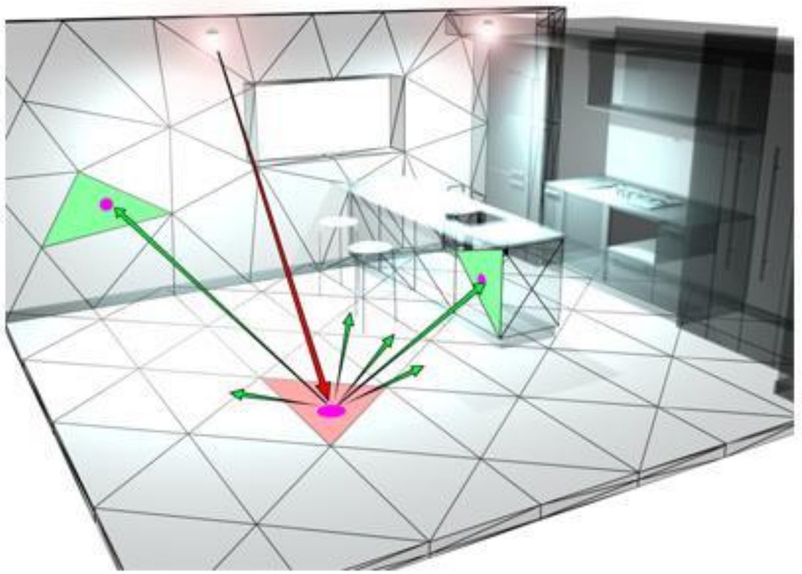
$$L_o(x) = L_e(x) + \frac{\rho(x)}{\pi} \int_{\mathrm{P}} L_i(x, \overrightarrow{\omega_i}) G(x, x') dP$$

- Now, discretizing and assuming constant values per triangle k:

$$L^k = L_e^k + \frac{\rho^k}{\pi} \sum_l L^l \, G^{k,l}$$

(could also take any orthogonal basis function over triangles instead)

Radiosity



$$L^k = L_e^k + \frac{\rho^k}{\pi} \sum_l L^l \, G^{k,l}$$

- Can be written in matrix form. Consider a vector L and matrix G :

$$L = L_e + diag \frac{\rho}{\pi} \, G \, L$$

Re-arranging terms:

$$L = \underbrace{\boxed{\left(Id \; - \; diag \frac{\rho}{\pi} \, G \right)}}_{M}{}^{-1} L_e$$

Can be solved numerically quite easily

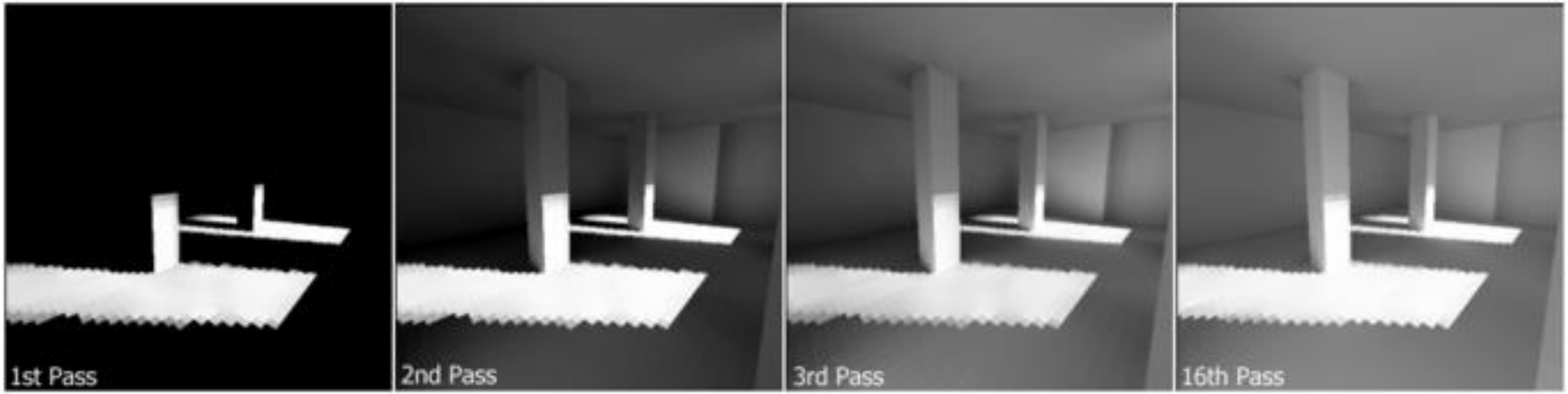
Radiosity

- Instead of a full direct linear solve, use Jacobi iterations

One iteration:

$$L_i = \frac{1}{M_{ii}}\left(L_e^i - \sum_{\substack{j=1 \\ j\neq i}}^{n} M_{ij}\, L_j\right) \qquad \forall i$$

  - Each iteration corresponds to 1 light bounce:



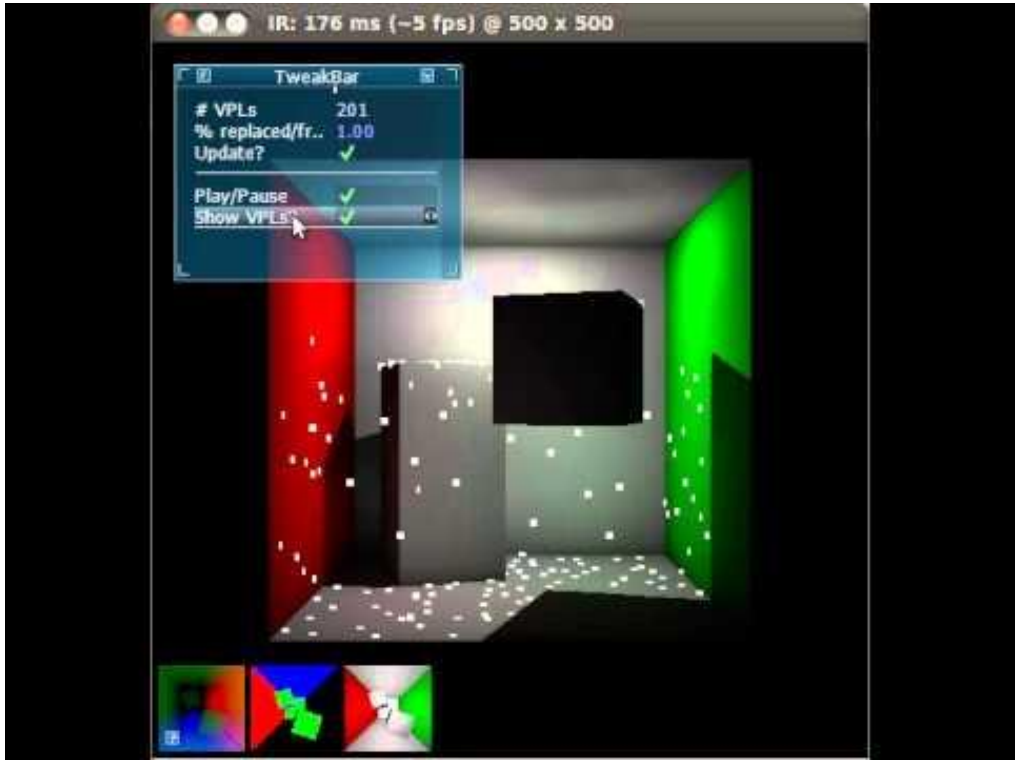
Radiosity

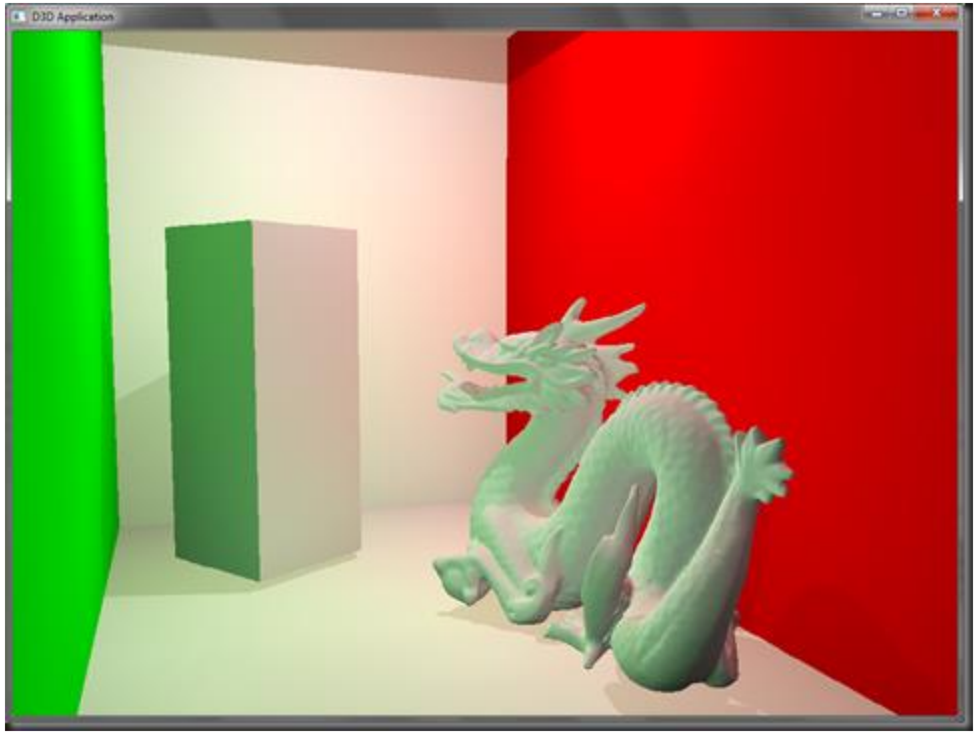
- Unfortunately, now mostly abandoned
  - Has been generalized to non-diffuse scenes
  - To (near) realtime settings*
  - But nice meshing is difficult
  - Conceptually simpler methods exist (e.g., photon maps)

* Implicit Visibility and Antiradiance for Interactive Global Illumination, Dachsbacher et al.

Physically-based rendering meets realtime

- Instant radiosity
  - Essentially unrelated to radiosity, but more related to photon mapping
  - Sends “Virtual Point Lights” from light sources, use them as new light sources



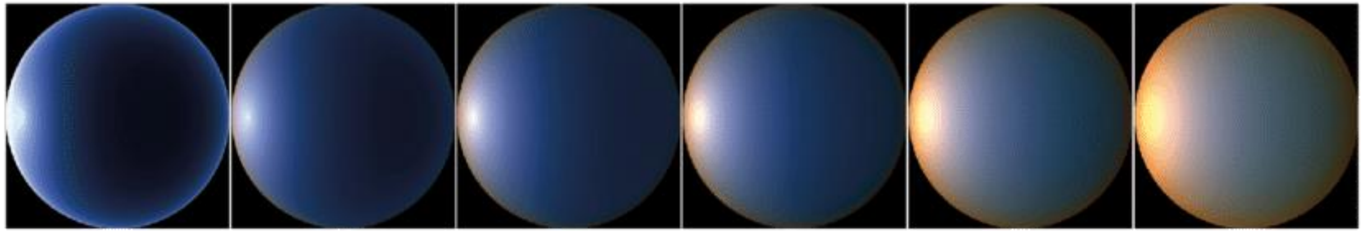


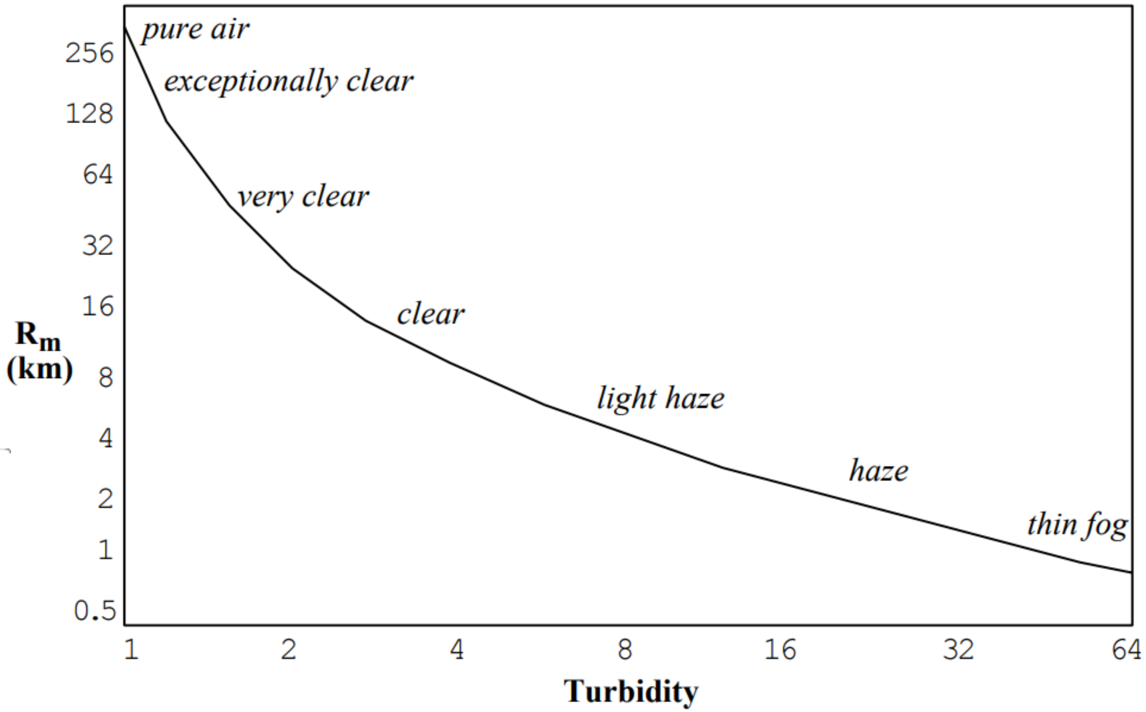
Distant illumination models

- Environment maps

Distant illumination models

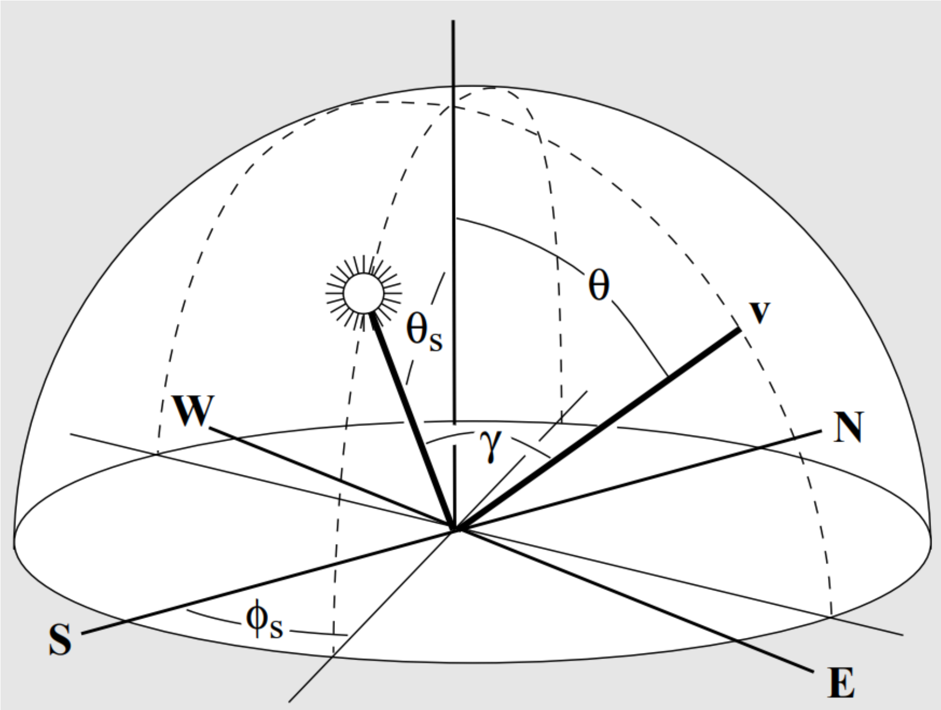
- Analytic Sky Model [Preetham et al. 1999] : parametric sky model
  - Turbidity: optical thickness of atmosphere including haze / optical thickness of atmosphere without haze





Distant illumination models

- Analytic Sky Model [Preetham et al. 1999] : parametric sky model
  - Skylight luminance from Perez et al.
    - $F(\theta,\gamma) = \left(1 + Ae^{-\frac{B}{\cos\theta}}\right)\left(1 + Ce^{D\gamma} + E\cos^2\gamma\right)$
  - Skylight chrominance
    - $x = x_z \frac{F(\theta,\gamma)}{F(0,\theta_s)}$ $\quad y = y_z \frac{F(\theta,\gamma)}{F(0,\theta_s)}$
    - Parameters different for x and y
    - Fitted from measurements



Results

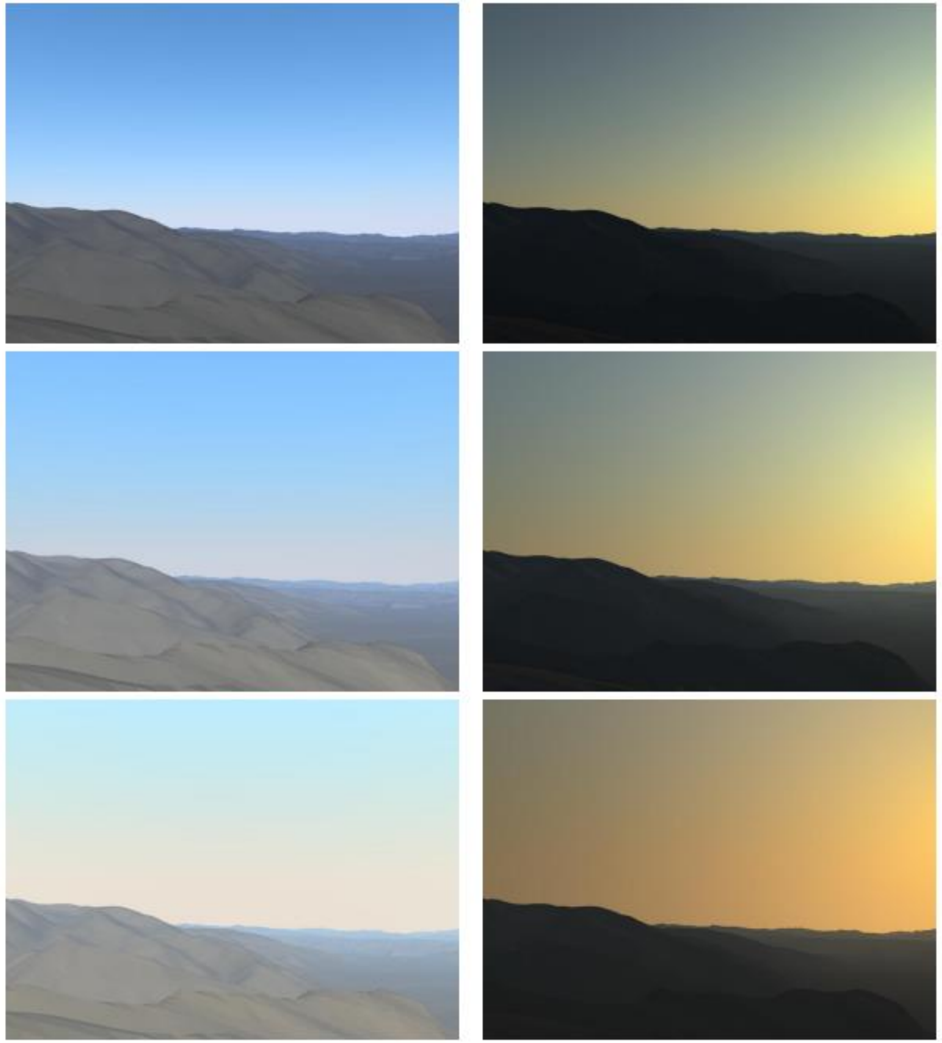


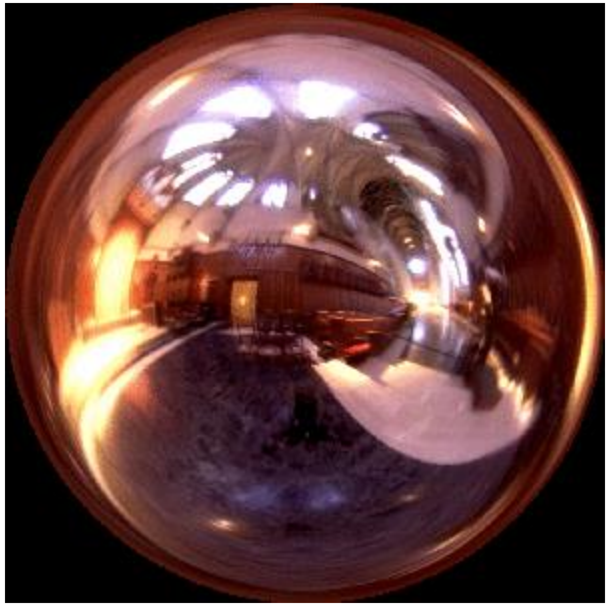
Figure 9: *The new model looking west at different times (left morning and right evening) and different turbidities (2, 3, and 6 top to bottom).*

Prefiltered environment maps

$$L_o(x,\overrightarrow{\omega_o}) = L_e(x,\overrightarrow{\omega_o}) + \underbrace{\int_\Omega f(\overrightarrow{\omega_i},\overrightarrow{\omega_o}) L_i(x,\overrightarrow{\omega_i}) \langle \overrightarrow{\omega_i},\vec{n}\rangle d\overrightarrow{\omega_i}}_{}$$

Spherical convolution between $f(.,\overrightarrow{\omega_o})\langle .,\vec{n}\rangle$ and L_i

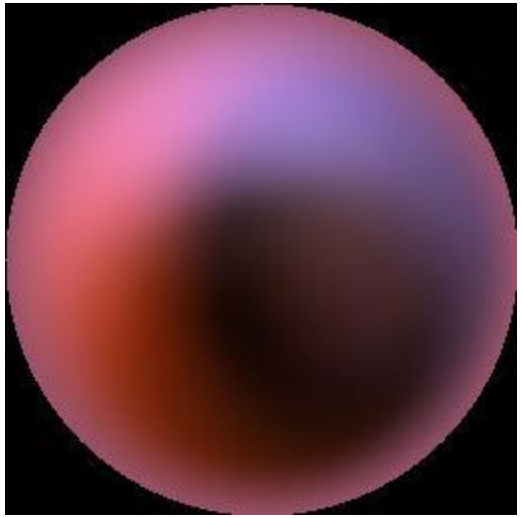
When the incident illumination is distant:

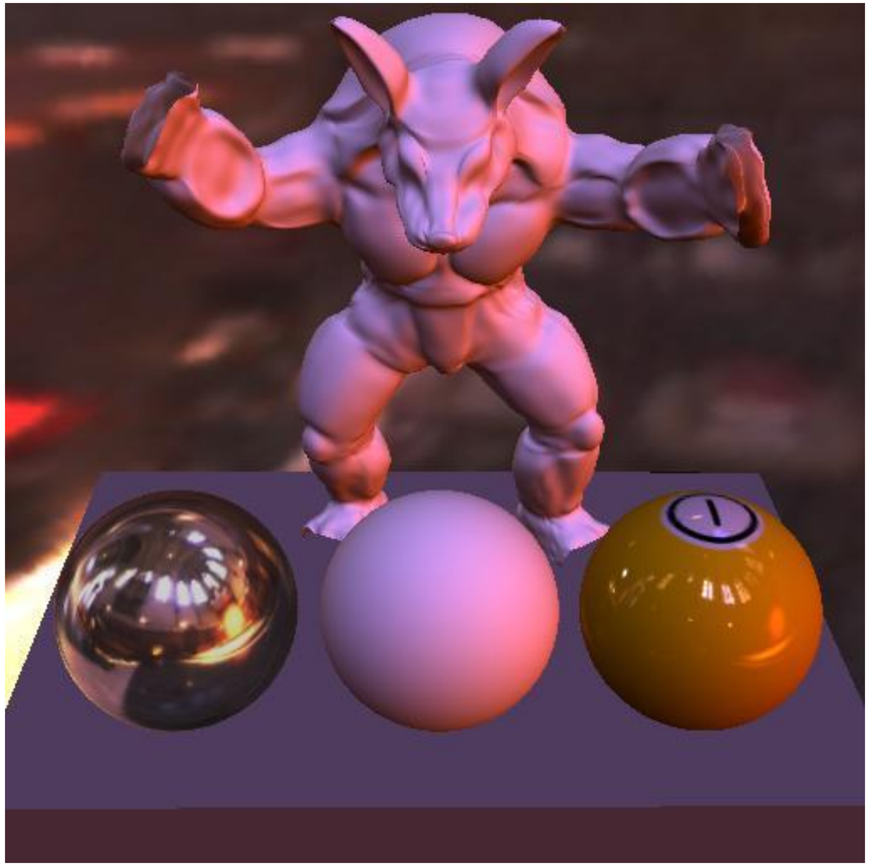


A Unified Approach to Prefiltered Environment Maps, Kautz et al.

Prefiltered environment maps

- Precomputing convolutions between environment map and BRDF for various ω_o
- Easy for some BRDF : Gaussian blur
- Just a lookup when rendering:





Ambient occlusion

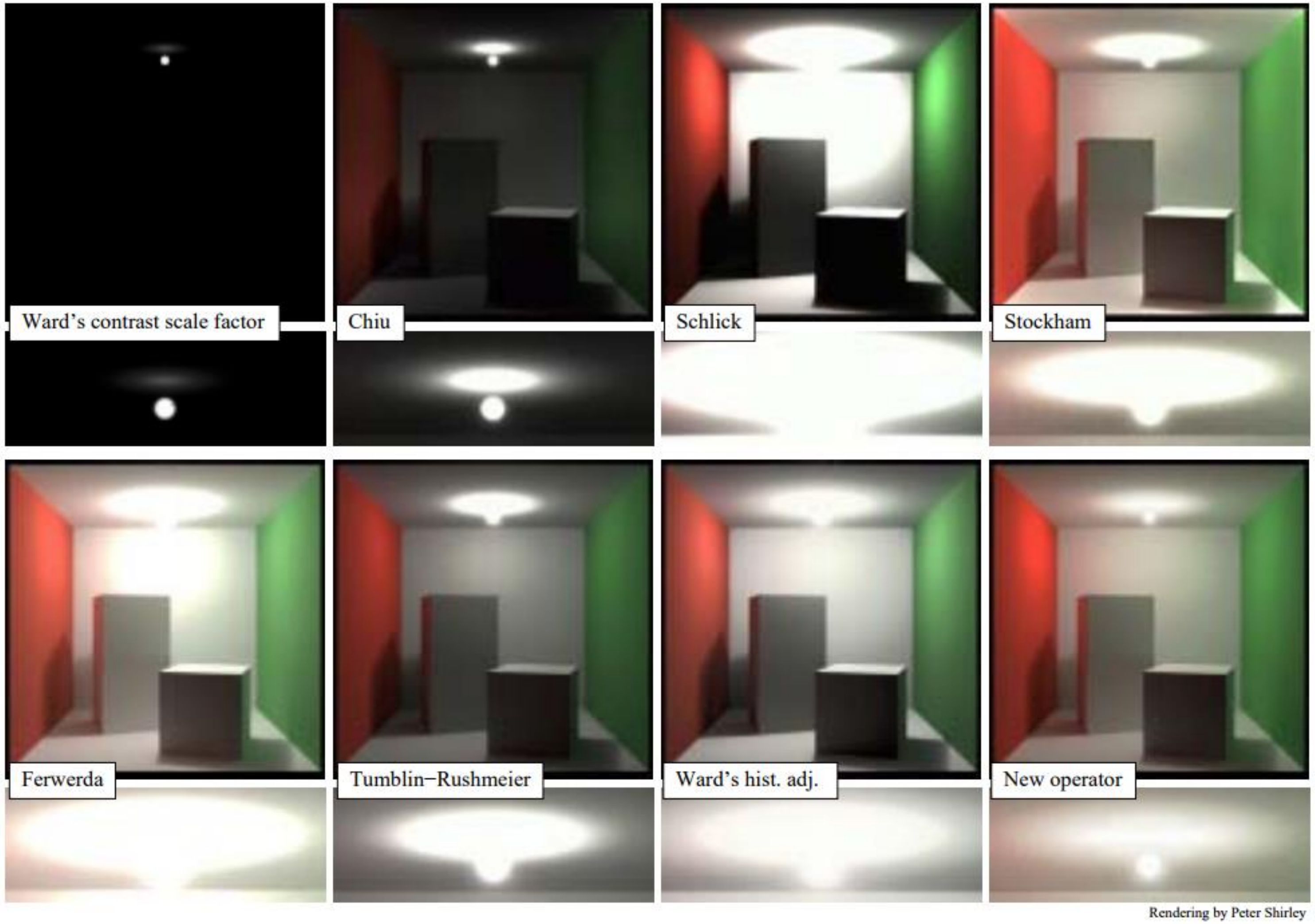
Original model

With ambient occlusion

Extracted ambient occlusion map

- Idea: precompute occlusion as
  - $O = \frac{1}{\pi} \int_{\Omega} V(\omega) . < \omega, n > d\omega$
  - Does not depend on the illumination
  - Often computed per object
    - Does not require raytracing the entire scene
    - Can be used for animated objects
  - Another option: screen space ambient occlusion
    - Does not trace rays in the scene: samples a sphere around fragments
    - More for realtime rendering

Tone Mapping



[Reinhard et al. 2002] « Photographic Tone Reproduction for Digital Images »

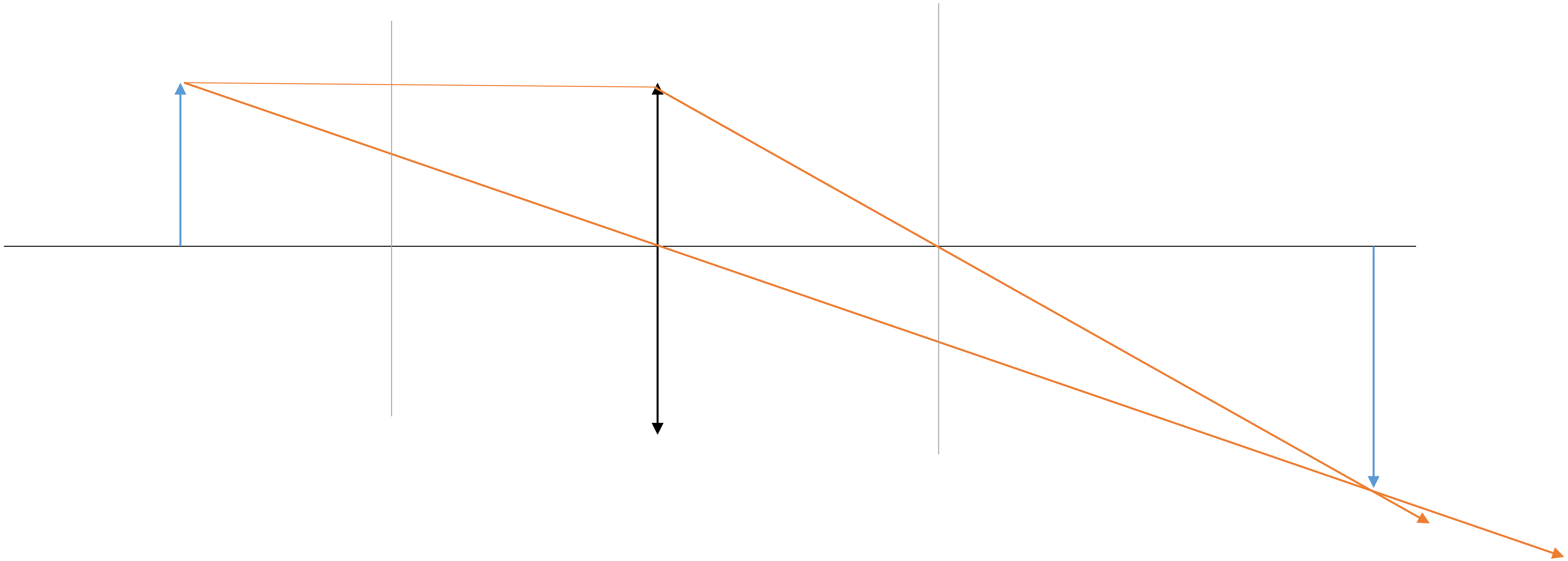
Tone Mapping

- Scene key: $\bar{L} = \frac{1}{N}\exp(\sum \log(\delta + I))$
- Adjusting image values: $L = \frac{0{,}18}{\bar{L}}\ I$
- Compressing highlights: $L_d = L\,\frac{1+\frac{L}{L_w^2}}{1+L}$

- If not sufficient, perform local adjustments:
  - Estimate contrast via difference of gaussian convolution at different scales
  - Locally find the smallest scale that produces low contrast
  - Adjust highlight compression step accordingly

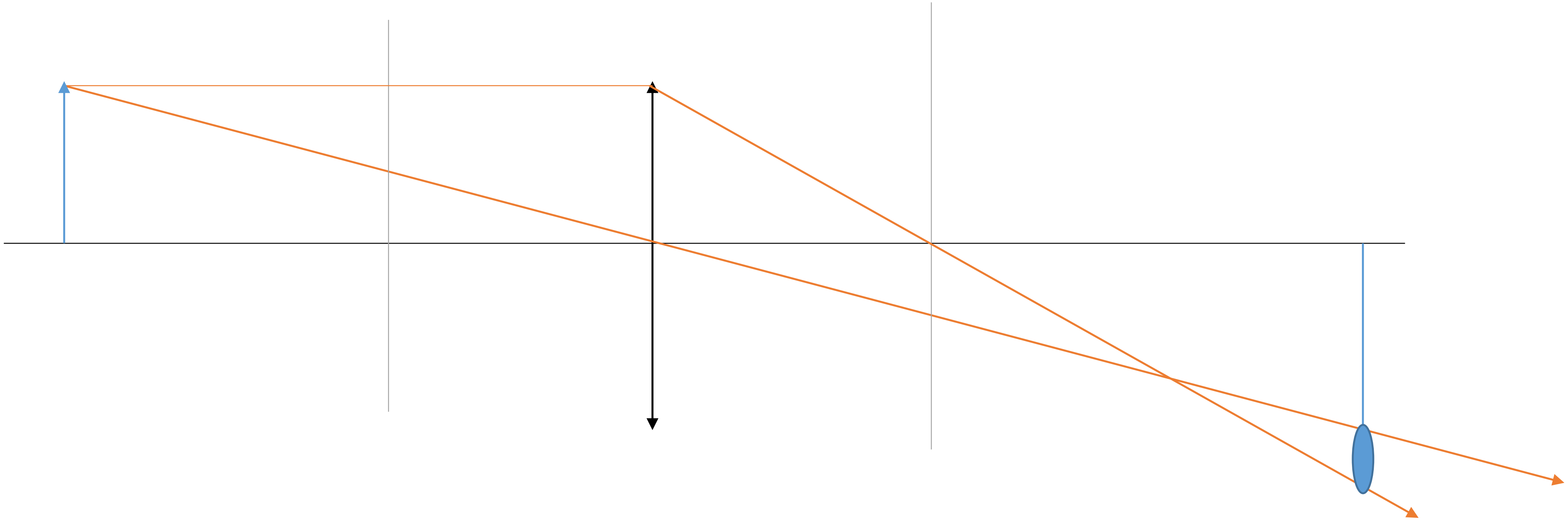
Camera models

- Depth of field



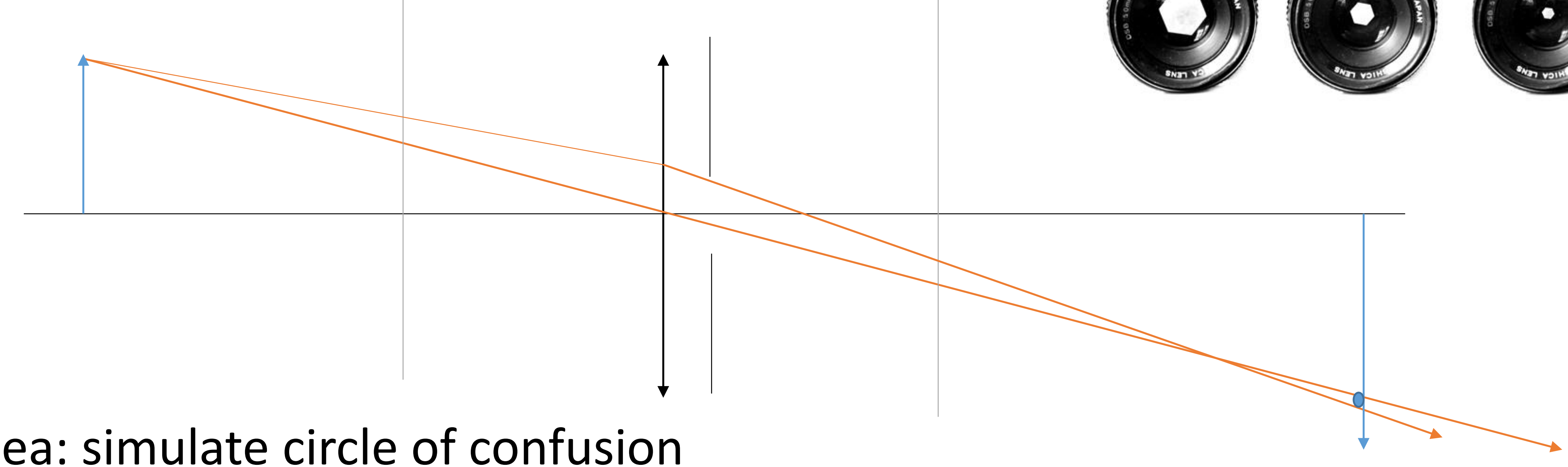
Camera models

- Depth of field



Camera models

- Depth of field



- Idea: simulate circle of confusion

Image-based rendering

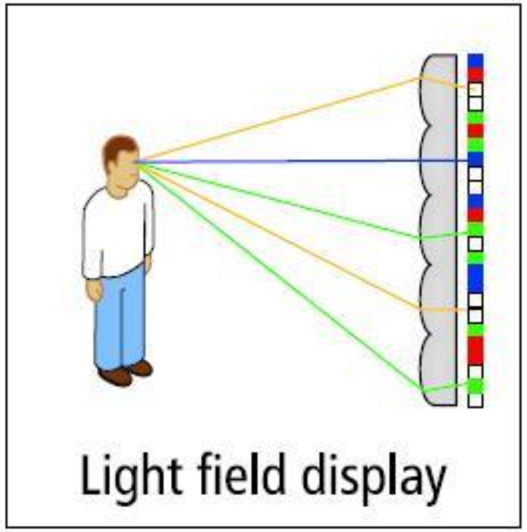
- Further from physically-based rendering
- Essentially interpolates between photographs
  - Lightfields / Lumigraph : dense array of photographs
  - Multi-view : sparse set
- Restricted to real-life scenes

Depth Synthesis and Local Warps for Plausible Image-based Navigation, Chaurasia et al.

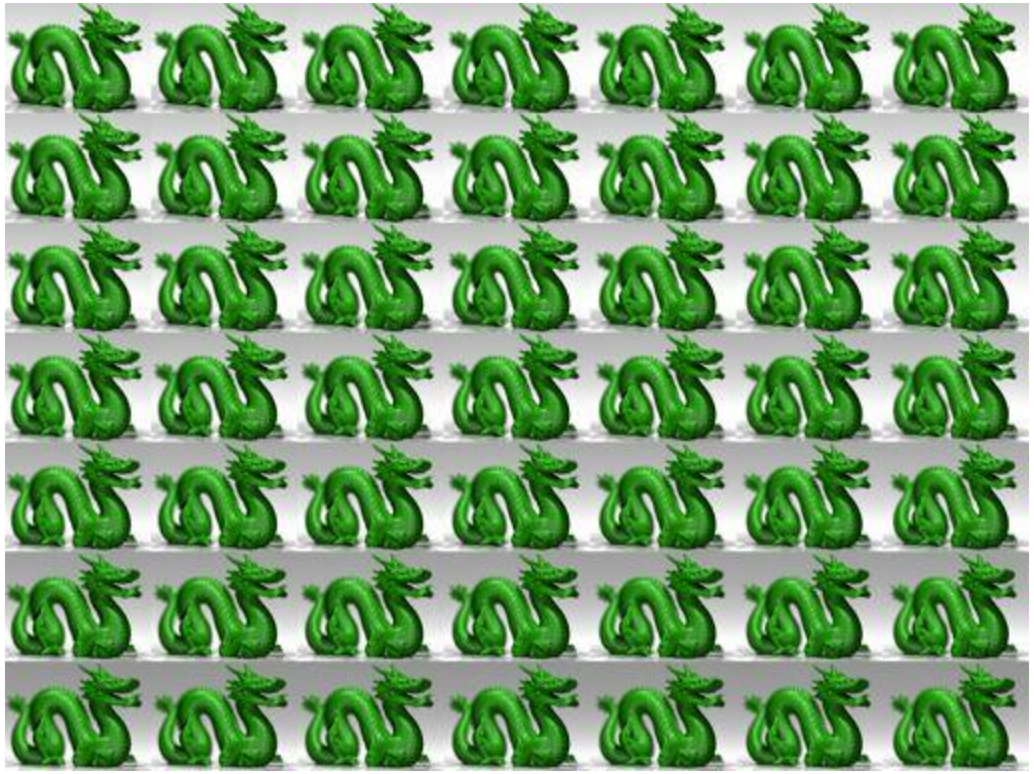
Light-fields

- The plenoptic function
  - $L(x, \omega)$
- Light-fields

- Display



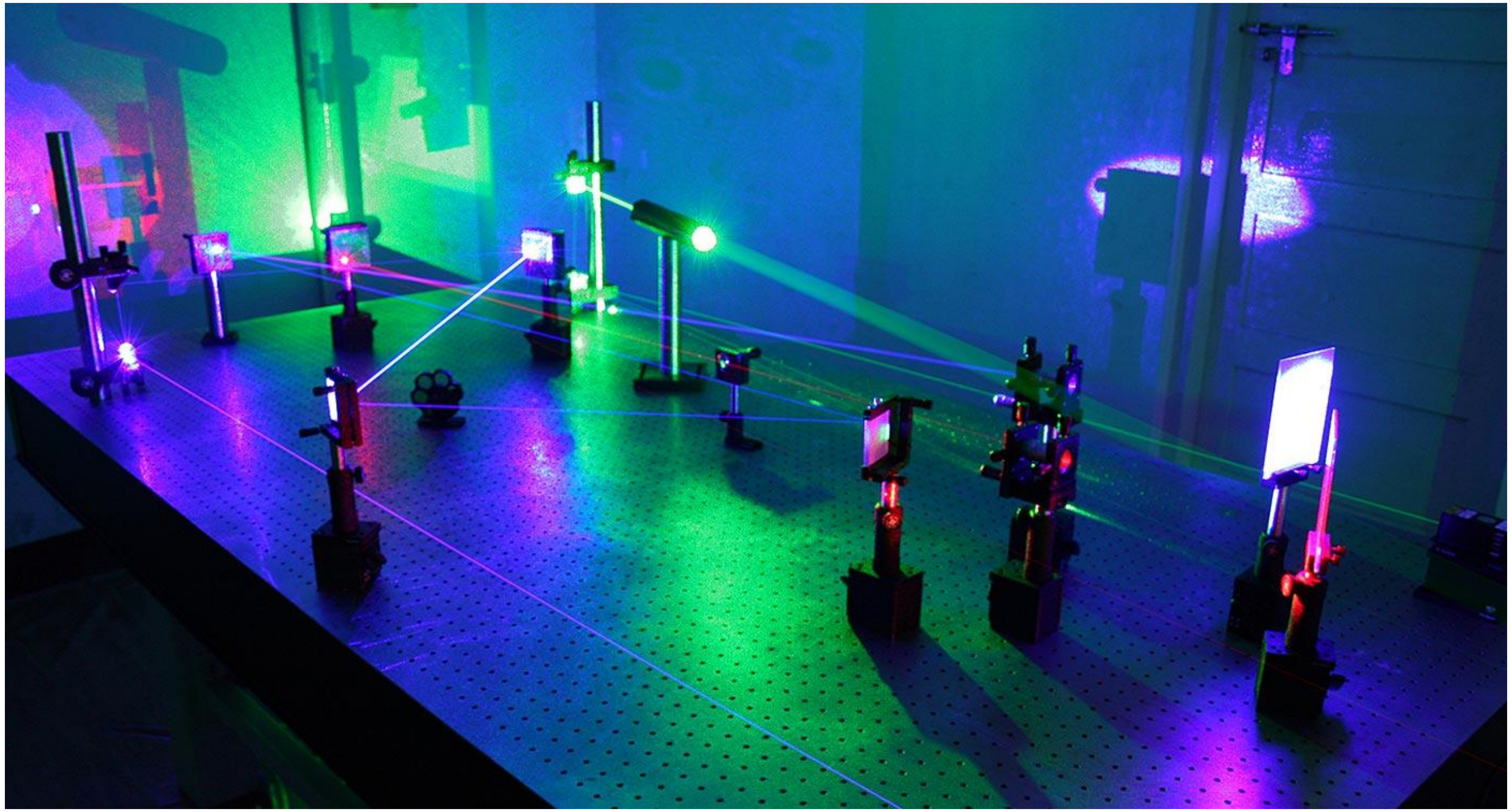
By Fuchs et al.



By Gordon Wetzstein

Holography & Computational holography

- If time permits…



Holography & Computational holography

Holography & Computational holography

Computational holography

- Simulates wavefront propagation from 3D scene
  - Tight time constraints – Gigapixels
  - Fourier optics

State-of-the-Art

- Fake or Photo ? https://area.autodesk.com/fakeorfoto/
- Graphics Turing Test