

Computer Graphics

- Distribution Ray Tracing -

Philipp Slusallek

Overview

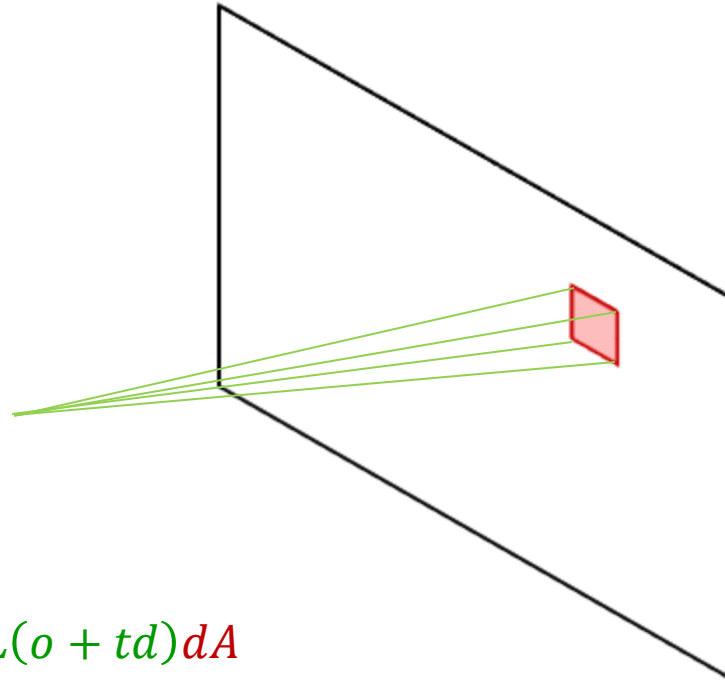
- **Other Optical Effects**
 - Not yet included in Whitted-style ray tracing
- **Stochastic Sampling**
- **Distribution Ray-Tracing**

Problems

- **Anti-aliasing**
- **Depth of field**
- **Motion blur**
- **BRDF**
- **Area Lights**

Anti-aliasing

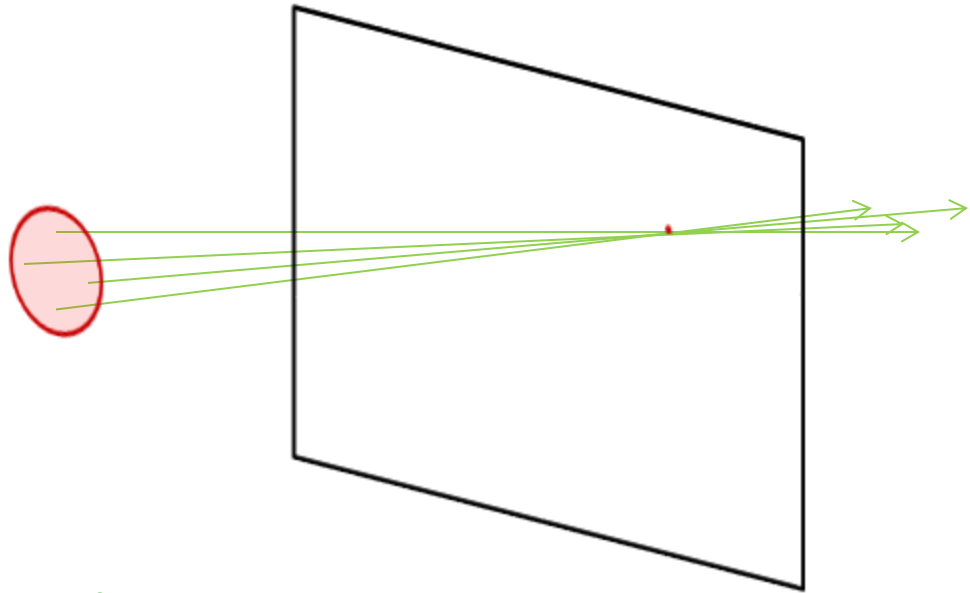
- Anti-aliasing
- Depth of field
- Motion blur
- BRDF
- Area Lights



$$I \approx \int_A L(o + td) dA$$

Depth of field

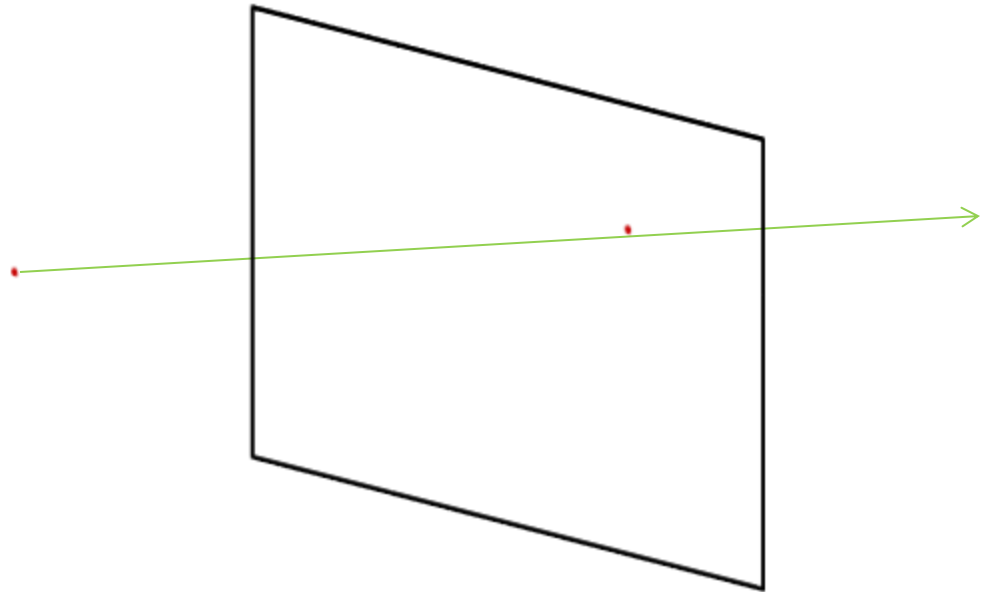
- Anti-aliasing
- Depth of field
- Motion blur
- BRDF
- Area Lights



$$I \approx \int_A L(o + td) dA$$

Motion blur

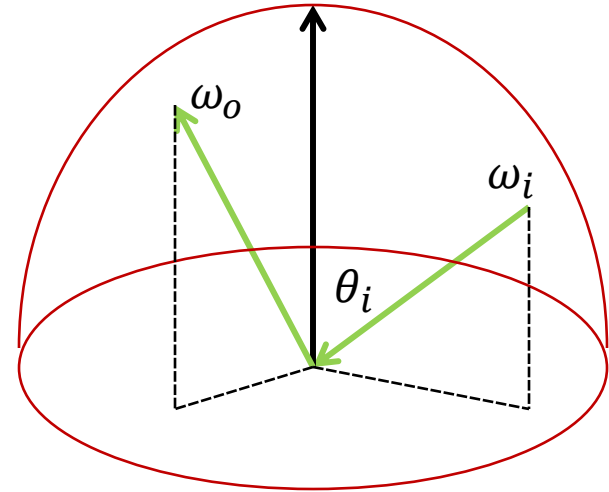
- Anti-aliasing
- Depth of field
- Motion blur
- BRDF
- Area Lights



$$L \approx \int_{[t_0, t_1]} L_T(o + td) dT$$

BRDF

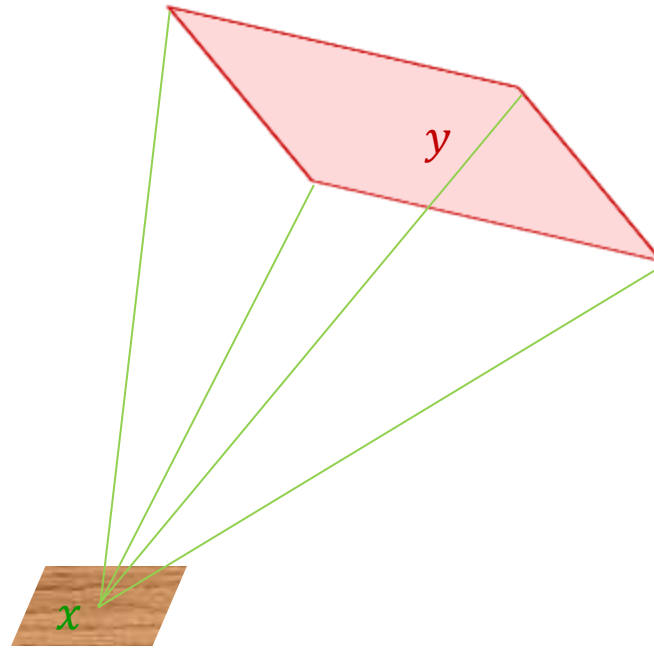
- Anti-aliasing
- Depth of field
- Motion blur
- BRDF
- Area Lights



$$L_o = L_e + \int_{\Omega_+} f_r L_i \cos \theta_i d\omega_i$$

Area Lights

- Anti-aliasing
- Depth of field
- Motion blur
- BRDF
- Area Lights



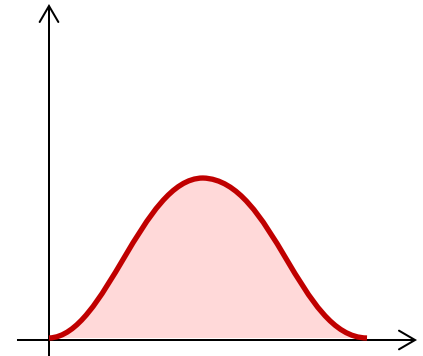
$$E_i = \int_A V(x, y) \frac{\cos \theta_A}{\|x - y\|^2} dA$$

Integration by MC-Sampling

- **Features**

- Anti-aliasing
- Depth of field
- Motion blur
- BRDF
- Area Lights

$$I = \int_D f(x) dx$$



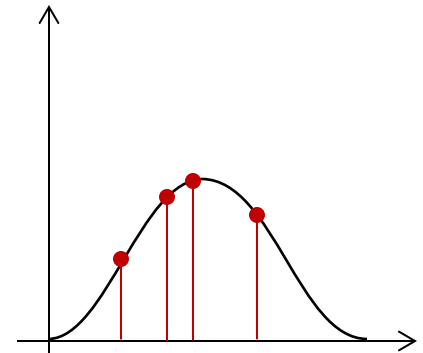
- **Monte-Carlo (MC) Integration**

- Stochastic sampling of domain
- Averaging of results, weighted by probability
- Careful choice of samples essential for good results



$$I \approx \frac{D}{n} \sum_{i=1}^n \frac{f(x_i)}{p(x_i)}$$

x_i sampled $\propto p(x)$



STOCHASTIC SAMPLING

(VERY SHORT INTRO)

Random Number

- **Random Number**
 - Uniformly distributed
 - ξ in $[0, 1)$

- **Pseudo-Random Number**
 - Linear congruential
 - Mersenne-Twister
 - ...
 - Speed / evenness trade-off

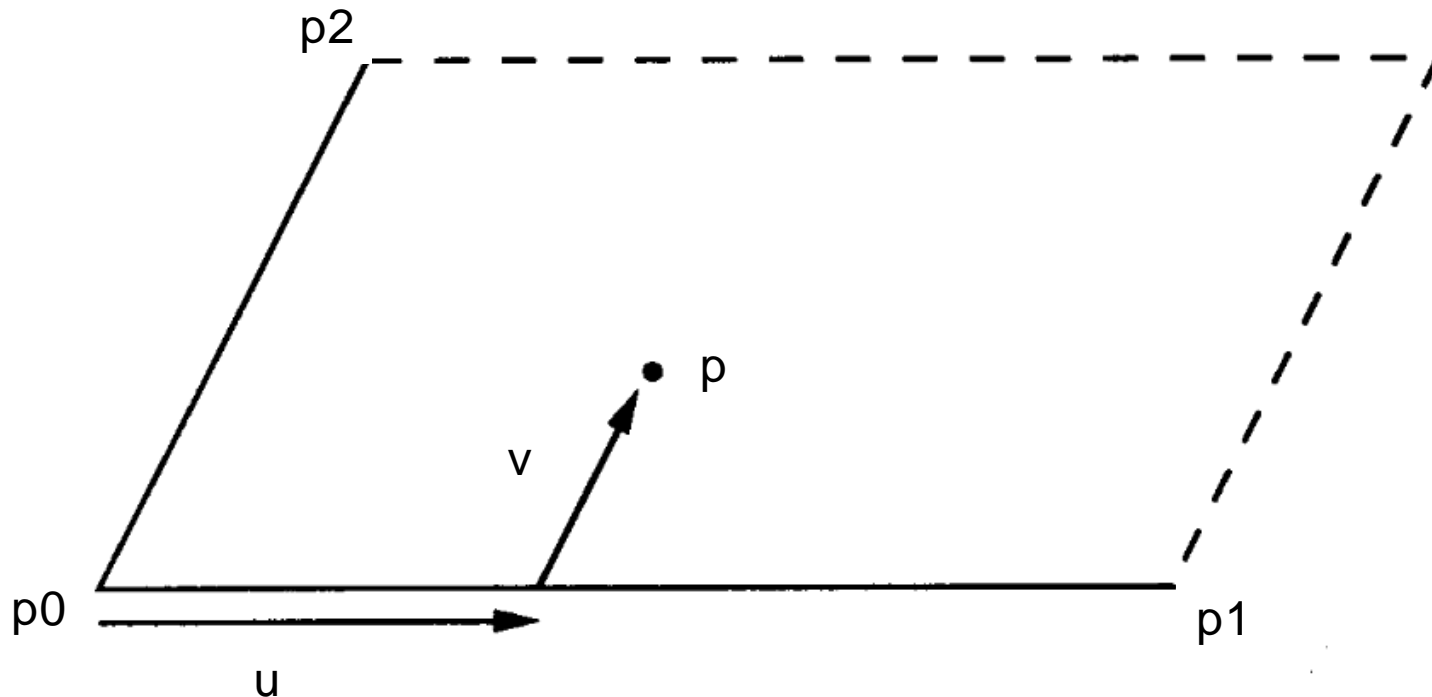
Parallelogram Sampling

- **Parametric Form**

- $p(u, v) = p_0 + u(p_1 - p_0) + v(p_2 - p_0) =$
 $(1 - u - v)p_0 + up_1 + vp_2$

- **Random Sampling**

- $p(\xi_1, \xi_2)$



Triangle Sampling

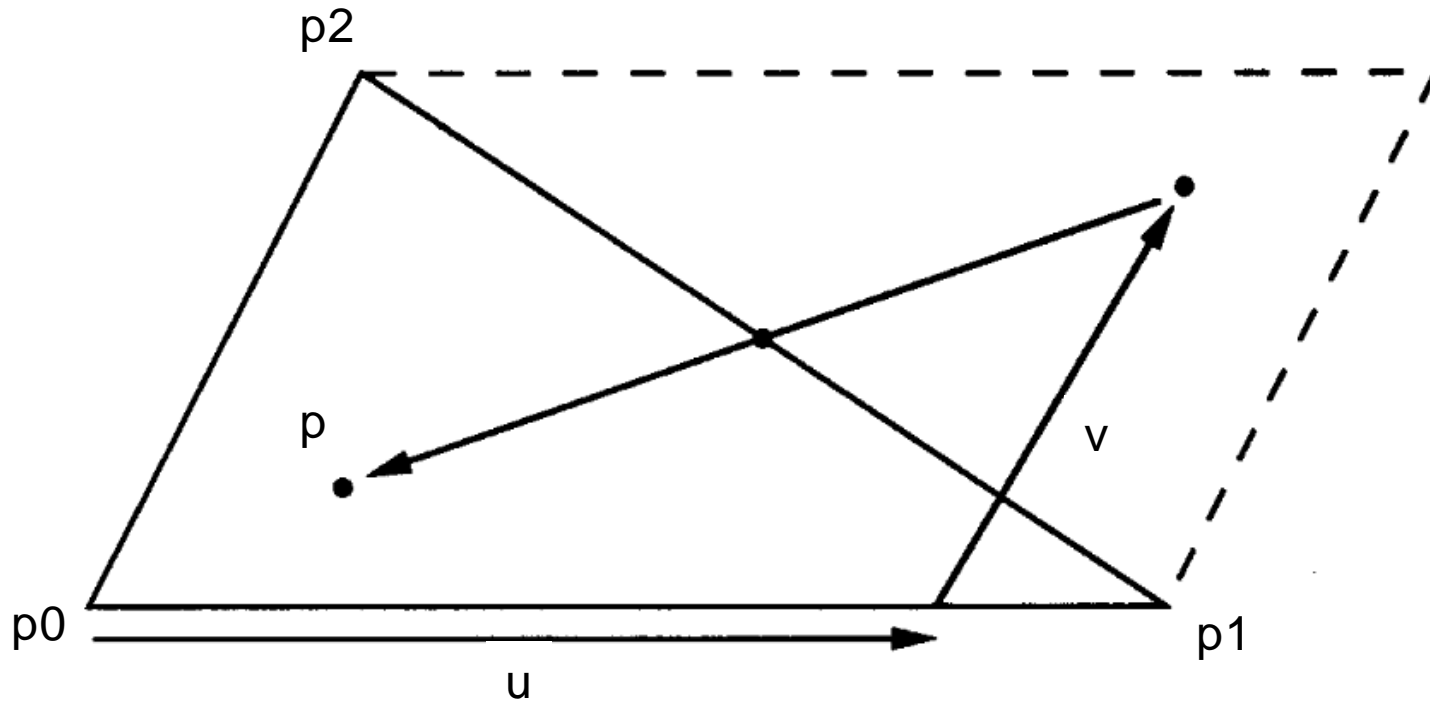
- **Parametric Form**

- $p(u, v) = (1 - u - v)p_0 + up_1 + v p_2$

- **Random Sampling**

- if $\xi_1 + \xi_2 < 1 : p(\xi_1, \xi_2)$

- if $\xi_1 + \xi_2 > 1 : p(1 - \xi_1, 1 - \xi_2)$



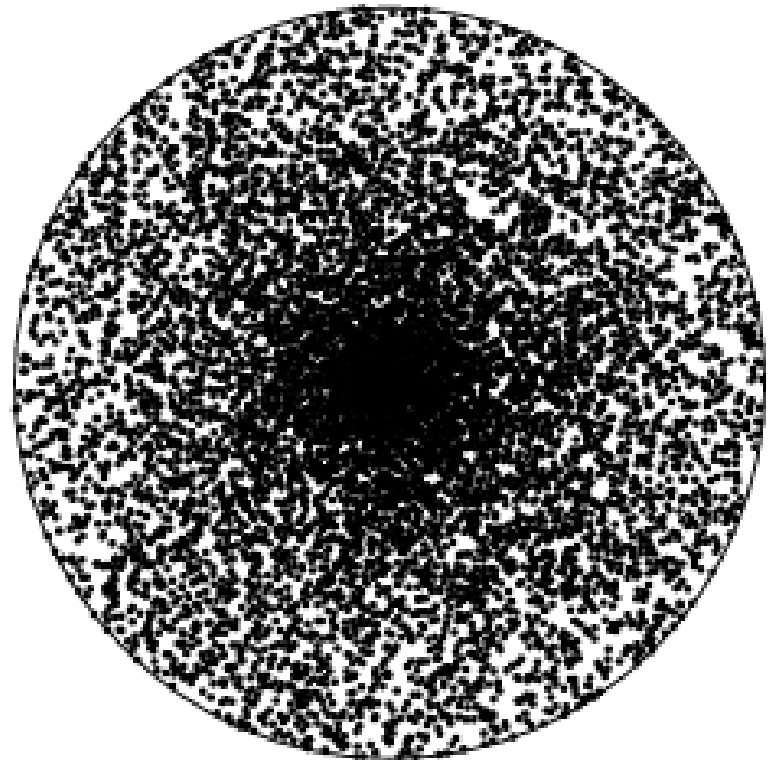
Disc Sampling

- **Parametric Form**

- $p(u, v) = \text{Polar2Cartesian}(R v, 2 \pi u)$ // disc radius R

- **Naïve Sampling (wrong!)**

- $p(\xi_1, \xi_2)$



Disc Sampling

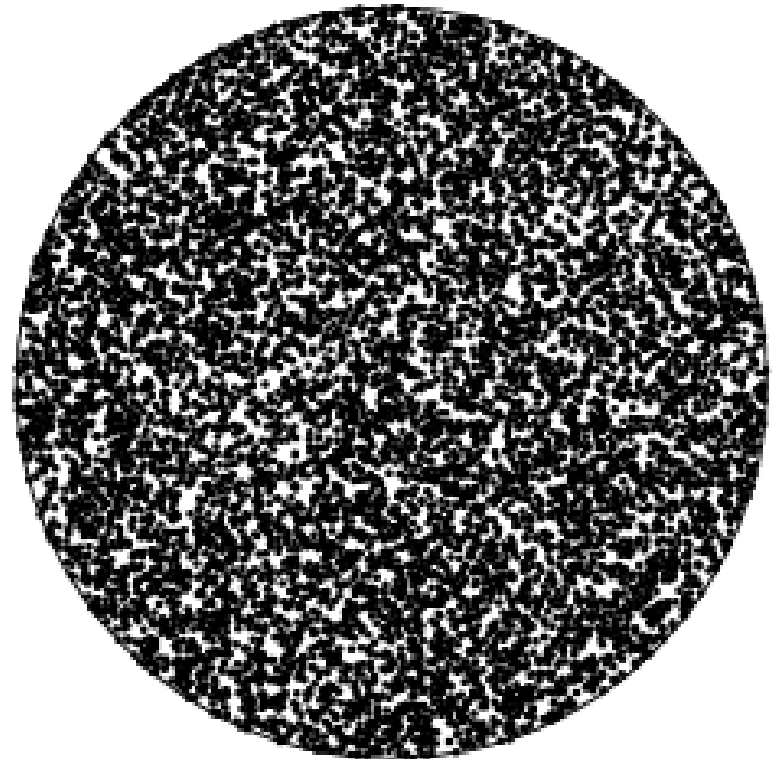
- **Parametric Form**

- $p(u, v) = \text{Polar2Cartesian}(R v, 2 \pi u)$ // disc radius R

- **Correct Sampling**

- $p(\xi_1, \sqrt{\xi_2})$

- Results in uniform sampling over area



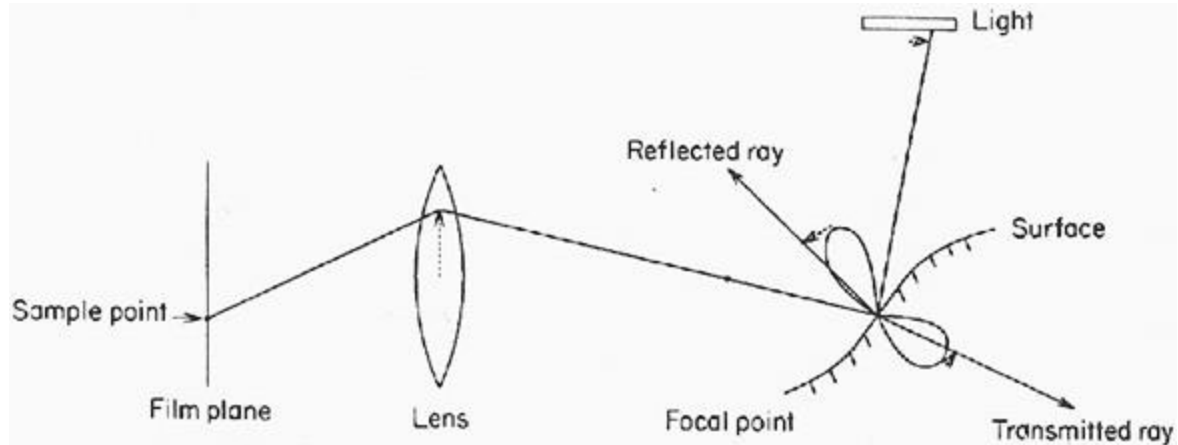
- **For other cases, see Phil Dutre's Global Illumination Compendium at**

<http://people.cs.kuleuven.be/~philip.dutre/GI/TotalCompendium.pdf>

DISTRIBUTION RAY-TRACING

Distribution Ray Tracing

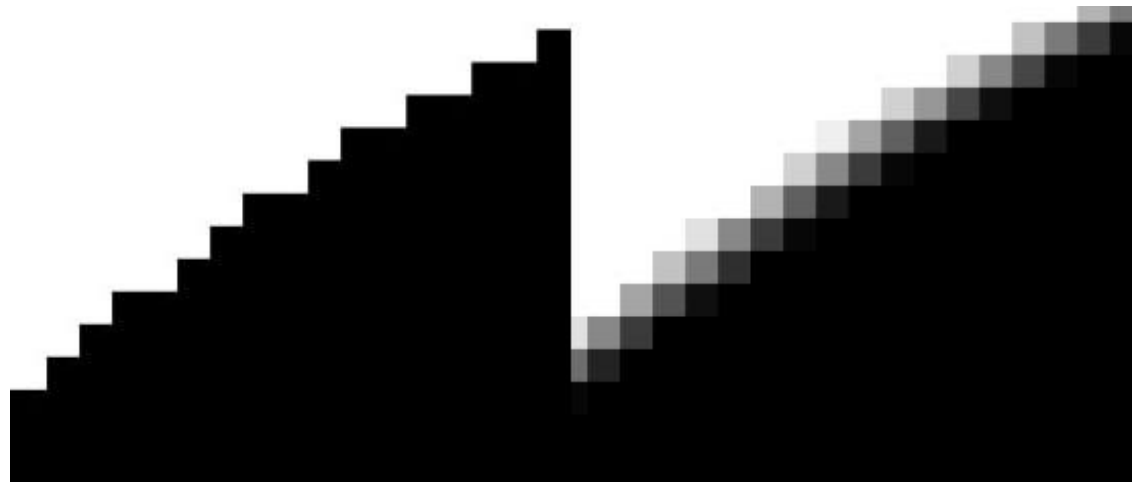
- Apply random sampling for many aspects in RT
 - Pixel
 - Anti-aliasing
 - Lens
 - Depth of field
 - Time
 - Motion blur
 - BRDF
 - Glossy reflections & refractions
 - Area Lights
 - Soft shadows
 - Based on paper:
R. Cook et al.,
Distributed Ray Tracing,
Siggraph'84



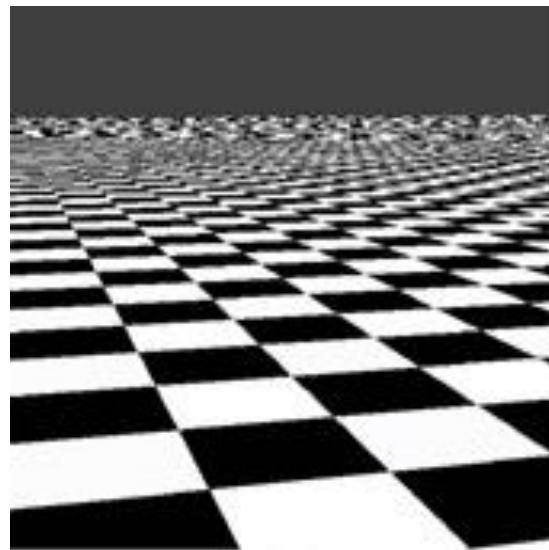
Anti-Aliasing

- **Artifacts**

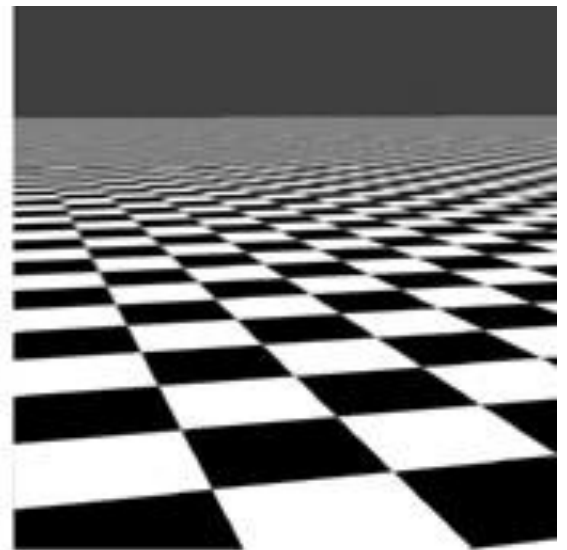
- Jagged edges



- Aliased patterns



(a)

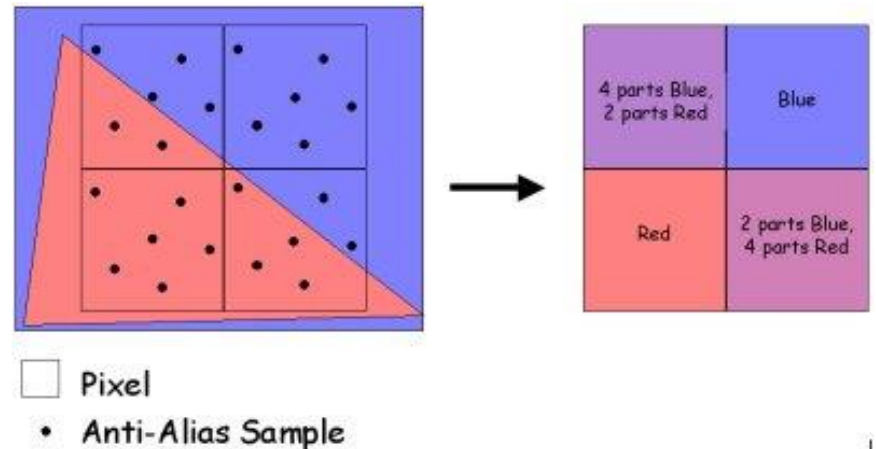


(b)

Anti-Aliasing

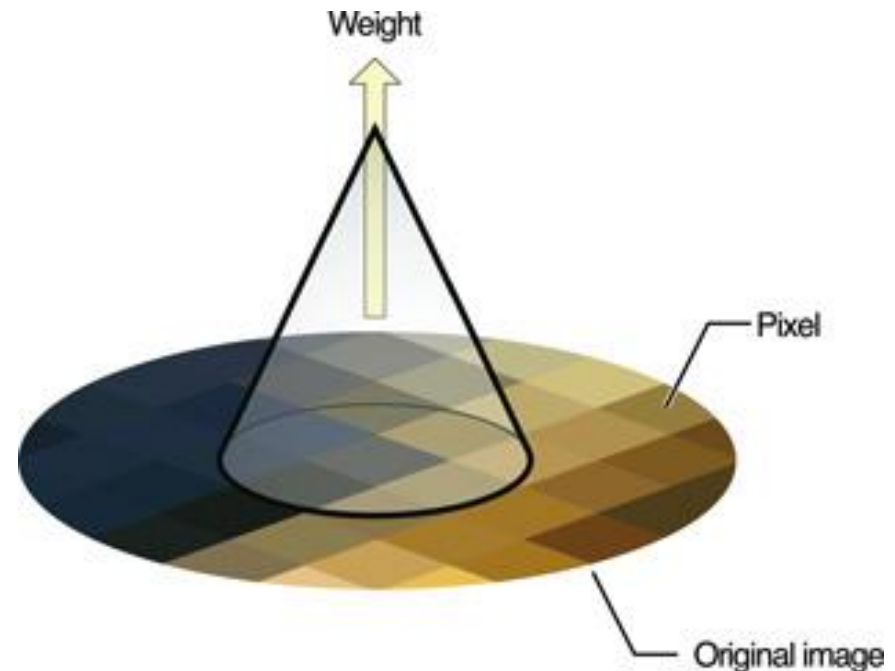
- **Basic Method**

- Plain average
- Box filter $f(x, y) = 1$
- $L = \frac{\sum_{i=1}^n L(\xi_{i1}, \xi_{i2})}{n}$



- **Filtering**

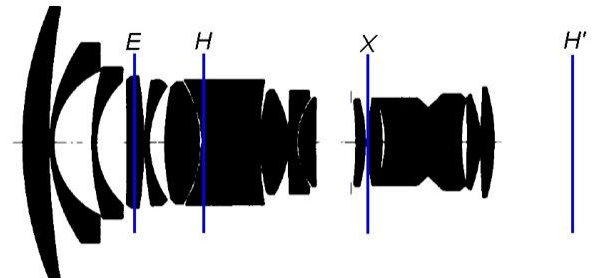
- Weighted average
- Filter $f(x, y)$
- $L = \frac{\sum_{i=1}^n f(\xi_{i1}, \xi_{i2})L(\xi_{i1}, \xi_{i2})}{\sum_{i=1}^n f(\xi_{i1}, \xi_{i2})}$



Depth of Field

- **Real Camera**

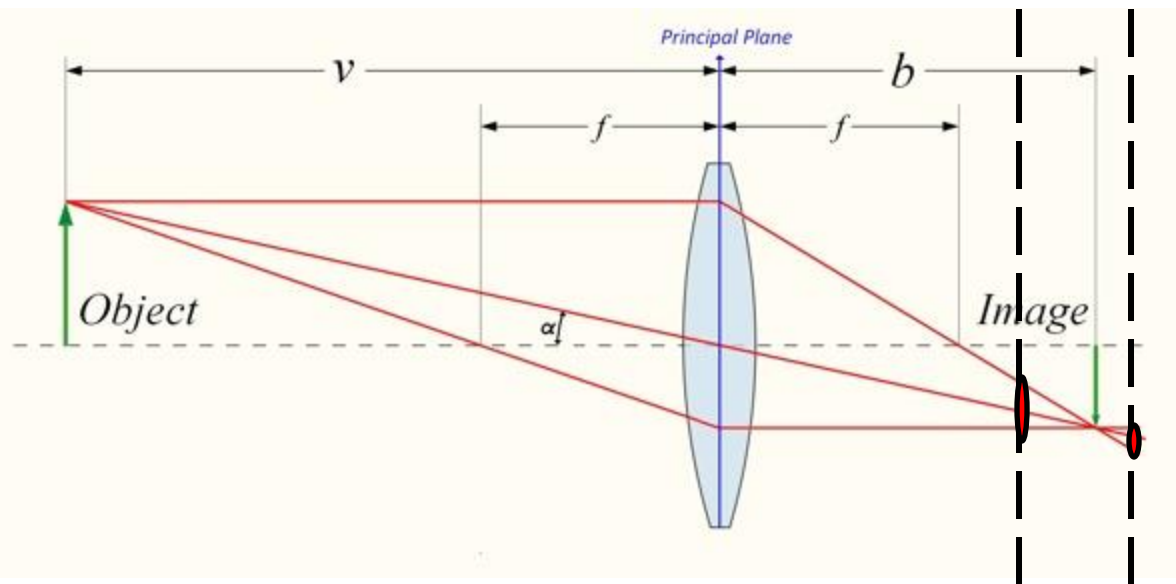
- Complex lenses that focus one distance onto the image
 - Finite aperture size
- Blurred features except for focal plane



Depth of Field

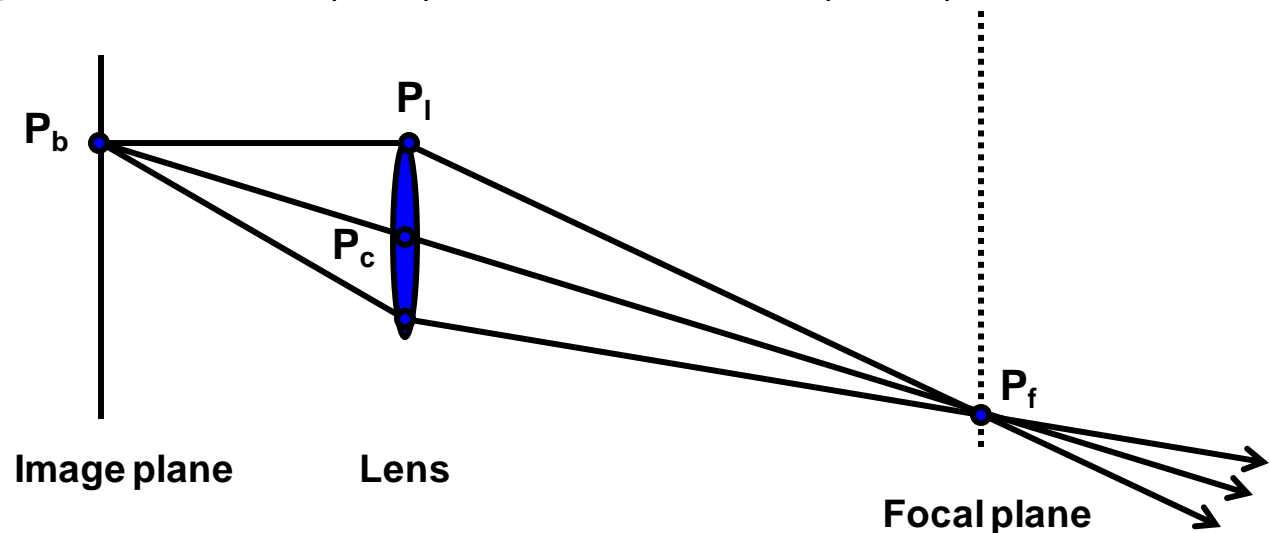
- **Thin Lens**

- Focus light rays from point on object onto image plane
 - Sharp features at focal plane
 - Blurred features before/beyond focal plane
- Depth of field: depth range with acceptably small *circle of confusion*
 - Smaller than one pixel

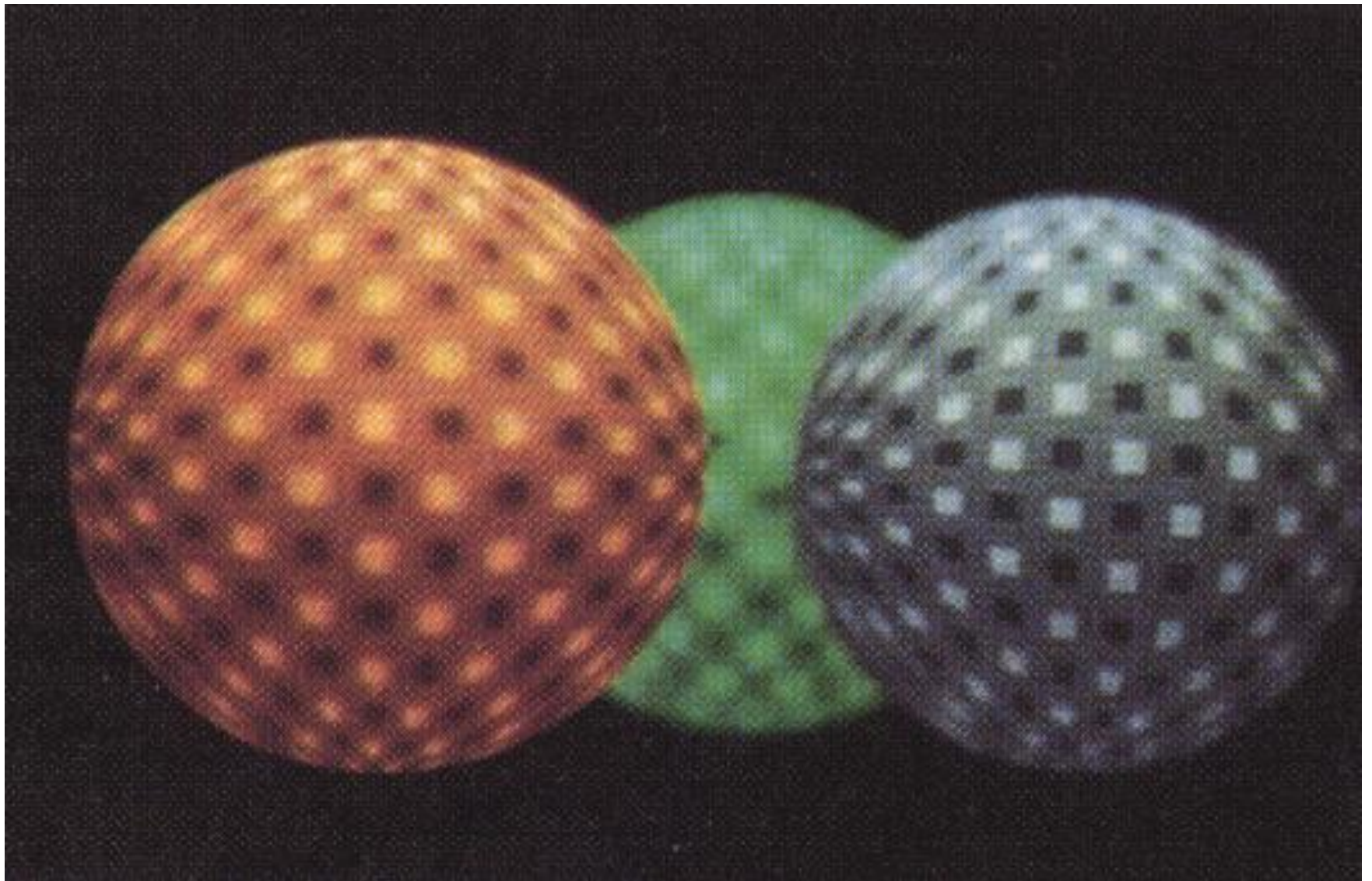


Depth of Field

- **Compute ray through lens center**
 - Compute focus point P_f on focal plane, determined by P_b and P_c
- **Compute new ray origin**
 - Sample coordinates (x, y) of aperture diameter ($= f / N$)
 - Compute P_l : $\text{ray.origin} += P_c + x * \text{camera.right} + y * \text{camera.up}$
 - Might include modeling the shape of the aperture
- **Compute new ray direction**
 - Compute $\text{ray.direction} = P_f - P_l \rightarrow$ vector from P_l to P_f
 - Normalize

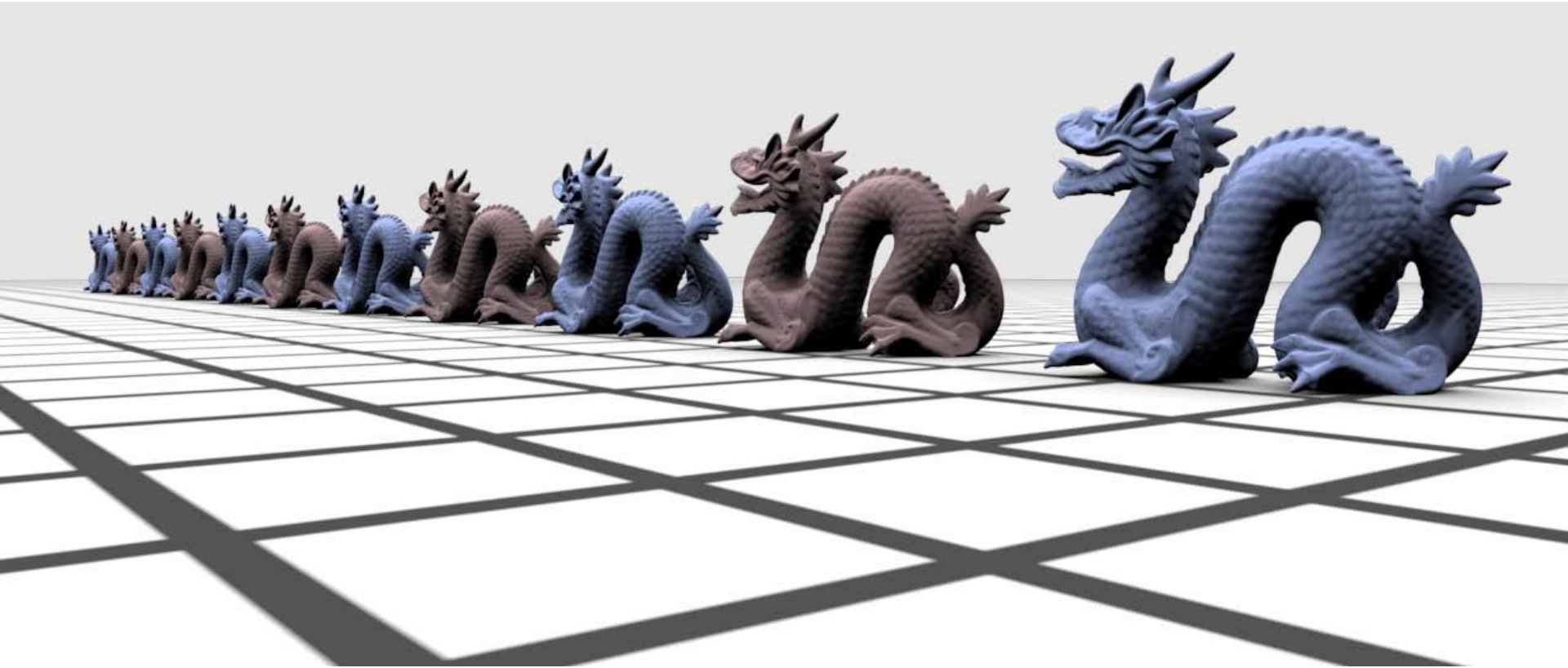


Depth of Field



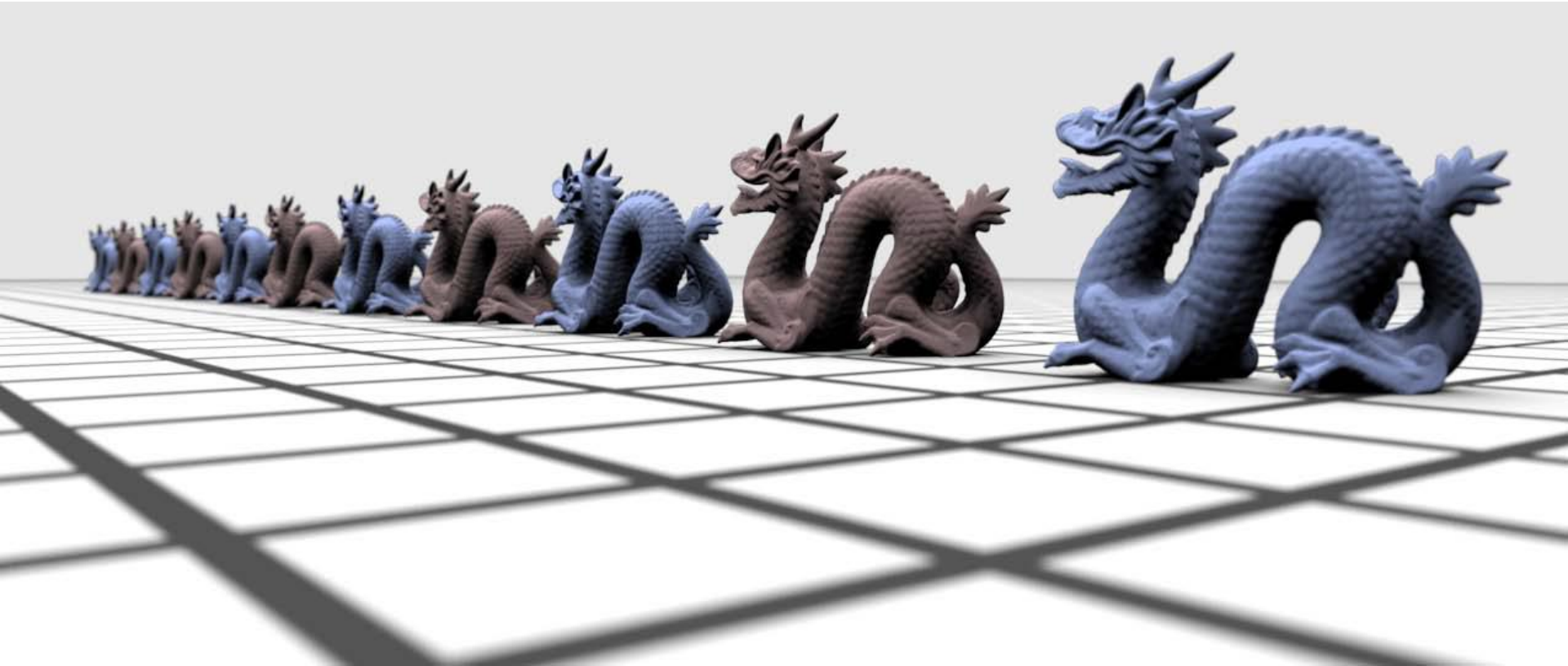
Depth of Field

- Zero Aperture



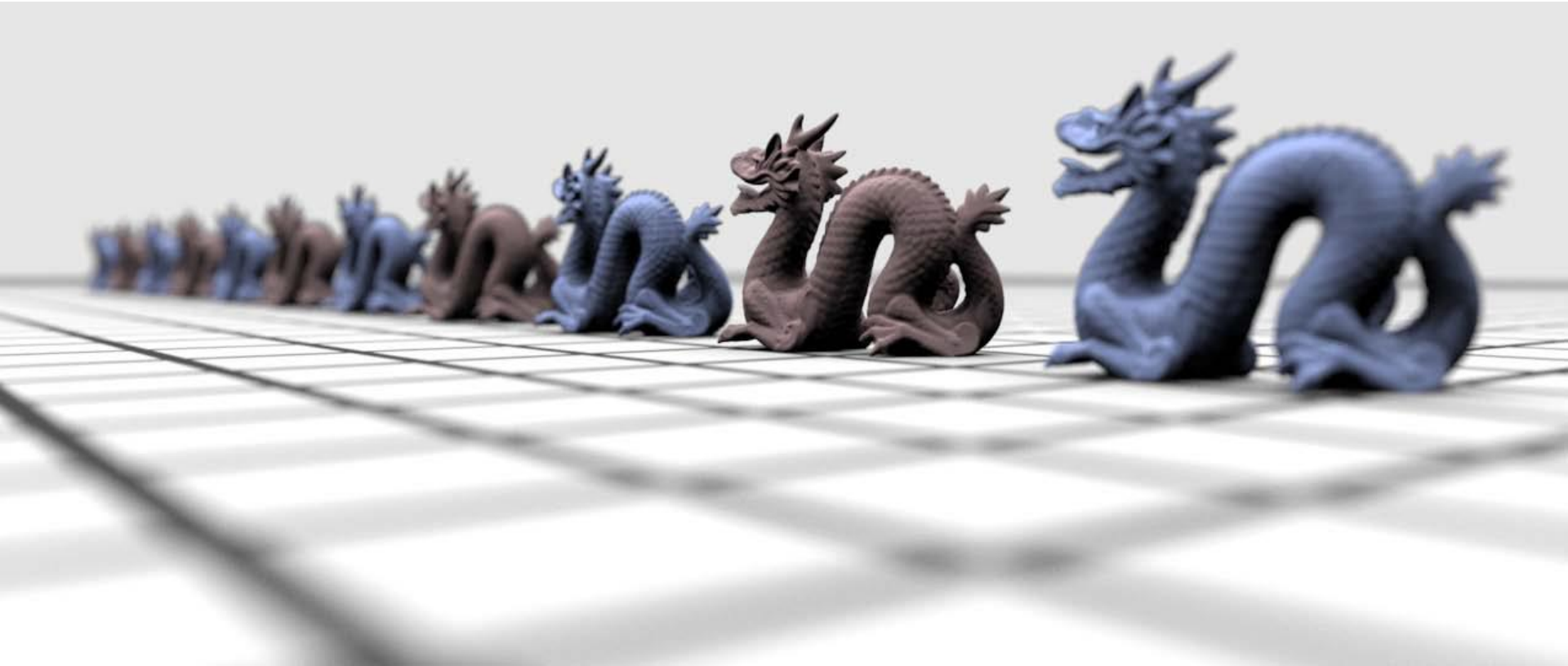
Depth of Field

- **Small Aperture**



Depth of Field

- Large Aperture



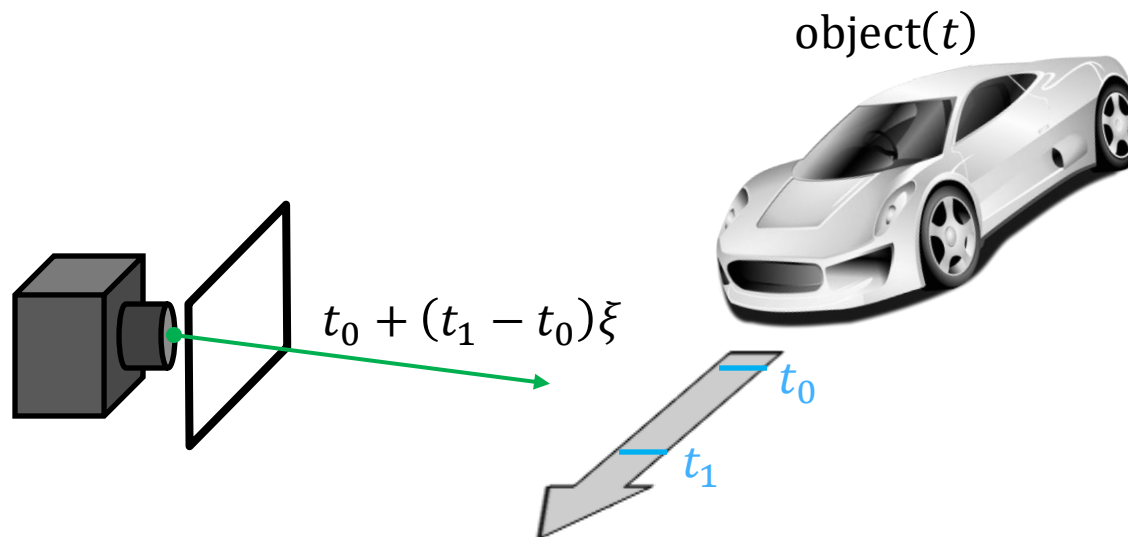
Depth of Field

- **Very Large Aperture**



Motion Blur

- **Real Camera**
 - Finite exposure time
 - Shutter opening at t_0
 - Shutter closing at t_1



Motion Blur

- **Real Camera**

- Finite exposure time
- Shutter opening at t_0
- Shutter closing at t_1

- **Approach**

- Sample time t in $[t_0, t_1]$: $t = t_0 + \xi (t_1 - t_0)$
- Assign time t to new camera ray/path
- Models with moving camera and/or moving objects in the scene
 - Time-dependent transformations
 - Transform objects *or inverse-transform ray* to proper positions at t
- Assume instantaneous opening and closing
 - Can be generalized by modeling shape of aperture over time

- **Gotchas**

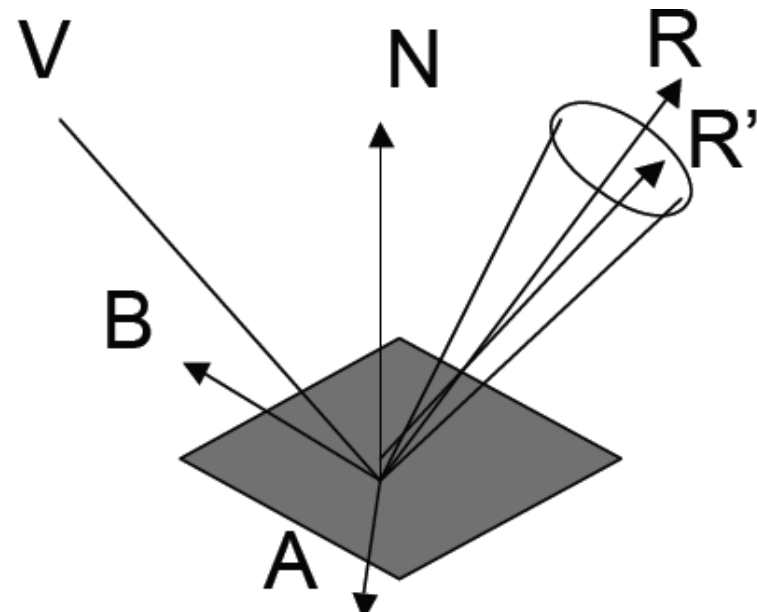
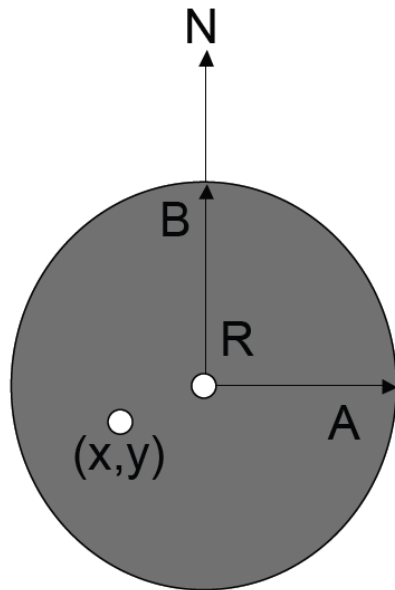
- Acceleration structures built over dynamic objects

Motion Blur



Fuzzy Reflections/Refractions

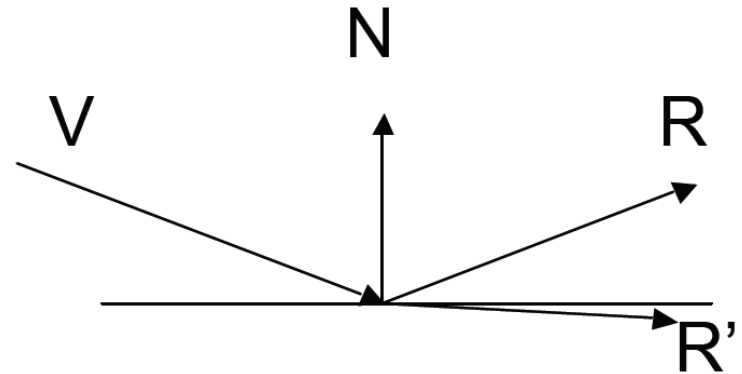
- **Real Materials**
 - Never perfectly smooth surfaces
- **Empirical Approach**
 - Compute orthonormal frame around reflected/refracted direction
 - Sample coordinates (x, y) on disc: $\text{ray.direction} += x * a + y * b$
- **Or better use \cos^n sampling (\rightarrow GI Compendium)**



Fuzzy Reflections/Refractions

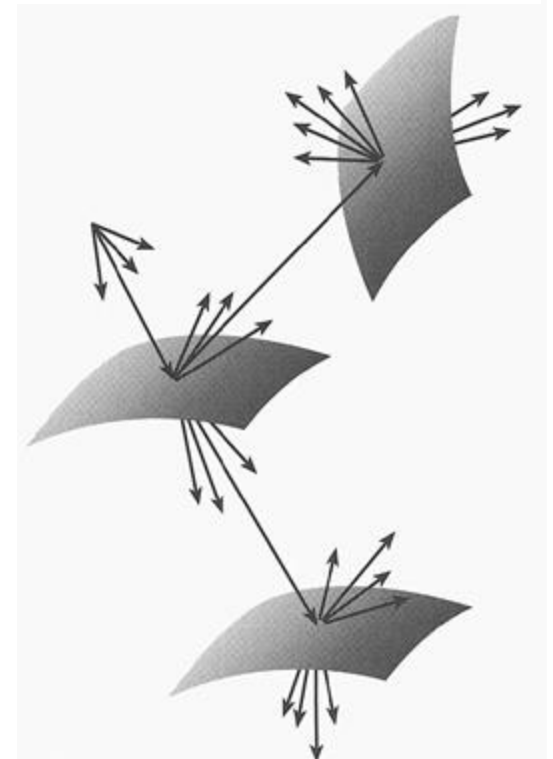
- **Gotchas**

- Perturbed ray may go inside
- Check sign of dot product with N
- Ignore rays on wrong side

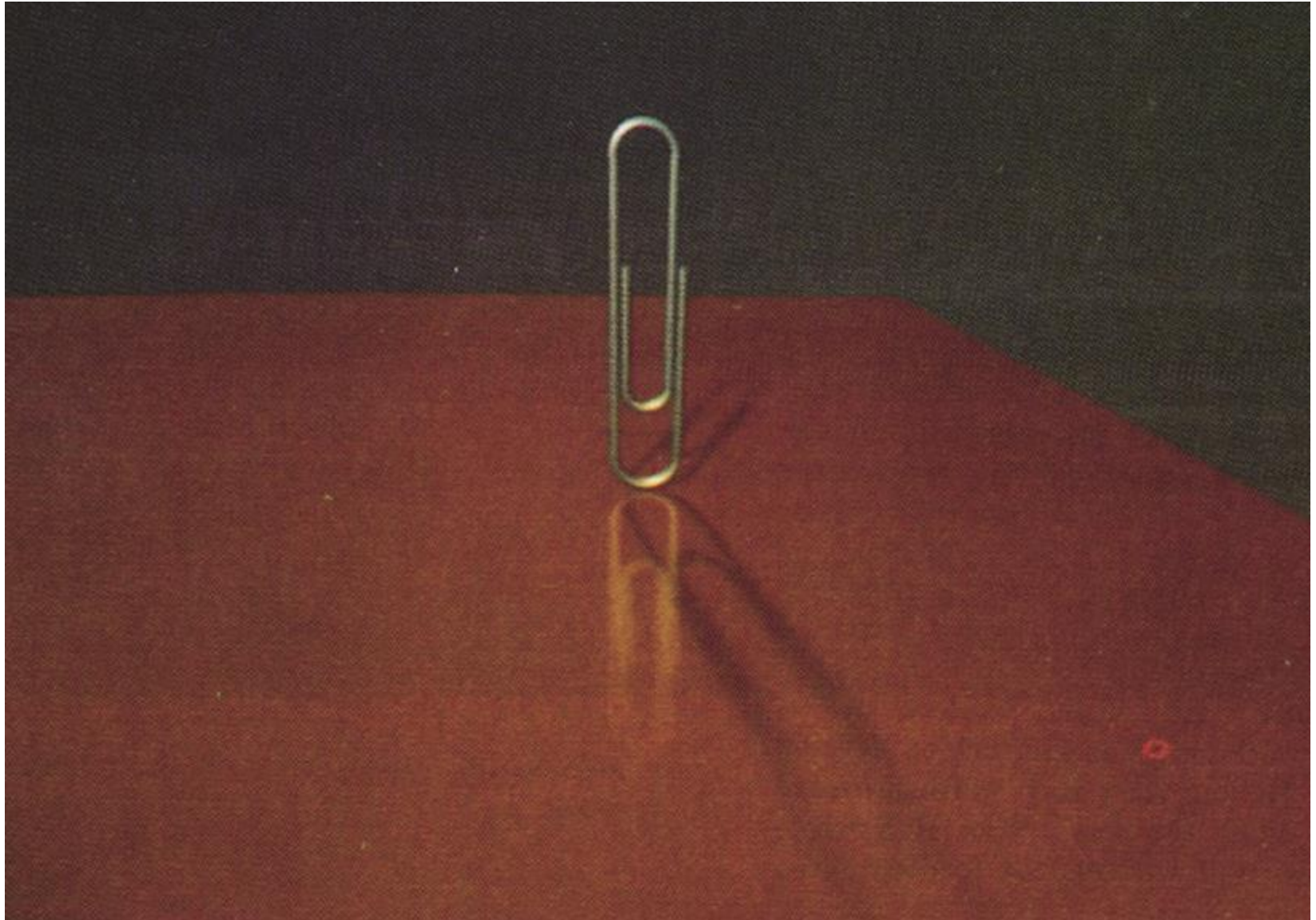


- **Inter-Reflections/Refractions**

- Recursively repeat process
 - At surfaces with corresponding materials

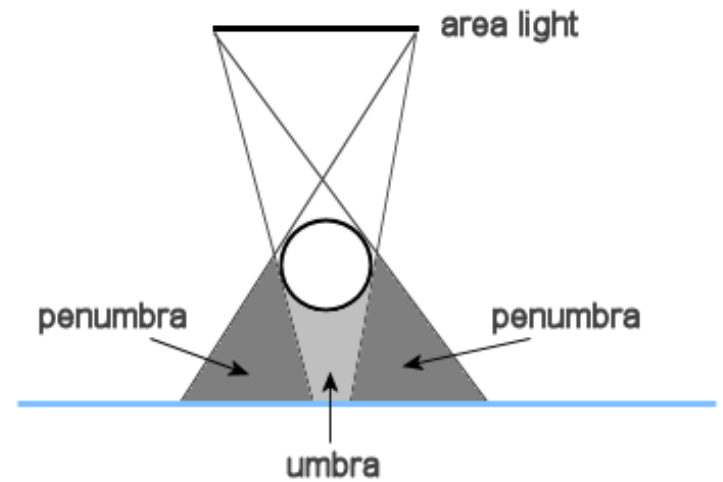
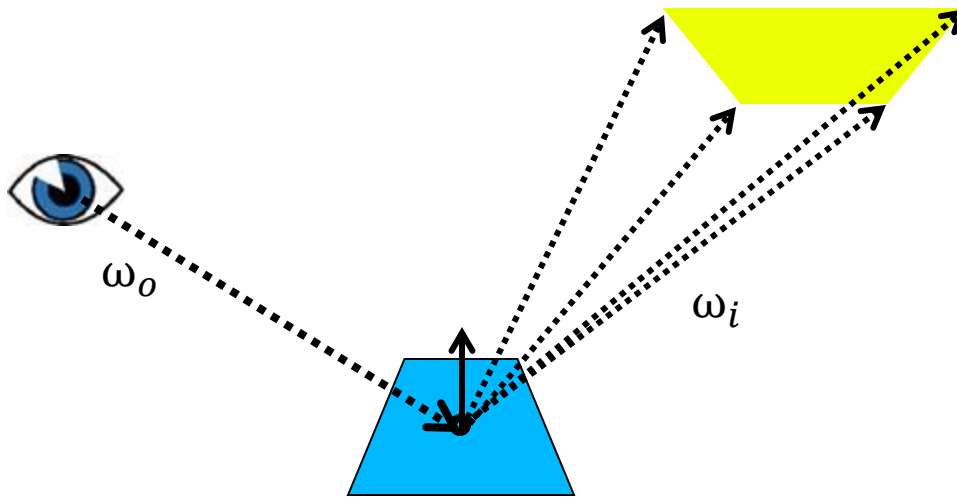


Fuzzy Reflections/Refractions



Soft Shadows

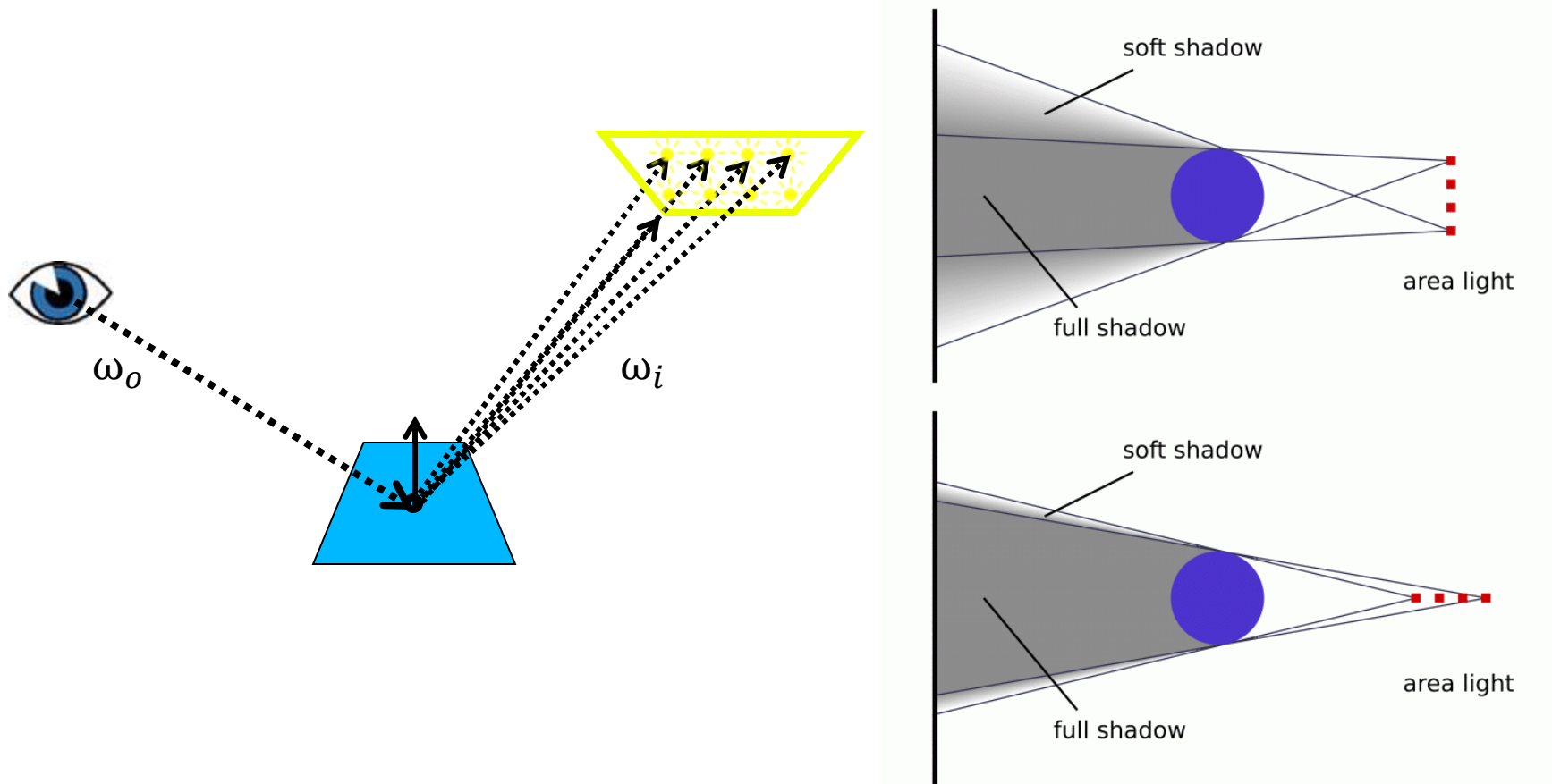
- **Real Light Sources**
 - Finite area



Soft Shadows

- **Approach**

- Random sample point on surface of light source
- Scale intensity by area and cosine



Soft Shadows

- **Small vs. Large Area Light**



Distribution Ray-Tracing

- **It can do all of these things:**
 - Anti-aliasing
 - Depth of field
 - Motion blur
 - BRDF
 - Area Lights
- **It's great, right?**
- **Or is it?**

Combined Effects

- **High-Dimensional Sampling Space**

- number of anti-aliasing samples
- x number of lens samples
- x number of time samples
- x number of material samples
- x number of light samples

➔ Exponential growth of number of rays w/ depth:

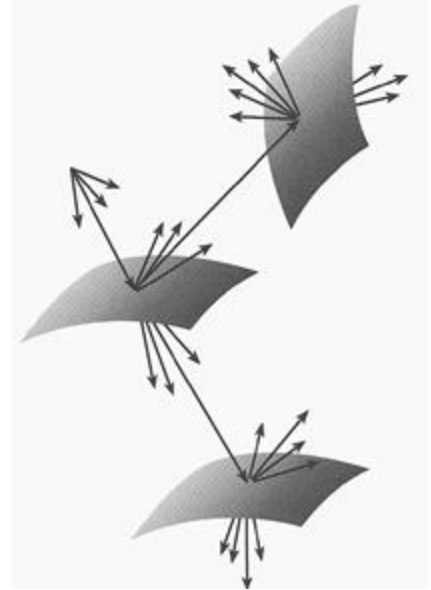
- Reflection < 1 at each bounce → exponentially less contribution with each additional bounce

→ Exponentially increasing number of higher-order rays, with exponentially decreasing effect on final pixels

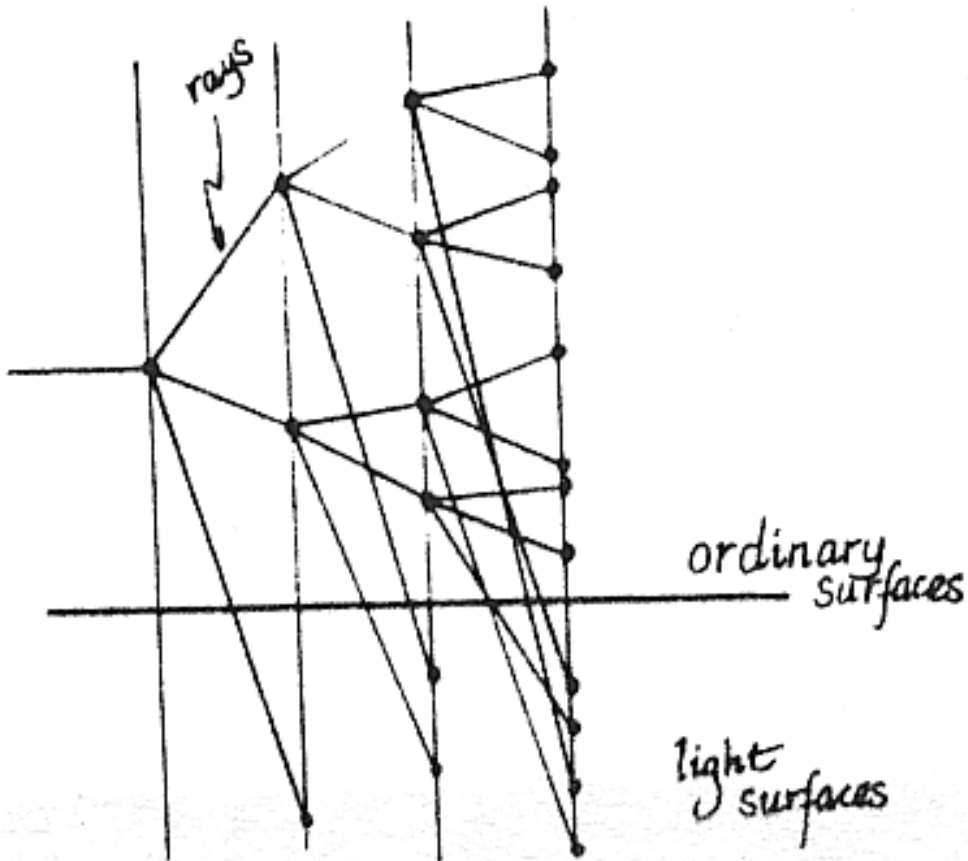
- This approach is (exponentially bad)²

- **Solution: Path-Based Approach**

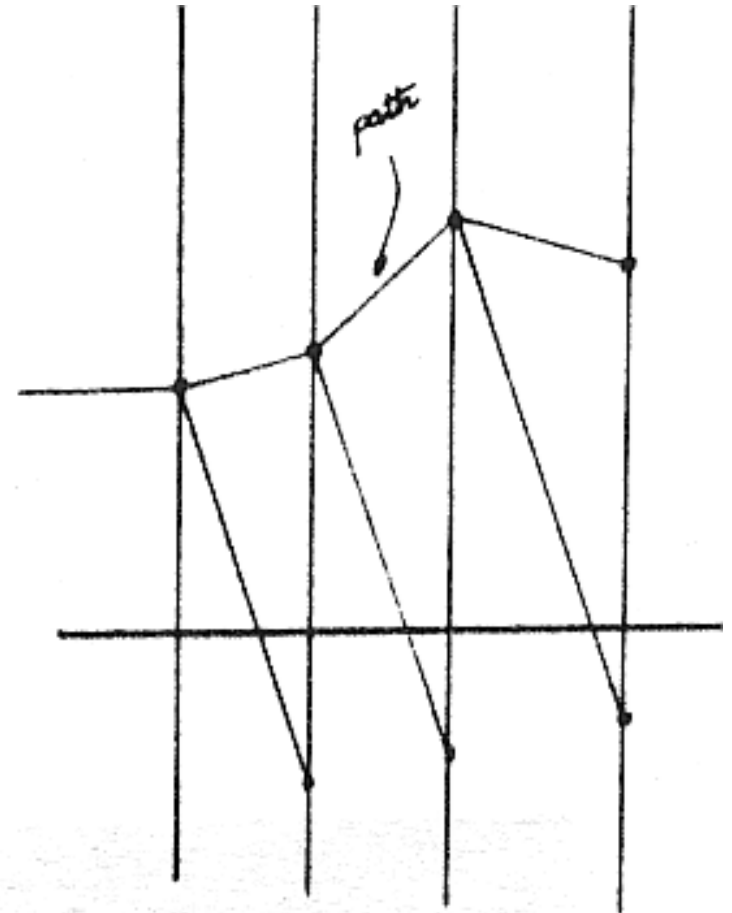
- Avoid exponential growth in ray tree
- Pick a single sample at each step: → Create a sample *path*
- Average results over several paths per pixel → *path tracing* (RIS)
 - Theoretical underpinning: Monte-Carlo Integration



Comparison to Path Tracing



Distribution Ray Tracing

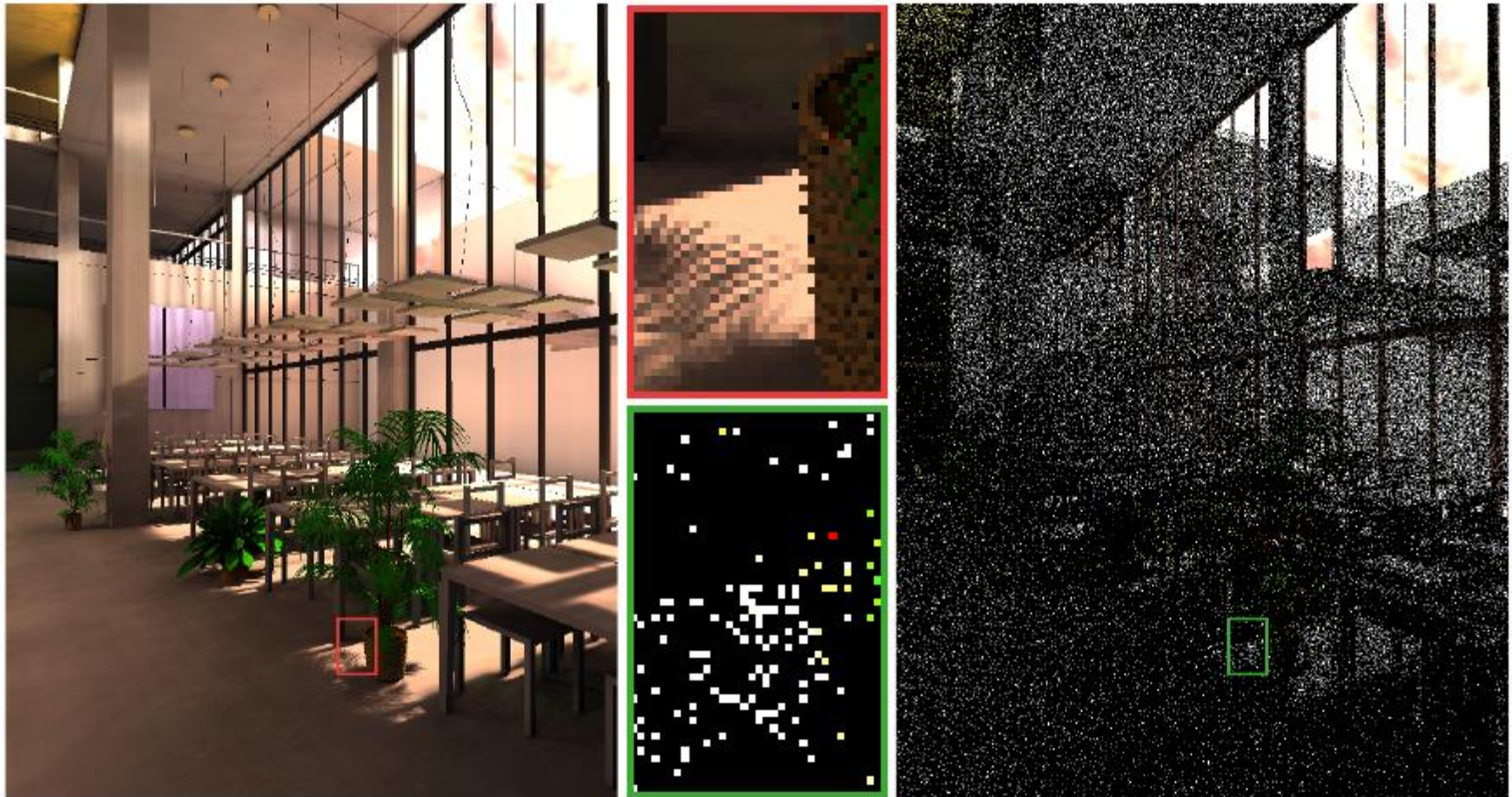


Path Tracing

(figure by Kajiya)

Recent Advances in Lighting Sim.

- **Importance Caching for Complex Illumination**
 - By Iliyan Georgiev et al., Eurographics 2012



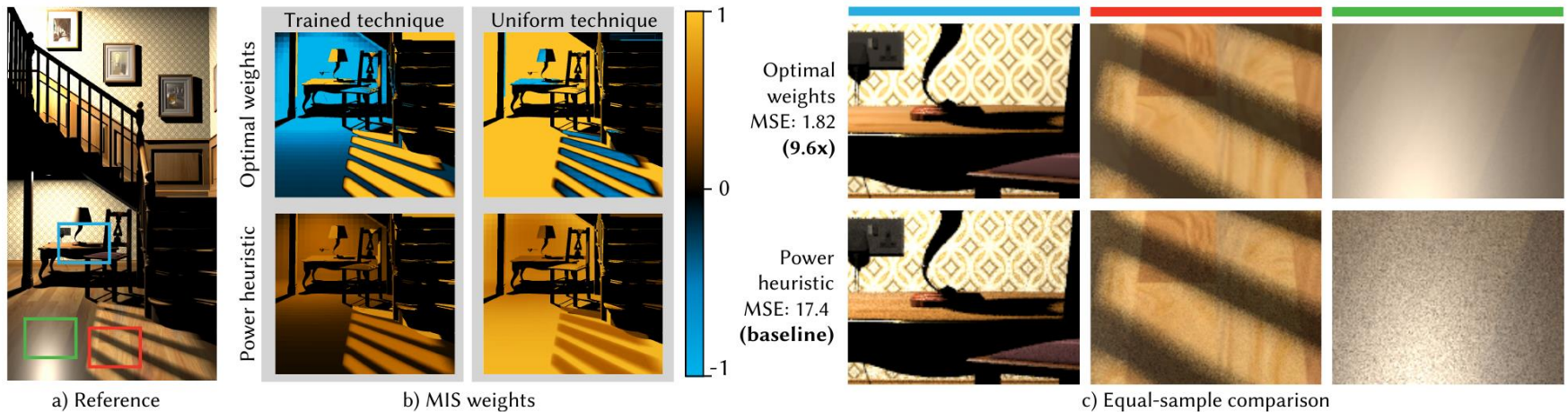
Recent Advances in Lighting Sim.

- **Light Transport Simulation with Vertex Connection and Merging (VCM)**
 - By Iliyan Georgiev et al., Siggraph 2012



Recent Advances in Lighting Sim.

- **Optimal Multiple Importance Sampling**
 - By Pascal Grittmann, Jaroslav Krivanek, et al., Siggraph 2019



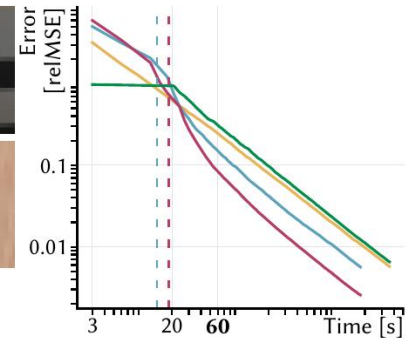
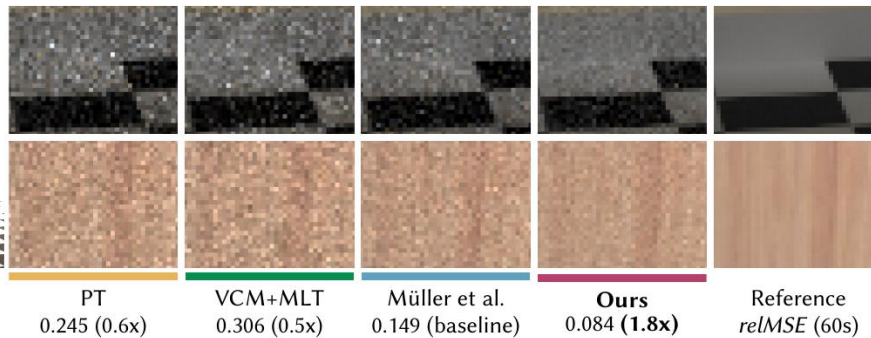
Recent Advances in Lighting Sim.

- **Variance-Aware Path Guiding**

- By Alexander Rath, Pascal Grittmann, et al., Siggraph 2020



VEACH DOOR



GLOSSY KITCHEN

