Ray Tracing in Entertainment Industry

Tanaboon Tongbuasirilai
Dept. Computer Science
Kasetsart University

Week 12
Variance reduction

Monte Carlo method

- Issues of using Monte Carlo method
  - Probabilistic approach
    - The quality of the solution depends on the chosen random variable. How ?
  - Slow convergence (Trade-off between accuracy and computational cost).
    - Rate of convergence $O(1/\sqrt{N})$
      - Ex. N = 4*4 Vs N = 8*8

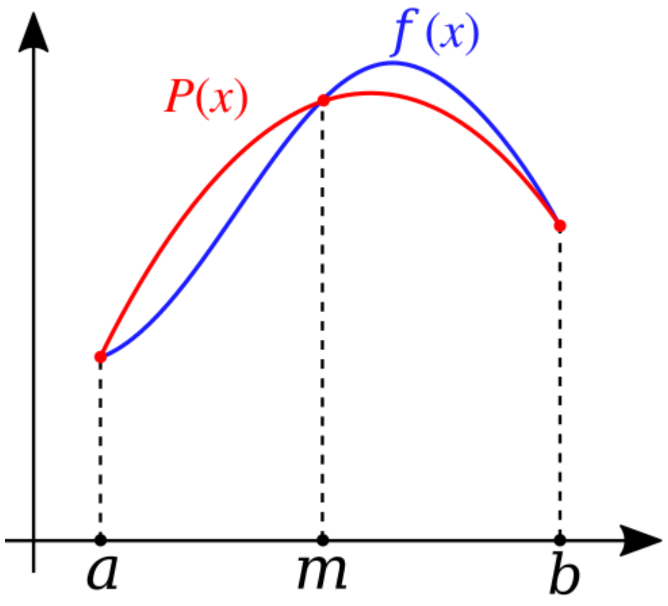
Rate of convergence

- Rate of convergence is a measure of how fast the difference between the solution point and its estimates goes to zero.
- Rate of convergence for Monte Carlo method : $O\left({}^{1}/_{\sqrt{N}}\right)$
    - Ex. N = 4*4 Vs N = 8*8

$$\overline{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i \Rightarrow \mathrm{Var}(\overline{X}_n) = \mathrm{Var}\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) = \frac{1}{n^2}\sum_{i=1}^{n} X_i = \frac{1}{n^2}n\sigma^2 = \frac{\sigma^2}{n}$$

Simpsons' rule

- Simpson's Rule, named after Thomas Simpson though also used by Kepler a century before, was a way to approximate integrals without having to deal with lots of narrow rectangles (which also implies lots of decimal calculations).



$$\int_a^b f(x)\,dx \approx \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]$$

Text from : https://web.stanford.edu/group/sisl/k12/optimization/MO-unit4-pdfs/4.2simpsonintegrals.pdf

Equation from : https://en.wikipedia.org/wiki/Simpson%27s_rule

Image from : By Popletibus This W3C-unspecified vector image was created with Inkscape . - Vectorization of File:Simpsons method illustration.png, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=47202885

Path tracing and Uniform sampling

Applying Monte Carlo method to obtain a path tracing technique could not avoid the downside of the Monte Carlo method.

Previously, we employed the uniform sampling techniques for finding ray scattered directions. This ensures that our integrator could give the correct result (unbiased).

The uniform sampling sacrifices the computational power with the unbiased result.

How could we render scenes faster using a path tracer ?

Expected value and Variance (Recap)

- The **expected value** of a function is defined as the average value of the function over some distribution of values over its domain.

$$E_p[f(x)] = \int_D f(x)\, p(x)\, \mathrm{d}x$$

$$E[\cos x] = \int_0^{\pi} \frac{\cos x}{\pi}\, \mathrm{d}x = \frac{1}{\pi}(\sin \pi - \sin 0) = 0$$

- The **variance** of a function is the expected squared deviation of the function from its expected value.

$$V[f(x)] = E\left[(f(x) - E[f(x)])^2\right]$$

Properties of the expected value and variance (Recap)

$$\boxed{V[f(x)] = E\left[(f(x) - E[f(x)])^2\right]}$$

$$E[af(x)] = aE[f(x)]$$

$$E\left[\sum_i f(X_i)\right] = \sum_i E[f(X_i)]$$

$$V[af(x)] = a^2 V[f(x)].$$

$$\sum_i V[f(X_i)] = V\left[\sum_i f(X_i)\right]$$

$$\boxed{V[f(x)] = E\left[(f(x))^2\right] - E[f(x)]^2}$$

$$F_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)}$$

$$\begin{aligned}
E[F_N] &= E\left[\frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)}\right] \\
&= \frac{1}{N} \sum_{i=1}^{N} \int_a^b \frac{f(x)}{p(x)} p(x)\,\mathrm{d}x \\
&= \frac{1}{N} \sum_{i=1}^{N} \int_a^b f(x)\,\mathrm{d}x \\
&= \int_a^b f(x)\,\mathrm{d}x.
\end{aligned}$$

The estimator of the integral for arbitrary PDF (Recap)

Careful sample placement – Variance reduction 1

Stratified sampling

Quasi Monte Carlo

Stratified sampling

Union of non-overlapping strata.

$$\bigcup_{i=1}^{n} \Lambda_i = \Lambda$$

If we draw samples independently of each stratum,
We get a monte carlo estimate of the stratum.

$$\boxed{F_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \frac{f(X_{i,j})}{p_i(X_{i,j})}}$$

$$\boxed{\mu_i = E\left[f(X_{i,j})\right] = \frac{1}{v_i} \int_{\Lambda_i} f(x)\, \mathrm{d}x}$$

$$\boxed{\sigma_i^2 = \frac{1}{v_i} \int_{\Lambda_i} (f(x) - \mu_i)^2 \mathrm{d}x}$$

$(v_i \in (0,1])$ Fractional volume of stratum i

Variance reduced by the stratified sampling

Overall estimate → $F = \sum_{i=1}^{n} v_i F_i$

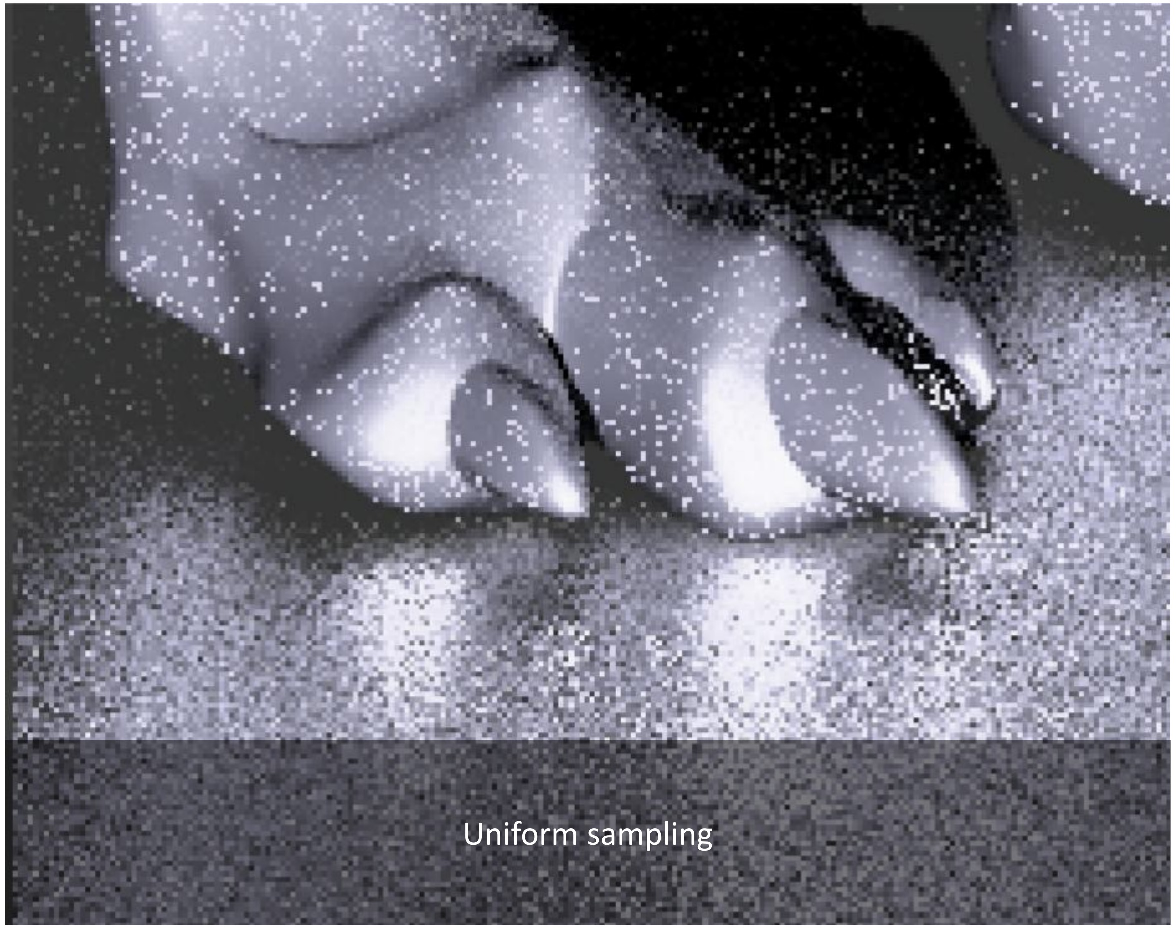
Variance per stratum → σ_i^2 / n_i

The total variance →

$$\begin{aligned} V[F] &= V\left[\sum v_i F_i\right] \\ &= \sum V\left[v_i F_i\right] \\ &= \sum v_i^2 V\left[F_i\right] \\ &= \sum \frac{v_i^2 \sigma_i^2}{n_i}. \end{aligned}$$

What factors impact to total variance ? →

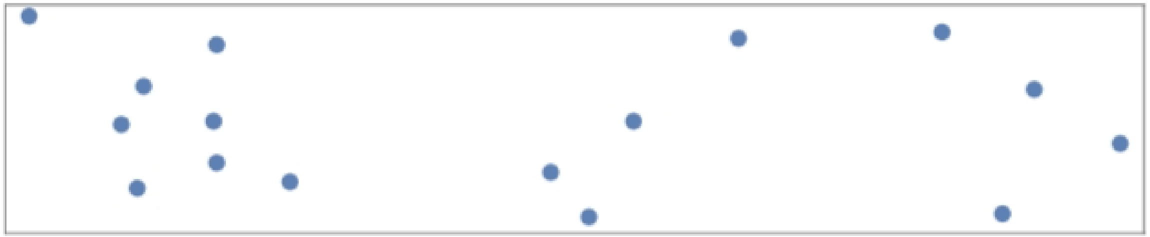
Result



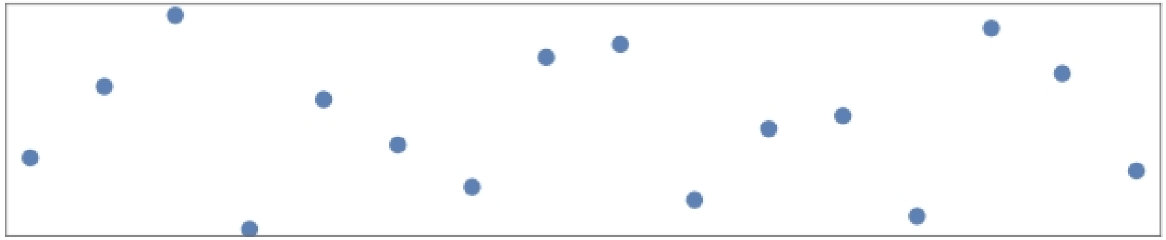
Quasi Monte Carlo sampling

- Low discrepancy sampling
  - Halton sampling
  - Hammersly sampling
  - (0,2) sequence sampling
- New pseudo random generator
  - Quasi random sampling

The difference between stratified and quasi in the pictures ?

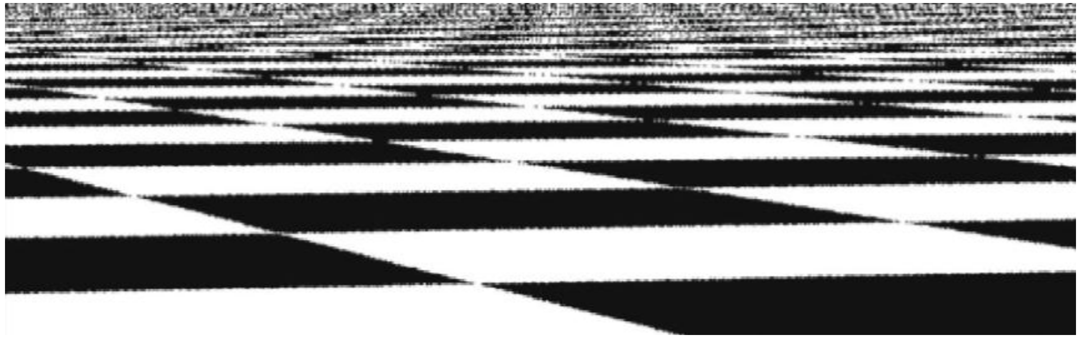


Stratified sampling



Quasi sampling

Comparison



Jittered stratified sampling

Halton sampling

https://www.scratchapixel.com/lessons/mathematics-physics-for-computer-graphics/monte-carlo-methods-in-practice/introduction-quasi-monte-carlo.html

https://pbr-book.org/3ed-2018/Sampling_and_Reconstruction/The_Halton_Sampler

https://pbr-book.org/3ed-2018/Sampling_and_Reconstruction/(0,_2)-Sequence_Sampler

Well-distributed samples

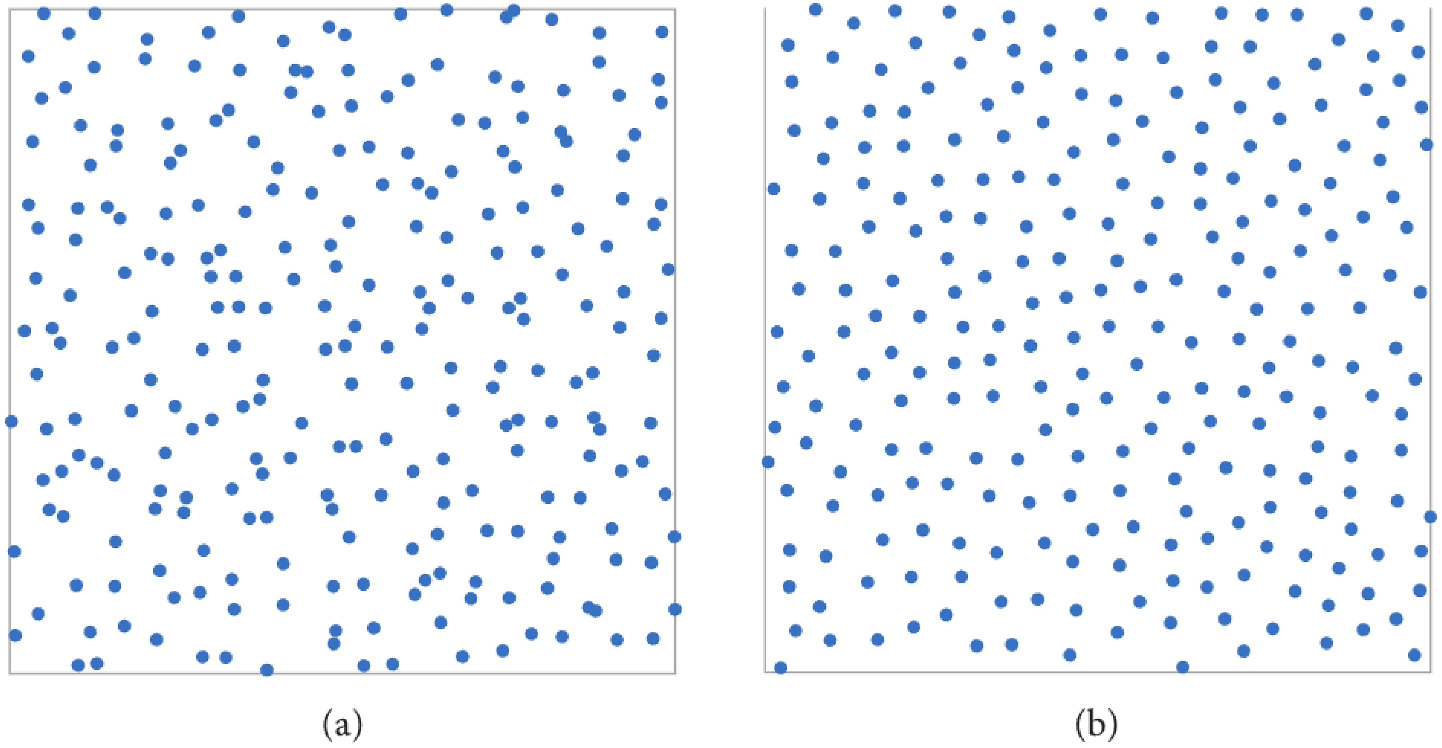


Figure 8.16: 256 sample points distributed using (a) a jittered distribution, and (b) a Poisson disk distribution. Poisson disk point sets combine some randomness in the locations of the points with some structure from no two of them being too close together.

https://pbr-book.org/4ed/Sampling_and_Reconstruction/Sampling_and_Integration

Power Spectral Density (PSD)

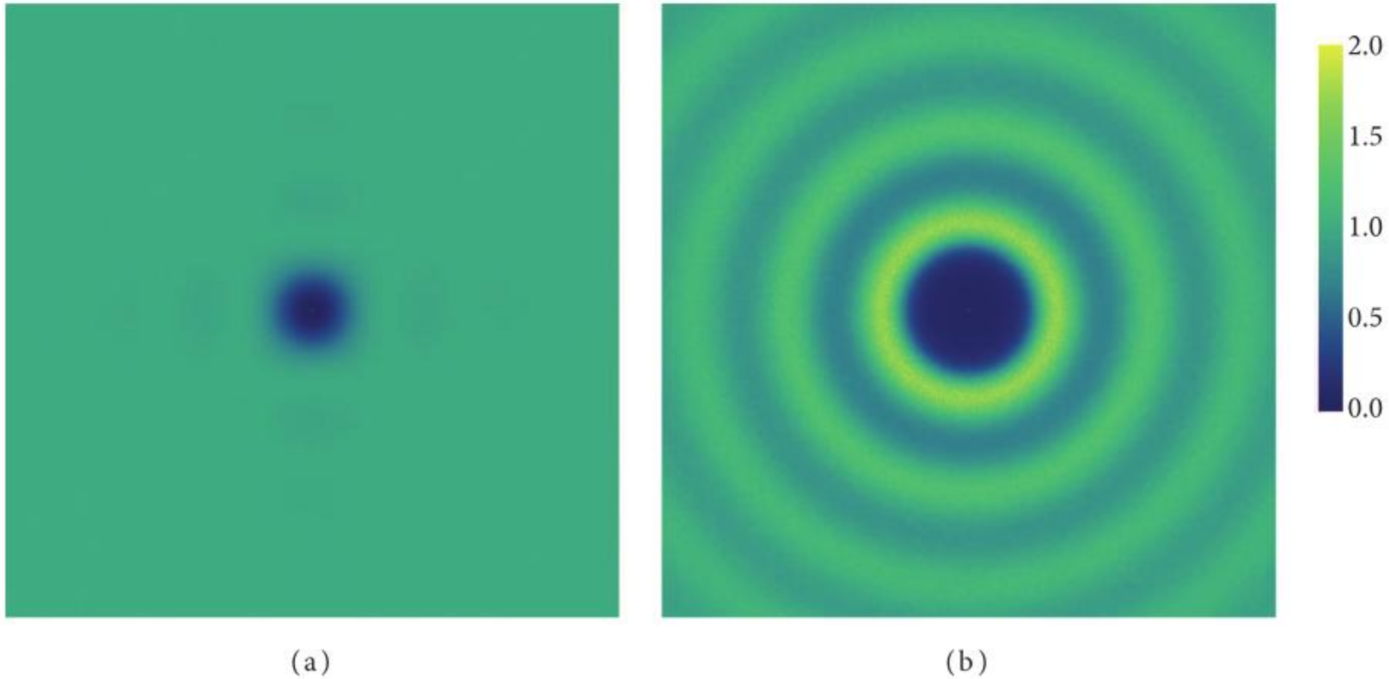


Figure 8.17: PSDs of (a) jittered and (b) Poisson disk–distributed sample points. The origin with the central spike is at the center of each image.

Radially averaged plots

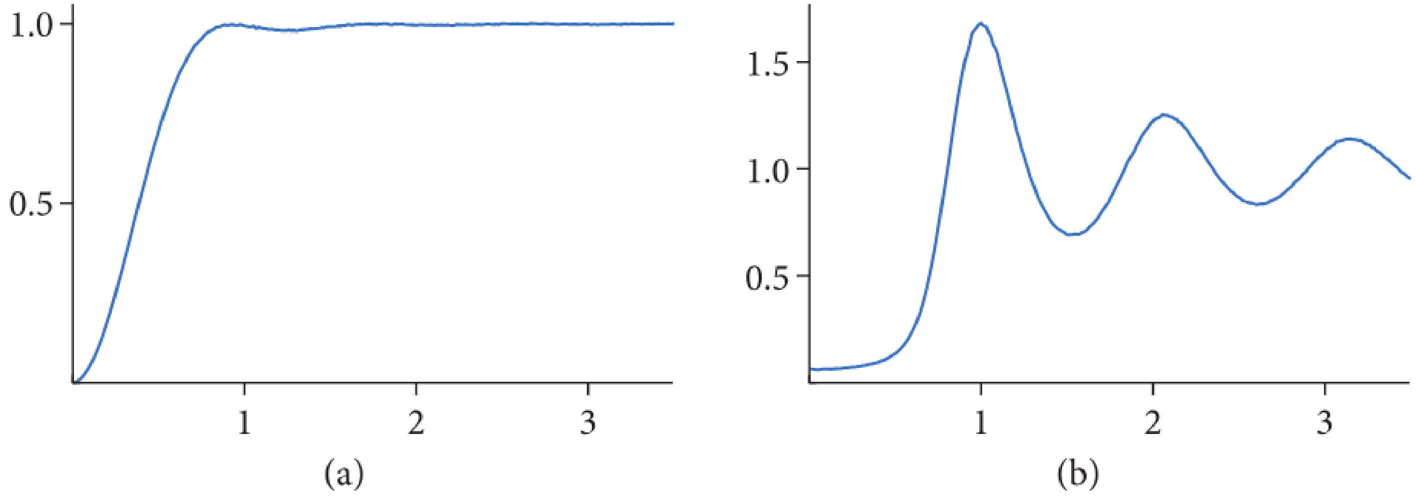


Figure 8.18: Radially averaged PSDs of (a) jittered and (b) Poisson disk–distributed sample points.

Bias – variance reduction 2

An estimator is unbiased if its expected value is equal to the correct answer.

The difference, between the estimator and its expected value, is called **bias**.

$$\boxed{\beta = E[F] - \int f(x)\,\mathrm{d}x}$$

A uniform distribution X_i

mean

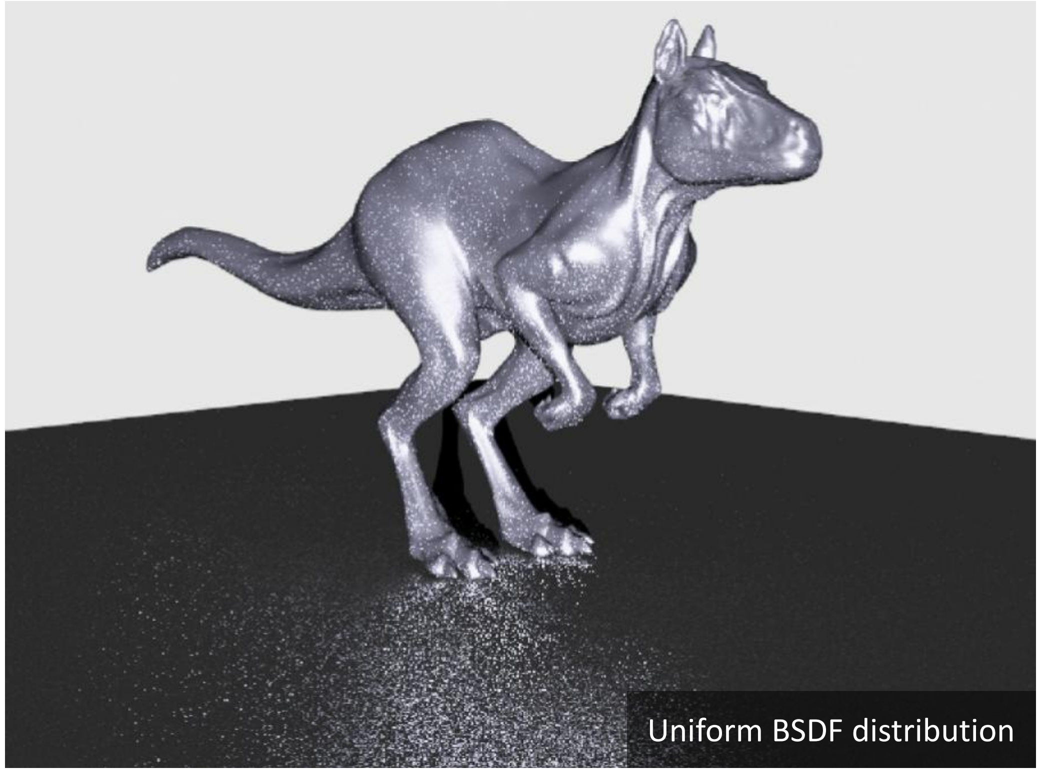
$$\frac{1}{N}\sum_{i=1}^{N} X_i \qquad\qquad 0.5\frac{N}{N+1} \neq 0.5$$

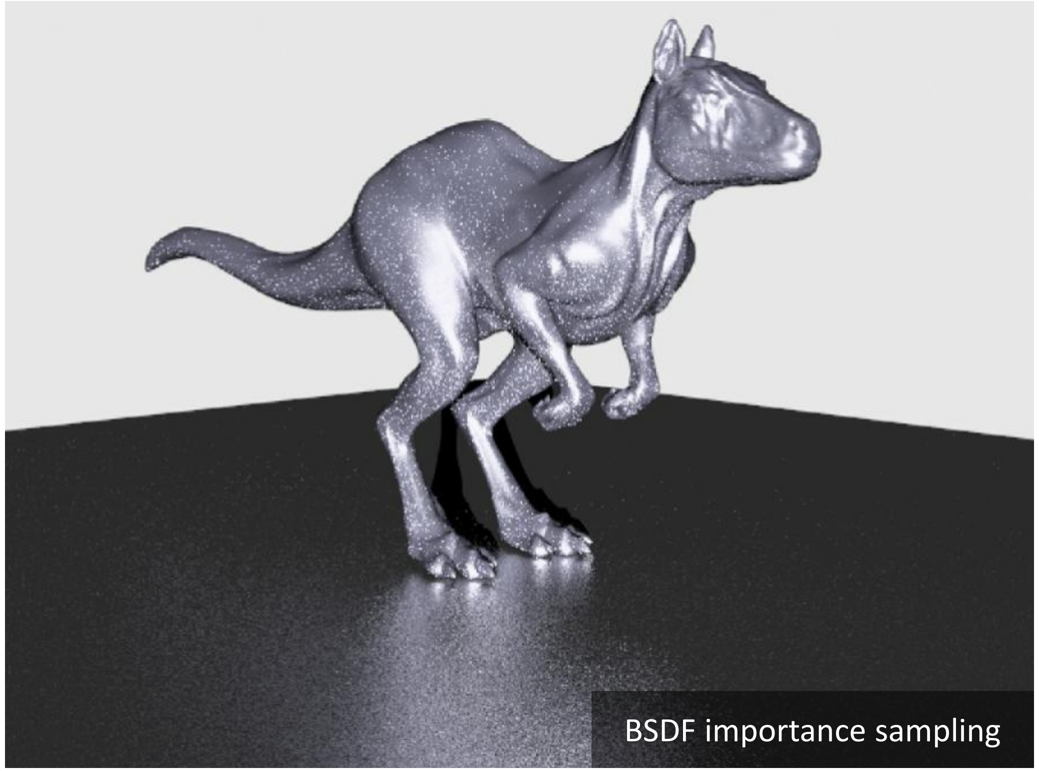
variance

$$O(N^{-1}) \qquad\qquad O(N^{-2})$$

Lower variance

Importance sampling – variance reduction 3





Lower variance through choosing suitable PDF.

$$F_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)}$$

$$\begin{aligned} E[F_N] &= E\left[\frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)}\right] \\ &= \frac{1}{N} \sum_{i=1}^{N} \int_a^b \frac{f(x)}{p(x)} p(x)\, dx \\ &= \frac{1}{N} \sum_{i=1}^{N} \int_a^b f(x)\, dx \\ &= \int_a^b f(x)\, dx. \end{aligned}$$

If the chosen PDF, $p(X_i)$, is similar to the estimated function, $f(X_i)$.

The resulted estimator, F_N, is (almost) a constant.

$$p(x) \propto f(x).$$

$$p(x) = cf(x)$$

Let c be

$$c = \frac{1}{\int f(x)\, dx}$$

$$\frac{f(X_i)}{p(X_i)} = \frac{1}{c} = \int f(x)\, dx$$

Variance = 0

Sampling BRDF (BSDF, BXDF) in PBRT

```
Spectrum BxDF::Sample_f(const Vector3f &wo, Vector3f *wi, const Point2f &u,
                        Float *pdf, BxDFType *sampledType) const {
    // Cosine-sample the hemisphere, flipping the direction if necessary
    *wi = CosineSampleHemisphere(u);
    if (wo.z < 0) wi->z *= -1;
    *pdf = Pdf(wo, *wi);
    return f(wo, *wi);
}

Float BxDF::Pdf(const Vector3f &wo, const Vector3f &wi) const {
    return SameHemisphere(wo, wi) ? AbsCosTheta(wi) * InvPi : 0;
}
```

Blinn sampling

```cpp
void Blinn::Sample_f(const Vector &wo, Vector *wi, float u1, float u2,
                     float *pdf) const {
    // Compute sampled half-angle vector $\wh$ for Blinn distribution
    float costheta = powf(u1, 1.f / (exponent+1));
    float sintheta = sqrtf(max(0.f, 1.f - costheta*costheta));
    float phi = u2 * 2.f * M_PI;
    Vector wh = SphericalDirection(sintheta, costheta, phi);
    if (!SameHemisphere(wo, wh)) wh = -wh;

    // Compute incident direction by reflecting about $\wh$
    *wi = -wo + 2.f * Dot(wo, wh) * wh;

    // Compute PDF for $\wi$ from Blinn distribution
    float blinn_pdf = ((exponent + 1.f) * powf(costheta, exponent)) /
                      (2.f * M_PI * 4.f * Dot(wo, wh));
    if (Dot(wo, wh) <= 0.f) blinn_pdf = 0.f;
    *pdf = blinn_pdf;
}
```

```cpp
float Blinn::Pdf(const Vector &wo, const Vector &wi) const {
    Vector wh = Normalize(wo + wi);
    float costheta = AbsCosTheta(wh);
    // Compute PDF for $\wi$ from Blinn distribution
    float blinn_pdf = ((exponent + 1.f) * powf(costheta, exponent)) /
                      (2.f * M_PI * 4.f * Dot(wo, wh));
    if (Dot(wo, wh) <= 0.f) blinn_pdf = 0.f;
    return blinn_pdf;
}
```

Microfacet Reflection sampling

```
Spectrum MicrofacetReflection::Sample_f(const Vector3f &wo, Vector3f *wi,
                                        const Point2f &u, Float *pdf,
                                        BxDFType *sampledType) const {
    // Sample microfacet orientation $\wh$ and reflected direction $\wi$
    if (wo.z == 0) return 0.;
    Vector3f wh = distribution->Sample_wh(wo, u);
    if (Dot(wo, wh) < 0) return 0.;   // Should be rare
    *wi = Reflect(wo, wh);
    if (!SameHemisphere(wo, *wi)) return Spectrum(0.f);

    // Compute PDF of _wi_ for microfacet reflection
    *pdf = distribution->Pdf(wo, wh) / (4 * Dot(wo, wh));
    return f(wo, *wi);
}

Float MicrofacetReflection::Pdf(const Vector3f &wo, const Vector3f &wi) const {
    if (!SameHemisphere(wo, wi)) return 0;
    Vector3f wh = Normalize(wo + wi);
    return distribution->Pdf(wo, wh) / (4 * Dot(wo, wh));
}
```

Sampling directions proportional to a BRDF model.

- Keys to consider when constructing (choosing) PDF proportional to a BRDF model
  - Parameterization
    - The default sampling setting is computed based on the standard parameterization, i.e., (ω_i, ω_o).
    - In many cases, new parameterization is more efficient to model BRDFs, e.g., half-out parameterization (ω_h, ω_o).
  - Separability of the BRDF model
    - When generating a random direction, a pair of uniform random samples is required.
    - Making the model separable can integrate the default ray tracing pipeline easily.

Model quality

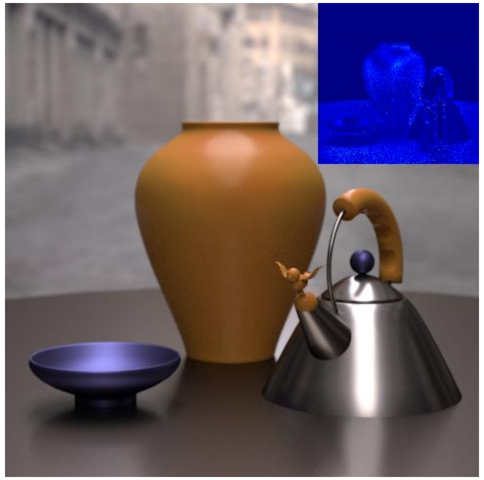
Generated with low number of samples
(Importance sampling)

Lawrence et al.[2004]

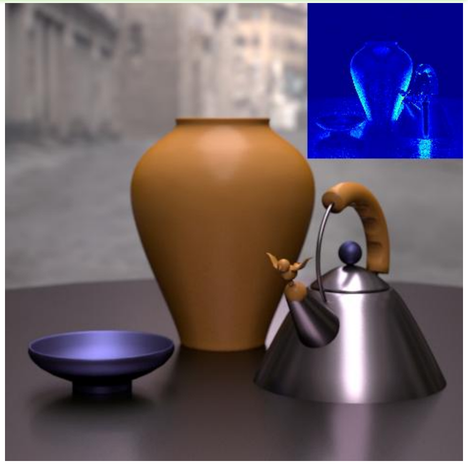
PSNR=33.40dB

Jason Lawrence, Szymon Rusinkiewicz, and Ravi Ramamoorthi. "Efficient BRDF Importance Sampling Using a Factored Representation." ACM Transactions on Graphics (Proc. SIGGRAPH), August 2004.

Our



PSNR=37.37dB



PSNR=35.27dB

Bilgili et al.[2011]

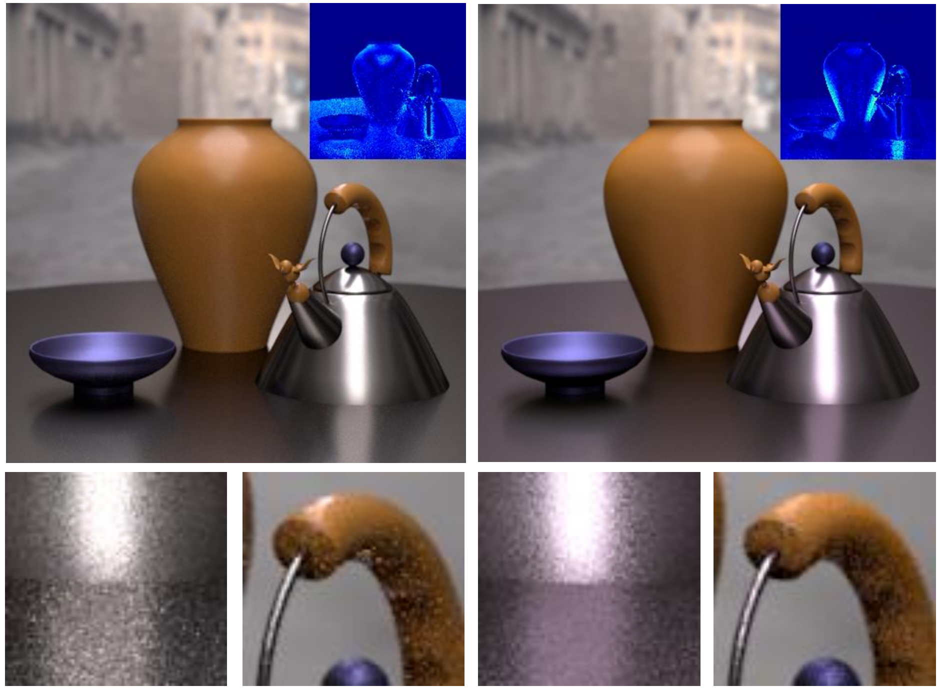
Ahmet Bilgili, Aydn Öztürk, Murat Kurt. A General BRDF Representation Based on Tensor Decomposition, Computer Graphics Forum 30(8):2427-2439, November 2011.

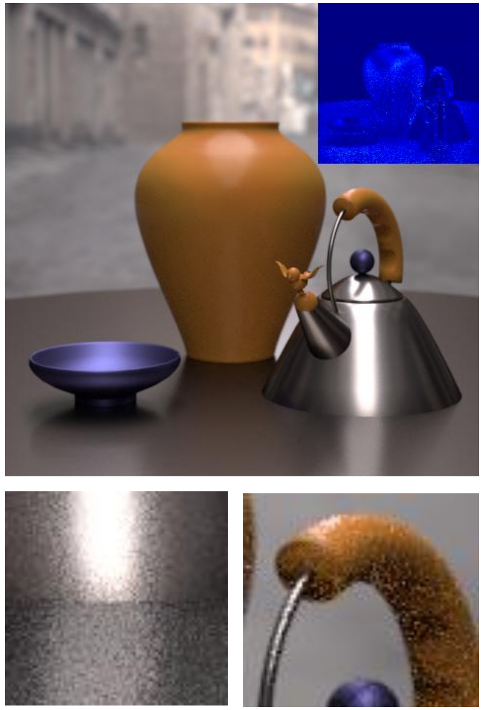
Parameterization affects the rendering quality

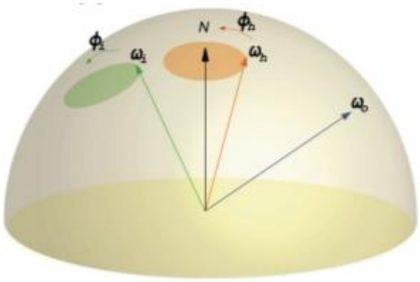
Uncompressed BRDF | In/Out Parameterization | Half/Diff Parameterization

https://diglib.eg.org/bitstream/handle/10.2312/conf.EG2013.tutorials.t6/5_Ruiters_TA_graphics_applications.pdf?sequence=7&isAllowed=y

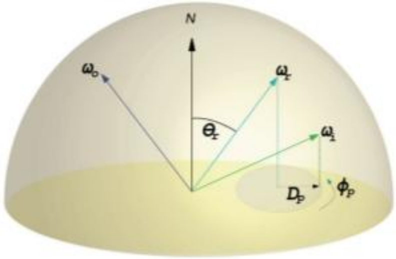
Different parameterizations, different sampling stratergy







$$\gamma_p(z, \phi_p|\omega_o) = \sum_{l=1}^{L} \frac{F_l(\omega_o)\, u_l(z)\, v_l(\phi_p)}{\sum_{j=1}^{L} F_j(\omega_o)}$$



$$\phi_p = 2\pi\xi_1$$
$$d_p = P^{-1}(\xi_2|\theta_r)$$

Transformation between distributions

Given that we want to transform distribution of X space (parameterization into distribution of Y space (another parameterization). How to choose and find such transformation T ?

$$Y = T(X)$$

$$p_y(y) = p_y(T(x)) = \frac{p_x(x)}{|J_T(x)|}$$

$$\begin{pmatrix} \partial T_1/\partial x_1 & \cdots & \partial T_1/\partial x_n \\ \vdots & \ddots & \vdots \\ \partial T_n/\partial x_1 & \cdots & \partial T_n/\partial x_n \end{pmatrix}$$

Example

$$x = r\cos\theta$$
$$y = r\sin\theta.$$

$$J_T = \begin{pmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} \end{pmatrix} = \begin{pmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{pmatrix}$$

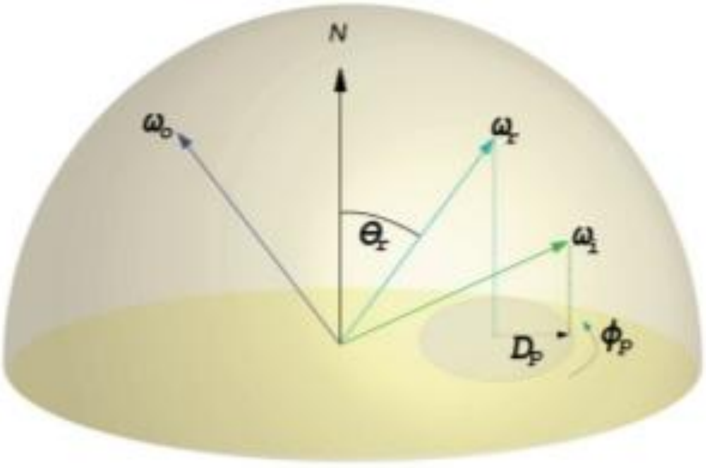
$$p(r,\theta) = r\,p(x,y)$$

***T(X) is bijective.**

$J_T(x)$ is the determinant of the Jacobi transformation of T(X).

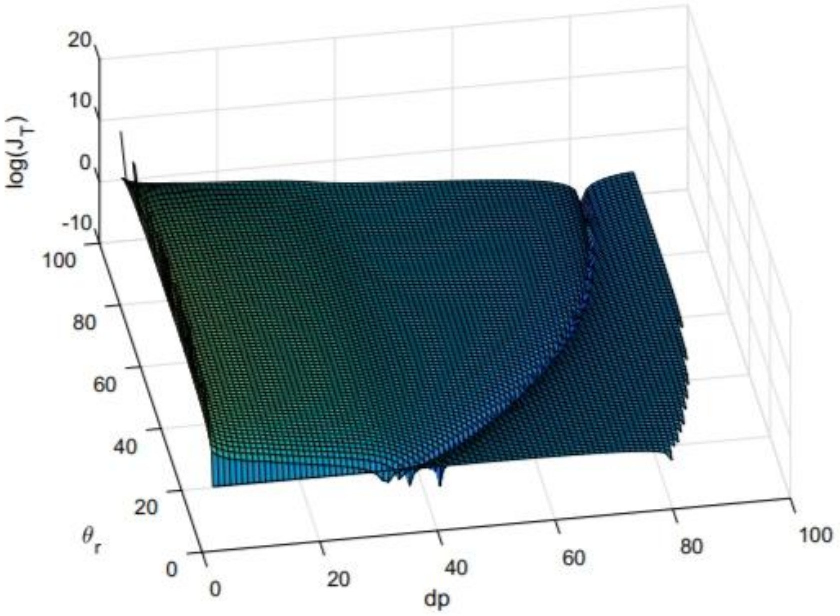
https://pbr-book.org/3ed-2018/Monte_Carlo_Integration/Transforming_between_Distributions

Jacobian transformation of the PDV parameterization



Projected Deviation Vector (PDV)

$$J_T = \frac{\sqrt{2}cos(\theta_i)sin(\theta_i)}{\sqrt{2-cos(2\theta_i)-cos(2\theta_o)+4cos(\phi_i-\phi_o)sin(\theta_i)sin(\theta_o)}}.$$



A mathematical framework for plug in 2 uniform random variables : Example -Uniform Sampling a Hemisphere

A marginal density function

$$p(\theta) = \int_0^{2\pi} p(\theta, \phi)\, \mathrm{d}\phi = \int_0^{2\pi} \frac{\sin\theta}{2\pi}\, \mathrm{d}\phi = \sin\theta$$

The corresponding density

$$p(\phi|\theta) = \frac{p(\theta, \phi)}{p(\theta)} = \frac{1}{2\pi}$$

Separating variables.

Cumulative Distribution Functions

$$P(\theta) = \int_0^{\theta} \sin\theta'\, \mathrm{d}\theta' = 1 - \cos\theta$$

$$P(\phi|\theta) = \int_0^{\phi} \frac{1}{2\pi}\, \mathrm{d}\phi' = \frac{\phi}{2\pi}.$$

Constructing CDFs.

Final outcome for uniform random samples

$$\theta = \cos^{-1}\xi_1$$

$$\phi = 2\pi\xi_2.$$

Cosine-Weighted Hemisphere Sampling

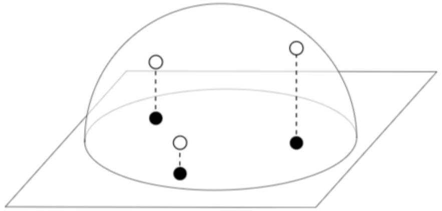
Sample a direction propotional to cosine

$$p(\omega) \propto \cos\theta.$$

The PDF corresponding to the above cosine

$$p(\theta, \phi) = \frac{1}{\pi}\cos\theta\sin\theta.$$

Sampling on a unit disk is more efficient.



$$p(r, \phi) = r/\pi$$

$$(r, \phi) = (\sin\theta, \phi) \rightarrow (\theta, \phi)$$

Want to transform between distributions ?

$$|J_T| = \begin{vmatrix} \cos\theta & 0 \\ 0 & 1 \end{vmatrix} = \cos\theta.$$

$$p(\theta, \phi) = |J_T| p(r, \phi) = \cos\theta\frac{r}{\pi} = (\cos\theta\sin\theta)/\pi$$

$$(r, \theta) = (\sqrt{\xi_1}, 2\pi\xi_2)$$

Microfacet distribution Importance Sampling

Beckmann–Spizzichino distribution $\longrightarrow$ $D(\omega_h) = \dfrac{e^{-\tan^2\theta_h/\alpha^2}}{\pi\alpha^2\cos^4\theta_h}$

$p_h(\theta,\phi)$

$$\boxed{\phi = 2\pi\xi.}$$

separable

$$p_h(\theta) = \frac{2e^{-\tan^2\theta/\alpha^2}\sin\theta}{\alpha^2\cos^3\theta}$$

Constructing a CDF

$$P_h(\theta') = \int_0^{\theta'} \frac{2e^{-\tan^2\theta/\alpha^2}\sin\theta}{\alpha^2\cos^3\theta}\,d\theta$$
$$= 1 - e^{-\tan^2\theta'/\alpha^2}.$$

$$\boxed{\tan^2\theta' = -\alpha^2\log(1-\xi).}$$

Half vector related to the distribution

$$\begin{aligned}
\frac{\mathrm{d}\omega_\mathrm{h}}{\mathrm{d}\omega_\mathrm{i}} &= \frac{\sin\theta_\mathrm{h}\,\mathrm{d}\theta_\mathrm{h}\,\mathrm{d}\phi_\mathrm{h}}{\sin 2\theta_\mathrm{h}\,2\,\mathrm{d}\theta_\mathrm{h}\,\mathrm{d}\phi_\mathrm{h}} \\
&= \frac{\sin\theta_\mathrm{h}}{4\cos\theta_\mathrm{h}\sin\theta_\mathrm{h}} \\
&= \frac{1}{4\cos\theta_\mathrm{h}} \\
&= \frac{1}{4(\omega_\mathrm{i}\cdot\omega_\mathrm{h})} = \frac{1}{4(\omega_\mathrm{o}\cdot\omega_\mathrm{h})}.
\end{aligned}$$

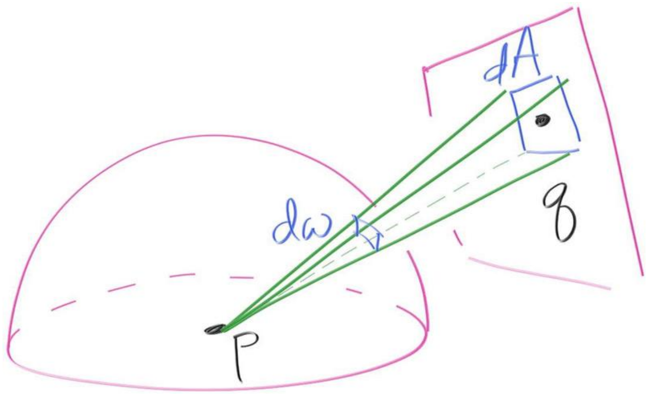
$$p(\theta) = \frac{p_\mathrm{h}(\theta_h)}{4(\omega_\mathrm{o}\cdot\omega_\mathrm{h})}.$$

```
Spectrum MicrofacetReflection::Sample_f(const Vector3f &wo, Vector3f *wi,
                                        const Point2f &u, Float *pdf,
                                        BxDFType *sampledType) const {
    // Sample microfacet orientation $\wh$ and reflected direction $\wi$
    if (wo.z == 0) return 0.;
    Vector3f wh = distribution->Sample_wh(wo, u);
    if (Dot(wo, wh) < 0) return 0.;   // Should be rare
    *wi = Reflect(wo, wh);
    if (!SameHemisphere(wo, *wi)) return Spectrum(0.f);

    // Compute PDF of _wi_ for microfacet reflection
    *pdf = distribution->Pdf(wo, wh) / (4 * Dot(wo, wh));
    return f(wo, *wi);
}

Float MicrofacetReflection::Pdf(const Vector3f &wo, const Vector3f &wi) const {
    if (!SameHemisphere(wo, wi)) return 0;
    Vector3f wh = Normalize(wo + wi);
    return distribution->Pdf(wo, wh) / (4 * Dot(wo, wh));
}
```

Light Sampling – variance reduction 4



3

$$\mathrm{p}(\omega)\cdot d\omega = \mathrm{p_q}(q)\cdot dA$$

$$\mathrm{p}(\omega)\cdot \frac{dA\cdot\cos(\theta)}{\mathrm{distance}^2(p,q)} = \mathrm{p_q}(q)\cdot dA$$

$$d\omega = \frac{dA\cdot\cos(\theta)}{\mathrm{distance}^2(p,q)}$$

1

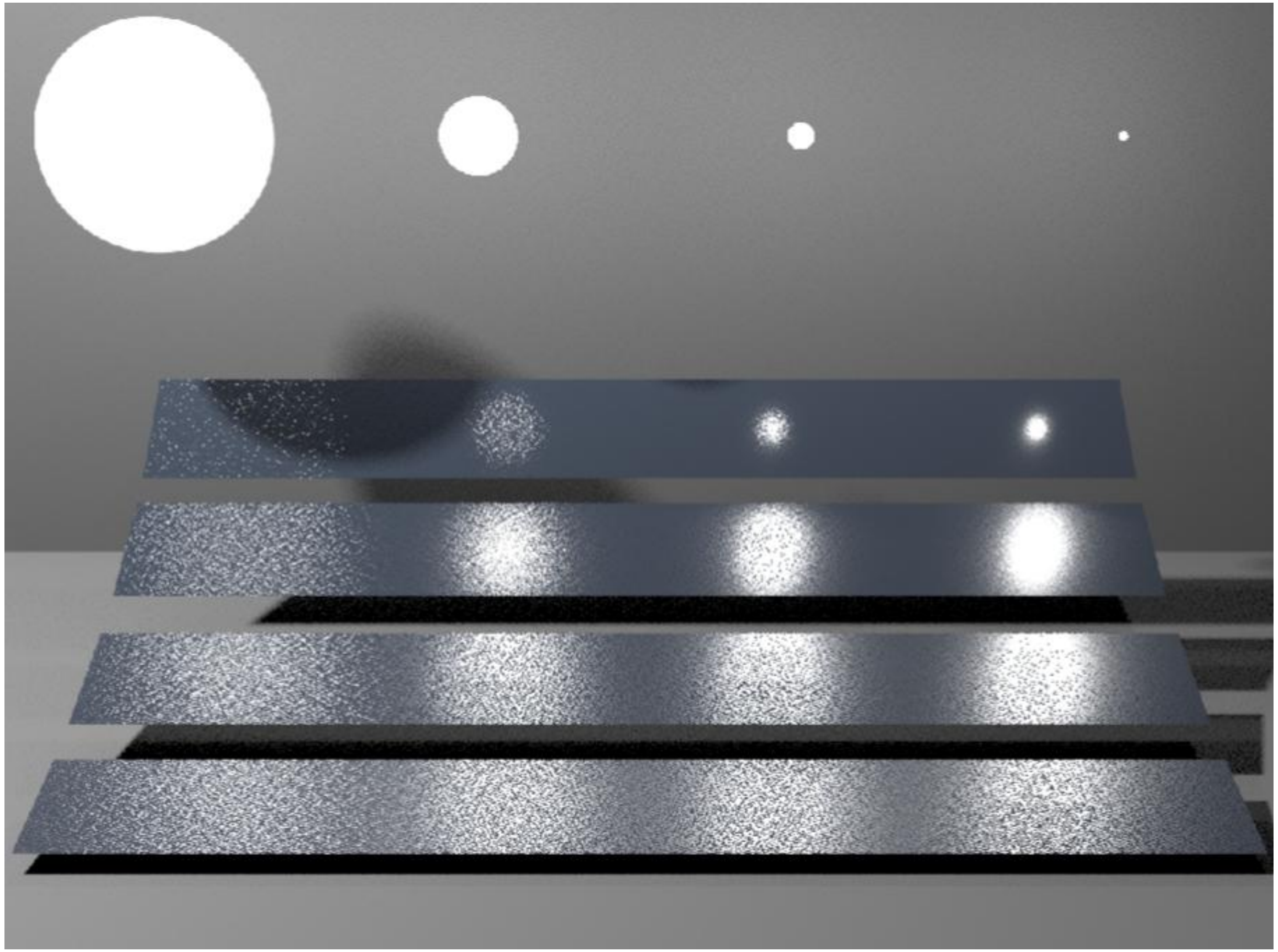
$$\mathrm{p_q}(q) = \frac{1}{A}$$

2

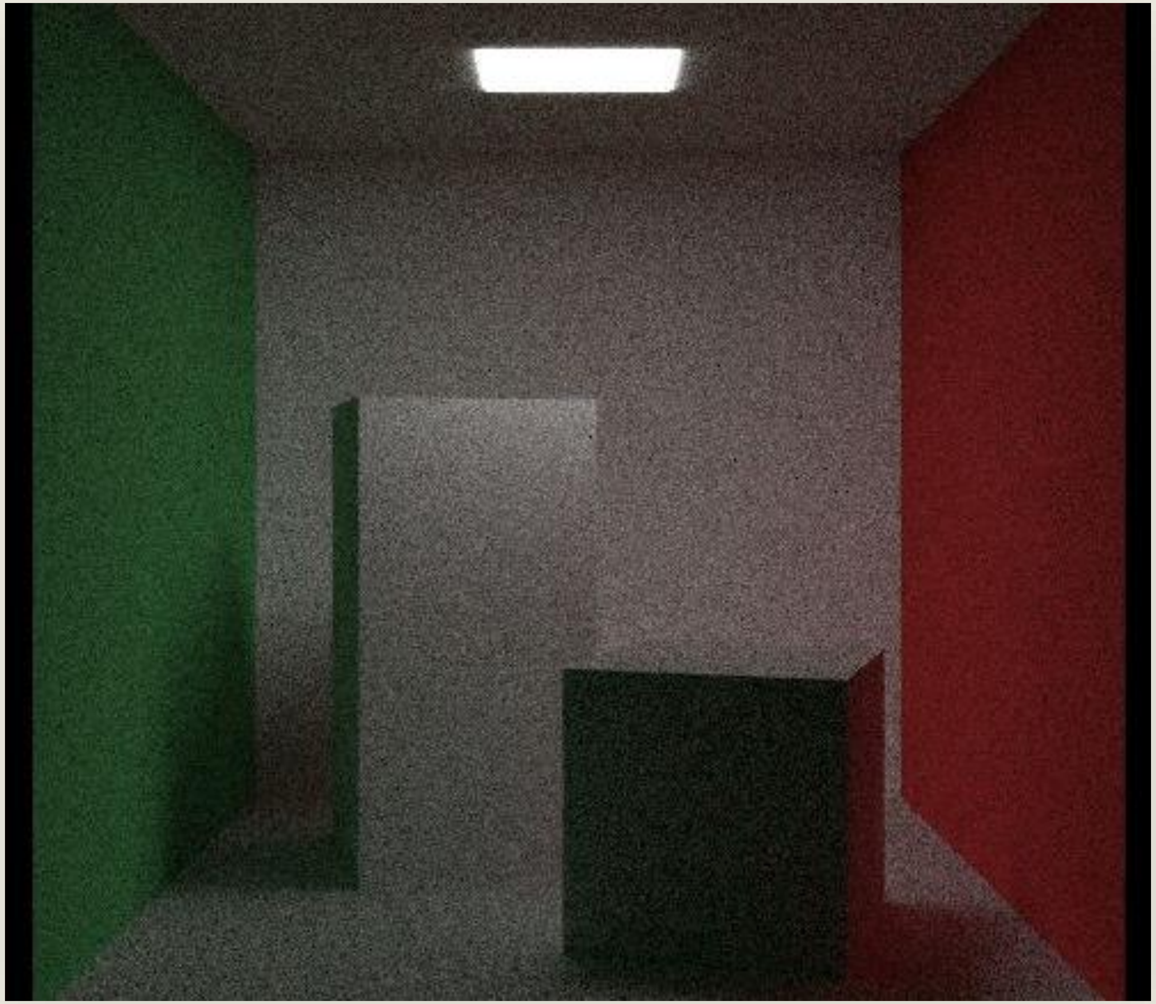
$$\mathrm{p}(\omega) = \frac{\mathrm{distance}^2(p,q)}{\cos(\theta)\cdot A}$$

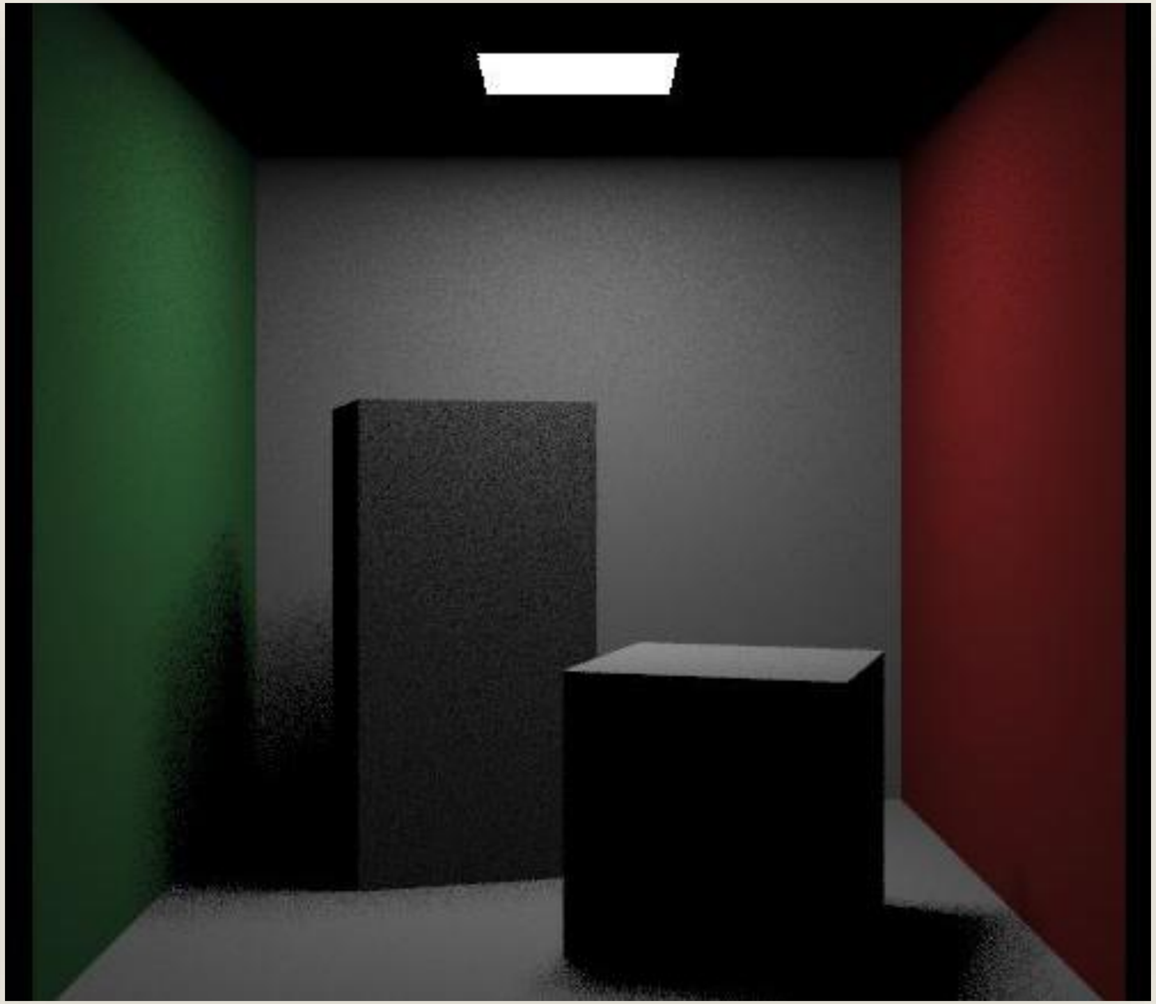
4

Light sampling results

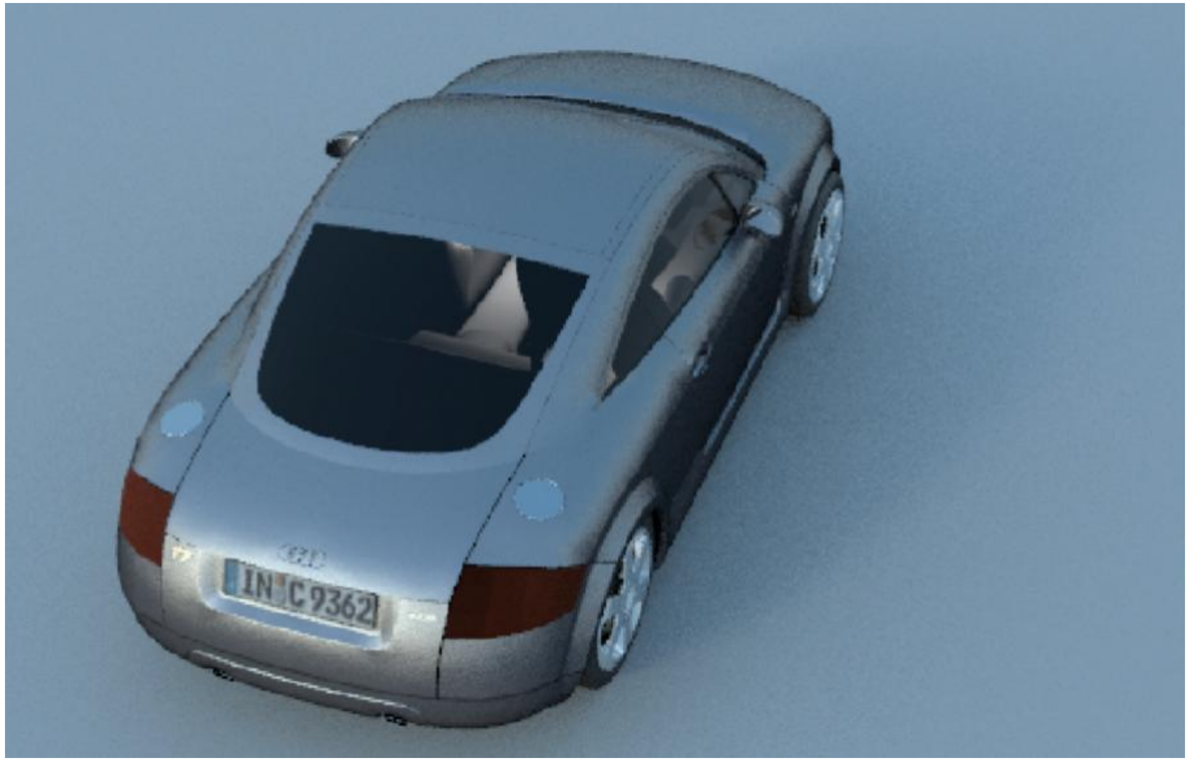


Comparison





More



Keys to do light sampling

Direct light sampling

- Avoid computational overhead for auxiliary paths (scattered directions).
- Ease of implementation.

Point light

- It is a delta distribution in which sampling problem is trivial.
- The PDF is 0 as there is no chance to sample other than only at the point.

Other light shapes

- Each shape requires its own PDF.
  - Quad
  - Triangle
  - Sphere

Multiple Importance Sampling – variance reduction 5

What if the integrand is the product of functions ?

$$\int f(x)g(x)\,\mathrm{d}x.$$

This requires different sampling strategies for each estimator.

Multiple importance sampling (MIS) addresses exactly this issue, with a simple and easy-to-implement technique. The basic idea is that, when estimating an integral, we should draw samples from multiple sampling distributions, chosen in the hope that at least one of them will match the shape of the integrand reasonably well, even if we don't know which one this will be.

https://pbr-book.org/3ed-2018/Monte_Carlo_Integration/Importance_Sampling

The MIS estimator

$$\frac{1}{n_f}\sum_{i=1}^{n_f}\frac{f(X_i)g(X_i)w_f(X_i)}{p_f(X_i)}+\frac{1}{n_g}\sum_{j=1}^{n_g}\frac{f(Y_j)g(Y_j)w_g(Y_j)}{p_g(Y_j)}$$

Choices of weighting functions. (Constraints)

Balance heuristic

$$w_s(x)=\frac{n_s p_s(x)}{\sum_i n_i p_i(x)}$$

Power heuristic

$$w_s(x)=\frac{(n_s p_s(x))^\beta}{\sum_i (n_i p_i(x))^\beta}$$

What each of the estimator means

$$L_o(\mathrm{p}, \omega_o) = \int_{\mathrm{S}^2} f(\mathrm{p}, \omega_o, \omega_i)\, L_d(\mathrm{p}, \omega_i)\, |\cos\theta_i|\, \mathrm{d}\omega_i\,.$$

$$\int f(x) g(x)\, \mathrm{d}x.$$

Sampling from BRDF

Sampling from light

Observing an estimator

$$\frac{f(X)g(X)}{p_f(X)}$$

Choosing $p_f(X)$

If it is similar to f(X), the variance is high due to g(X).

If it is similar to g(X), the variance is high due to f(X).

The weighting functions address this issue.

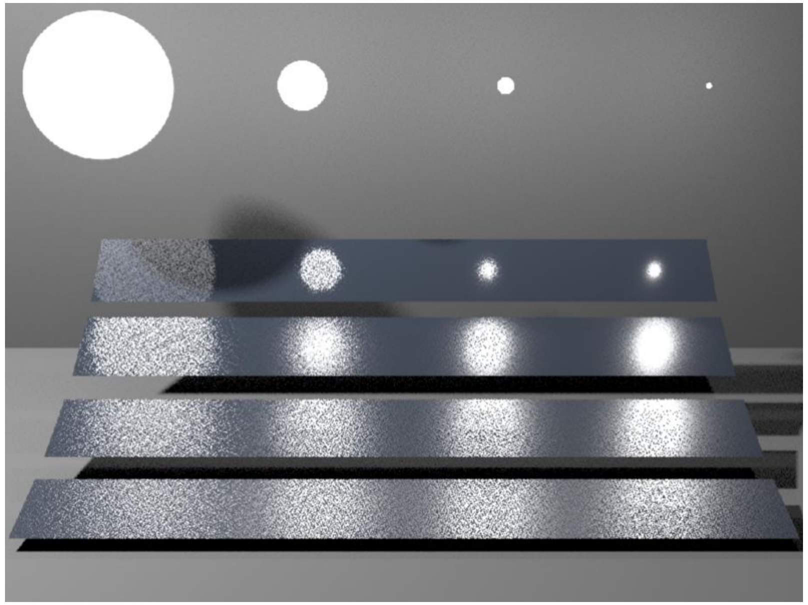
$$\frac{f(X)g(X)w_f(X)}{p_f(X)} = \frac{f(X)g(X)\,n_f\,p_f(X)}{p_f(X)(n_f\,p_f(X) + n_g\,p_g(X))} = \frac{f(X)g(X)\,n_f}{n_f\,p_f(X) + n_g\,p_g(X)}$$

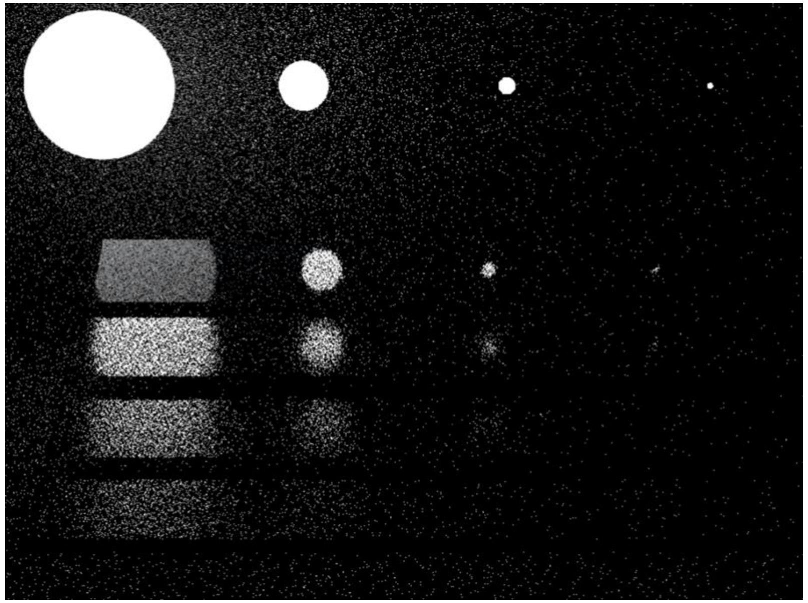
MIS result

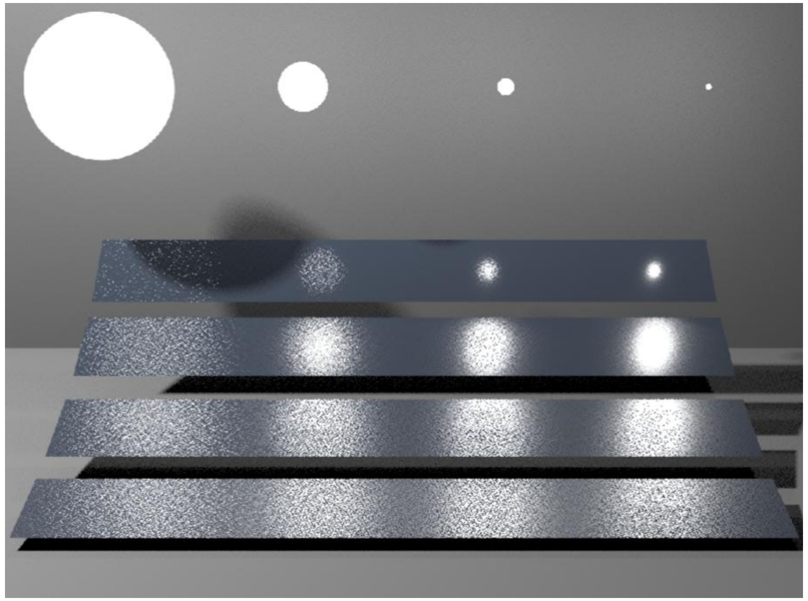
MIS

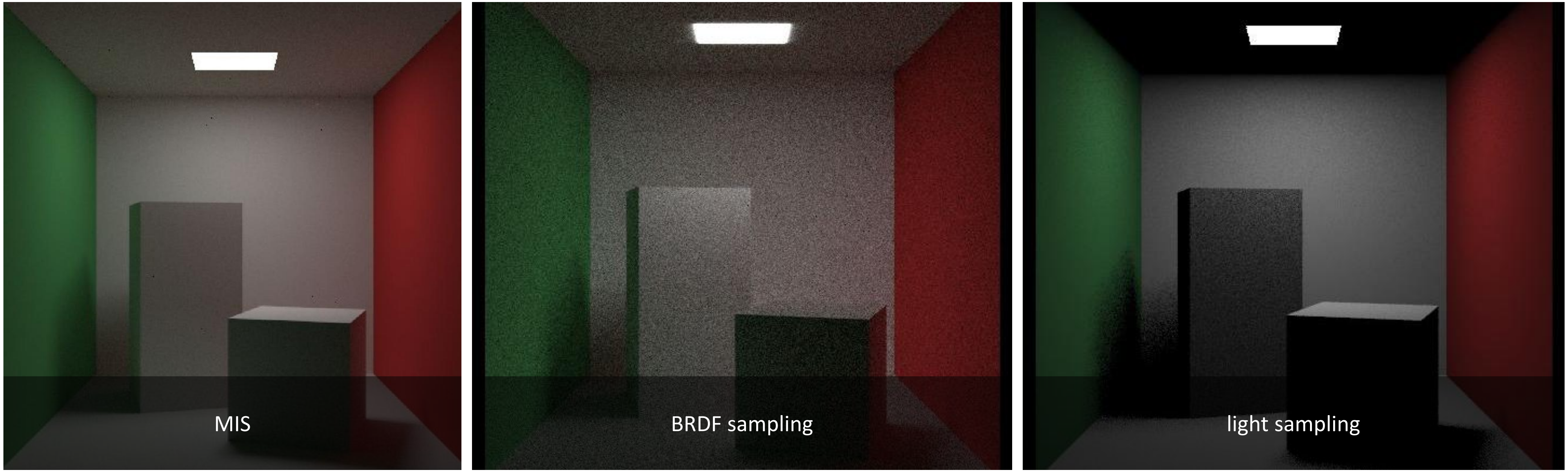
BRDF sampling

Light sampling









Cornell box