CS5630 Physically Based Realistic Rendering

Steve Marschner

“Spring” 2026

09 Multiple Importance Sampling

Problem: Choosing Sampling Strategies

MC local illumination requires a sampling stragegy

- BRDF sampling
- light sampling

BRDF sampling works well under low-frequency lighting (overcast sky)

Light sampling works well on diffuse surfaces lit by concentrated light sources

Neither works well in challenging cases

- scenes with mixed materials and lighting conditions
- sunny day + shiny surfaces

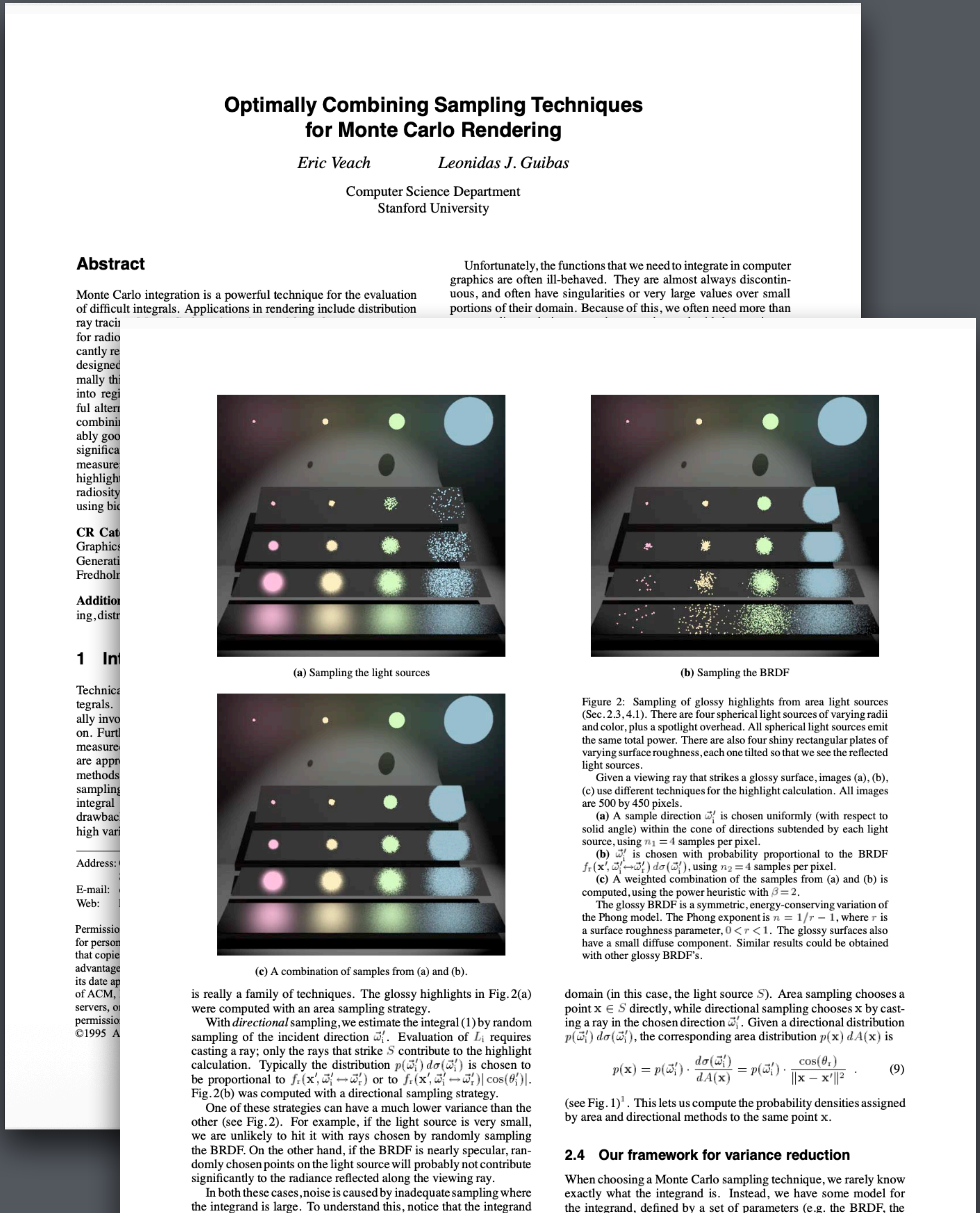
Historical Context

Major problem for 90s renderers

- easy cases: user can select sampling strategy
- some success with heuristics to swtich schemes
- heuristics fail in complex environments

1995: classic paper by Eric Veach

- proposed system for combining estimates
  from different estimators
- proved some optimality results
- applied both to local illumination and
  to bidirectional path tracing (we’ll see later)



Optimally Combining Sampling Techniques for Monte Carlo Rendering

Eric Veach *Leonidas J. Guibas*

Computer Science Department
Stanford University

Abstract

Monte Carlo integration is a powerful technique for the evaluation of difficult integrals. Applications in rendering include distribution

Unfortunately, the functions that we need to integrate in computer graphics are often ill-behaved. They are almost always discontinuous, and often have singularities or very large values over small portions of their domain. Because of this, we often need more than

(a) Sampling the light sources

(b) Sampling the BRDF

(c) A combination of samples from (a) and (b).

Figure 2: Sampling of glossy highlights from area light sources (Sec. 2.3, 4.1). There are four spherical light sources of varying radii and color, plus a spotlight overhead. All spherical light sources emit the same total power. There are also four shiny rectangular plates of varying surface roughness, each one tilted so that we see the reflected light sources.

Given a viewing ray that strikes a glossy surface, images (a), (b), (c) use different techniques for the highlight calculation. All images are 500 by 450 pixels.

(a) A sample direction $\vec{\omega}'_i$ is chosen uniformly (with respect to solid angle) within the cone of directions subtended by each light source, using $n_1 = 4$ samples per pixel.

(b) $\vec{\omega}'_i$ is chosen with probability proportional to the BRDF $f_r(\mathbf{x}', \vec{\omega}'_i \leftrightarrow \vec{\omega}'_r)\, d\sigma(\vec{\omega}'_i)$, using $n_2 = 4$ samples per pixel.

(c) A weighted combination of the samples from (a) and (b) is computed, using the power heuristic with $\beta = 2$.

The glossy BRDF is a symmetric, energy-conserving variation of the Phong model. The Phong exponent is $n = 1/r - 1$, where r is a surface roughness parameter, $0 < r < 1$. The glossy surfaces also have a small diffuse component. Similar results could be obtained with other glossy BRDF’s.

is really a family of techniques. The glossy highlights in Fig. 2(a) were computed with an area sampling strategy.

With *directional* sampling, we estimate the integral (1) by random sampling of the incident direction $\vec{\omega}'_i$. Evaluation of L_i requires casting a ray; only the rays that strike S contribute to the highlight calculation. Typically the distribution $p(\vec{\omega}'_i)\, d\sigma(\vec{\omega}'_i)$ is chosen to be proportional to $f_r(\mathbf{x}', \vec{\omega}'_i \leftrightarrow \vec{\omega}'_r)$ or to $f_r(\mathbf{x}', \vec{\omega}'_i \leftrightarrow \vec{\omega}'_r)|\cos(\theta'_i)|$. Fig. 2(b) was computed with a directional sampling strategy.

One of these strategies can have a much lower variance than the other (see Fig. 2). For example, if the light source is very small, we are unlikely to hit it with rays chosen by randomly sampling the BRDF. On the other hand, if the BRDF is nearly specular, randomly chosen points on the light source will probably not contribute significantly to the radiance reflected along the viewing ray.

In both these cases, noise is caused by inadequate sampling where the integrand is large. To understand this, notice that the integrand

domain (in this case, the light source S). Area sampling chooses a point $\mathbf{x} \in S$ directly, while directional sampling chooses $\mathbf{x}$ by casting a ray in the chosen direction $\vec{\omega}'_i$. Given a directional distribution $p(\vec{\omega}'_i)\, d\sigma(\vec{\omega}'_i)$, the corresponding area distribution $p(\mathbf{x})\, dA(\mathbf{x})$ is

$$p(\mathbf{x}) = p(\vec{\omega}'_i) \cdot \frac{d\sigma(\vec{\omega}'_i)}{dA(\mathbf{x})} = p(\vec{\omega}'_i) \cdot \frac{\cos(\theta_r)}{\|\mathbf{x} - \mathbf{x}'\|^2} \quad . \tag{9}$$

(see Fig. 1)[1]. This lets us compute the probability densities assigned by area and directional methods to the same point $\mathbf{x}$.

2.4 Our framework for variance reduction

When choosing a Monte Carlo sampling technique, we rarely know exactly what the integrand is. Instead, we have some model for the integrand, defined by a set of parameters (e.g. the BRDF, the

A simple viewpoint: Averaging PDFs

A subtle change of viewpoint when using multiple estimators

- doesn't work: choose randomly between estimators derived from sampling procedures A and B
- works: use the sampling procedure "choose between A and B and generate a sample" to define an estimator

Mathematically…

- integrand f; sampling pdfs p_1 and p_2
- separate estimators are $g_1(x) = f(x)/p_1(x)$ and $g_2(x) = f(x)/p_2(x)$
- if I flip a coin then sample from p_1 or p_2 the end-to-end pdf is $p = \frac{1}{2}(p_1 + p_2)$
- unified estimator is $g(x) = 2f(x)/(p_1(x) + p_2(x))$

Why Averaging PDFs Works

Perfect sampling pdf would be proportial to integrand

- estimator is always the same — no variance
- of course this doesn't generally happen

Imperfect sampling behaves differently when too high or too low

- pdf too high: estimator is low (at worst zero)
- pdf too low: estimator is high (and unbounded!)
- differing sample counts do ensure the mean is correct, but low pdfs can cause a lot of variance

The bad case is under-sampled peaks

- average pdf can only be very low when all the pdfs are very low

Example: Sunny Sidewalk Scene

In the shade BRDF sampling works well

- boils down to uniform sampling if the concrete is lambertian

In the sun BRDF sampling fails badly

- very vew samples hit the sun and they get very high values with low (constant) pdf

Sampling just the sun doesn't actually work

- leaves out part of the domain → biased
- slightly off in sun, dramatically off in shade

Combining the two works pretty well

Sergej Karpow

Sunny Sidewalk Numbers

Some data

- sun radius: 7e5 km
- sun-earth distance: 1.5e8 km
- variance of Bernoulli distribution with mean p: $p(1-p)$

To compute

- solid angle of sun
- fraction of hemisphere (aka. probability of hitting sun with BRDF sampling)
- back-of-envelope relative standard deviation of estimator
- number of samples to get 10% error with BRDF sampling

MIS: 50% sun + 50% uniform

- how does it work in each of the cases?

MIS as a weighted average of estimators

Suppose we take two samples, one from each estimator's pdf

- earlier argument leads to:

$$\frac{1}{2}\left(2\frac{f(\omega_1)}{p_1(\omega_1)+p_2(\omega_1)}+2\frac{f(\omega_2)}{p_1(\omega_2)+p_2(\omega_2)}\right)=\frac{f(\omega_1)}{p_1(\omega_1)+p_2(\omega_1)}+\frac{f(\omega_2)}{p_1(\omega_2)+p_2(\omega_2)}$$

- rewriting this a bit, it's a weighted average of the two separate estimators:

$$\left(\frac{p_1(\omega_1)}{p_1(\omega_1)+p_2(\omega_1)}\right)\frac{f(\omega_1)}{p_1(\omega_1)}+\left(\frac{p_2(\omega_2)}{p_1(\omega_2)+p_2(\omega_2)}\right)\frac{f(\omega_2)}{p_2(\omega_2)}$$

- naming the weights w_1 and w_2; note that $w_1 + w_2 = 1$

$$w_1(\omega_1)\frac{f(\omega_1)}{p_1(\omega_1)}+w_2(\omega_2)\frac{f(\omega_2)}{p_2(\omega_2)}=w_1(\omega_1)g_1(\omega_1)+w_2(\omega_2)g_2(\omega_2)$$

- this makes it obvious that when $E\{g_1\} = E\{g_2\}$ then $g(\omega_1, \omega_2)$ has the same expectation
- …and in fact this is valid for any pair of weights that sums to 1

MIS in the canonical form

Start with n **estimators** $g_1, \ldots, g_n$ **with their pdfs** $p_1, \ldots, p_n$

Draw samples $x_1, \ldots, x_n$ **from the respective pdfs**

Evaluate the estimators $g_i(x_i)$

Evaluate the weights $w_1(x_1), \ldots, w_n(x_n)$

- Veach's "power heuristic" is commonly used: $w_i(x_i) = \dfrac{p_i(x_i)^\beta}{\sum_j p_j(x_i)^\beta}$
- $\beta = 1$ produces the pdf-averaging strategy, call the "balance heuristic;" $\beta = 2$ also useful

Compute combined estimator $g = \sum_i w_i(x_i) g_i(x_i)$

Implementation Considerations

Without MIS, only need to evaluate estimators

- sometimes terms cancel in the ratio $f(x)/p(x)$ which is handy
- we only need to think about $p(x)$ for the x that we sampled from p

With MIS, you need to have $p(x)$ separately

- when computing estimator need to return $p(x)$ as well as that ratio
- need separate code to compute $p(x)$ for xs that we **did not** sample from p

Common arrangement for an object that represents a function f:

- evaluate(x) → f(x)
- sample(seed) → x, g(x), p(x) — where g(x) = f(x)/p(x)
- pdf(x) → p(x)

Example interfaces

```
AreaLight {
   eval(y : pt, ω : dir) → L : radiance
   sample(x : pt, seed : vec2) → [y : pt, g : intensity, p : 1/area]
   pdf(y : pt) → p : 1/area
   N(y : pt) → vec3
}

BRDF {
   eval(ω1 : dir, ω2 : dir) → f : 1/sr
   sample(ω1 : dir, seed : vec2) → [ω2 : dir, g : unitless, p : 1/sr]
   pdf(ω1 : dir, ω2 : dir) → p : 1/sr
}

EnvironmentLight {
   eval(ω : dir) → L : radiance
   sample(seed : vec2) → [ω : dir, g : irradiance, p : 1/sr]
   pdf(ω : dir) → p : 1/sr
}
```

Strategies for direct illumination

area source sampling

```
Color shade(x, V, brdf, N, seed) {
    result = black;
    g_y, y, p_y = light.sample(x, seed);
    ω = normalize(y - x);
    if visible(x, y) {
        g = g_y * (-ω · light.N(y))
            / distSqr(x, y);
        f_r = brdf.eval(V, ω);
        result += g * f_r * ω · N;
    }
    return result;
}
```

BRDF sampling

```
Color shade(x, V, brdf, N, seed) {
    result = black;
    g, ω, p_ω = brdf.sample(V, seed);
    y = light.intersect(x, ω);
    if (y and visible(x, y)) {
        L = light.eval(y, -ω)
        result += L * g * ω · N;
    }
    return result;
}
```

What alternative design could avoid the change of measure?
How does this code behave when the source is small and far?
How would it change if the light just returned y and p?

Why is there a minus sign?
How does this code behave when the source is small and far?

MIS for direct illumination

area source sampling

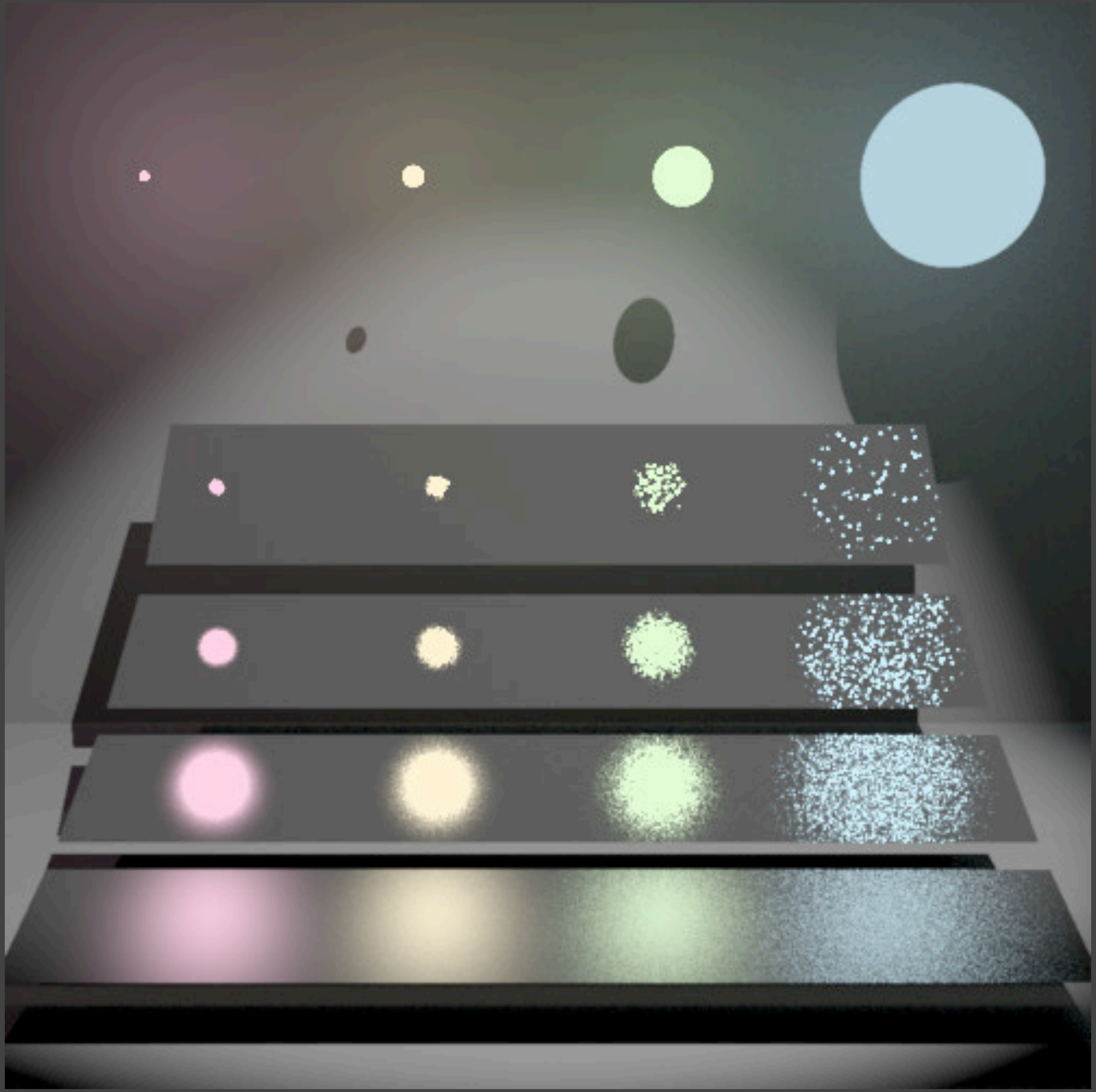
```
Color shade(x, V, brdf, N, seed) {
    result = black;
    g_y, y, p_y = light.sample(x, seed);
    ω = normalize(y – x);
    if visible(x, y) {
        g = g_y * (–ω · light.N(y))
            / distSqr(x, y);
        f_r = brdf.eval(V, ω);
        result += g * f_r * ω · N;
    }
    return result;
}
```

multiple importance sampling

```
Color shade(x, V, brdf, N, seed) {
    result = black;
    g_y, y, p_y = light.sample(x, seed);
    ω_l = normalize(y – x);
    if visible(x, y) {
        G = (–ω_l · light.N(y)) / distSqr(x, y)
        g_l = g_y * G;
        p_l = p_y / G;
        f_r = brdf.eval(V, ω_l);
        p_b = brdf.pdf(V, ω_l);
        w_l = p_l / (p_b + p_l);
        result += w_l * g_l * f_r * ω_l · N;
    }
    g_b, ω_b, p_b = brdf.sample(V, seed);
    y_b = light.intersect(x, ω_b);
    if (y_b and visible(x, y_b)) {
        G = (–ω_b · light.N(y)) / distSqr(x, y_b)
        L = light.eval(y, –ω_b)
        p_l = light.pdf(y_b) / G
        w_b = p_b / (p_b + p_l)
        result += w_b * L * g_b * ω_b · N;
    }
    return result;
}
```

BRDF sampling

```
Color shade(x, V, brdf, N, seed) {
    result = black;
    g, ω, p_ω = brdf.sample(V, seed);
    y = light.intersect(x, ω);
    if (y and visible(x, y)) {
        L = light.eval(y, –ω)
        result += L * g * ω · N;
    }
    return result;
}
```



Veach thesis, 1997

sampling the luminaires

Veach thesis, 1997

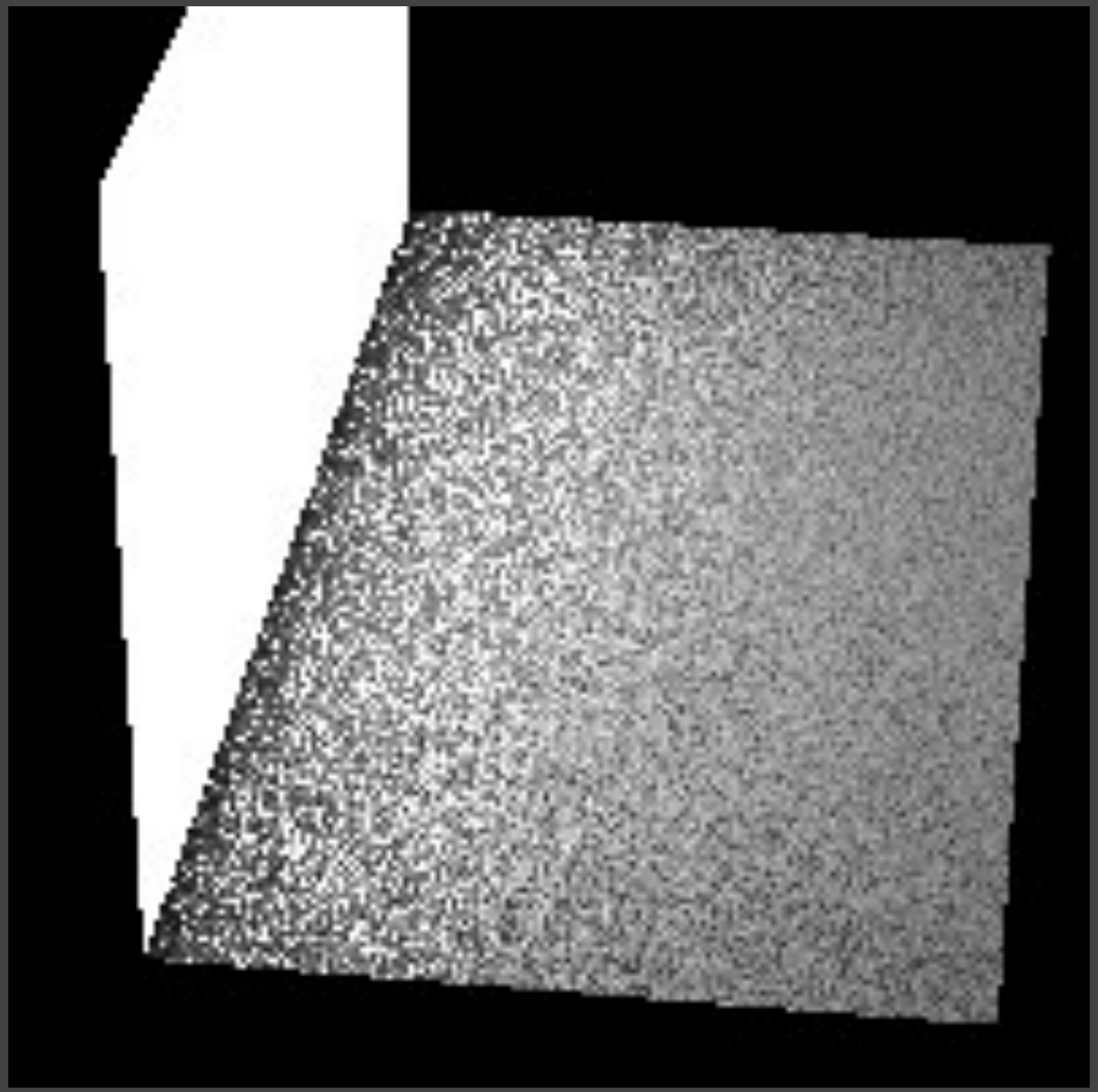
sampling the BRDF

Veach thesis, 1997

sampling sum of pdfs (balance heuristic)

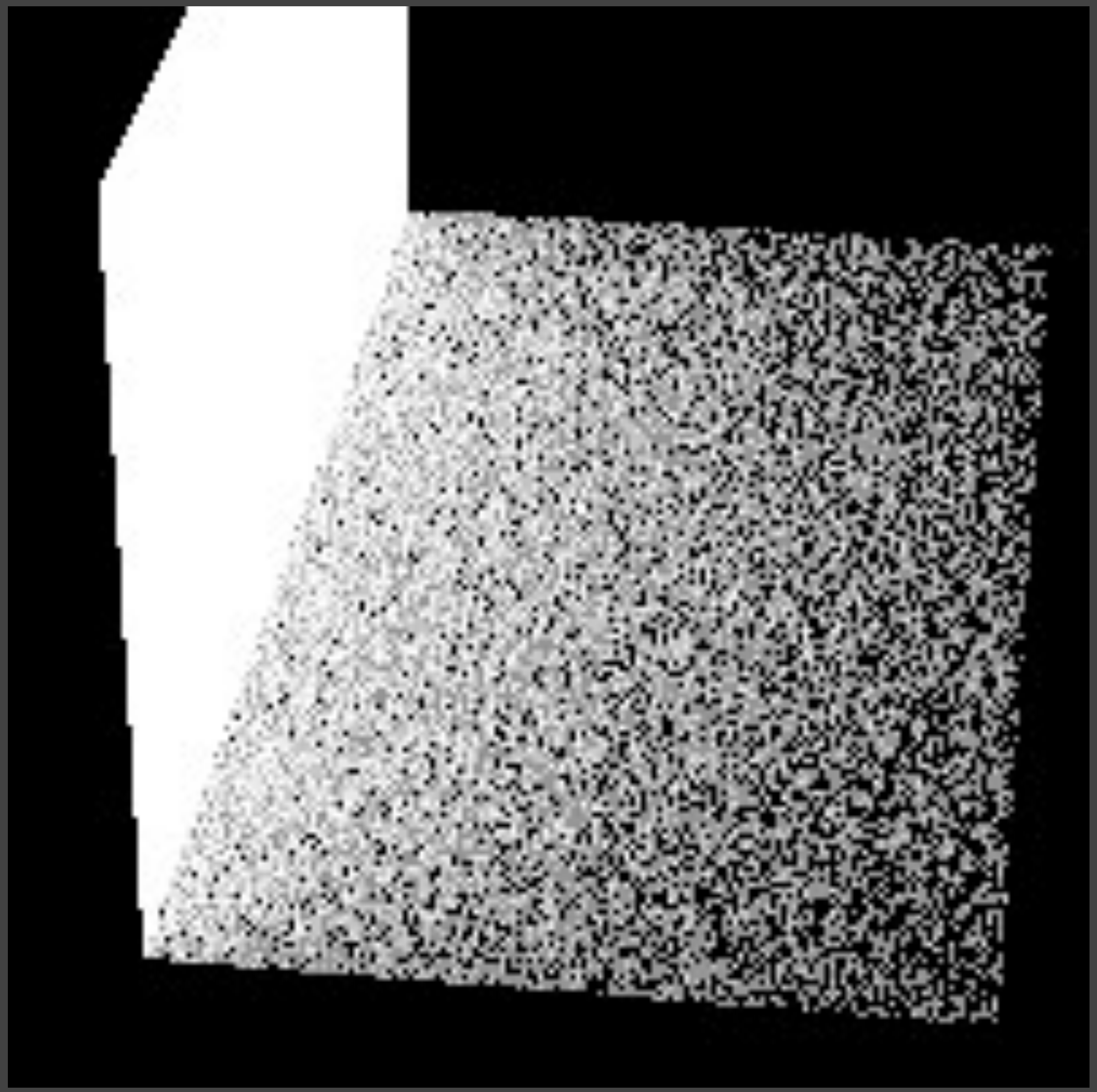
Veach thesis, 1997

combining samples with power heuristic



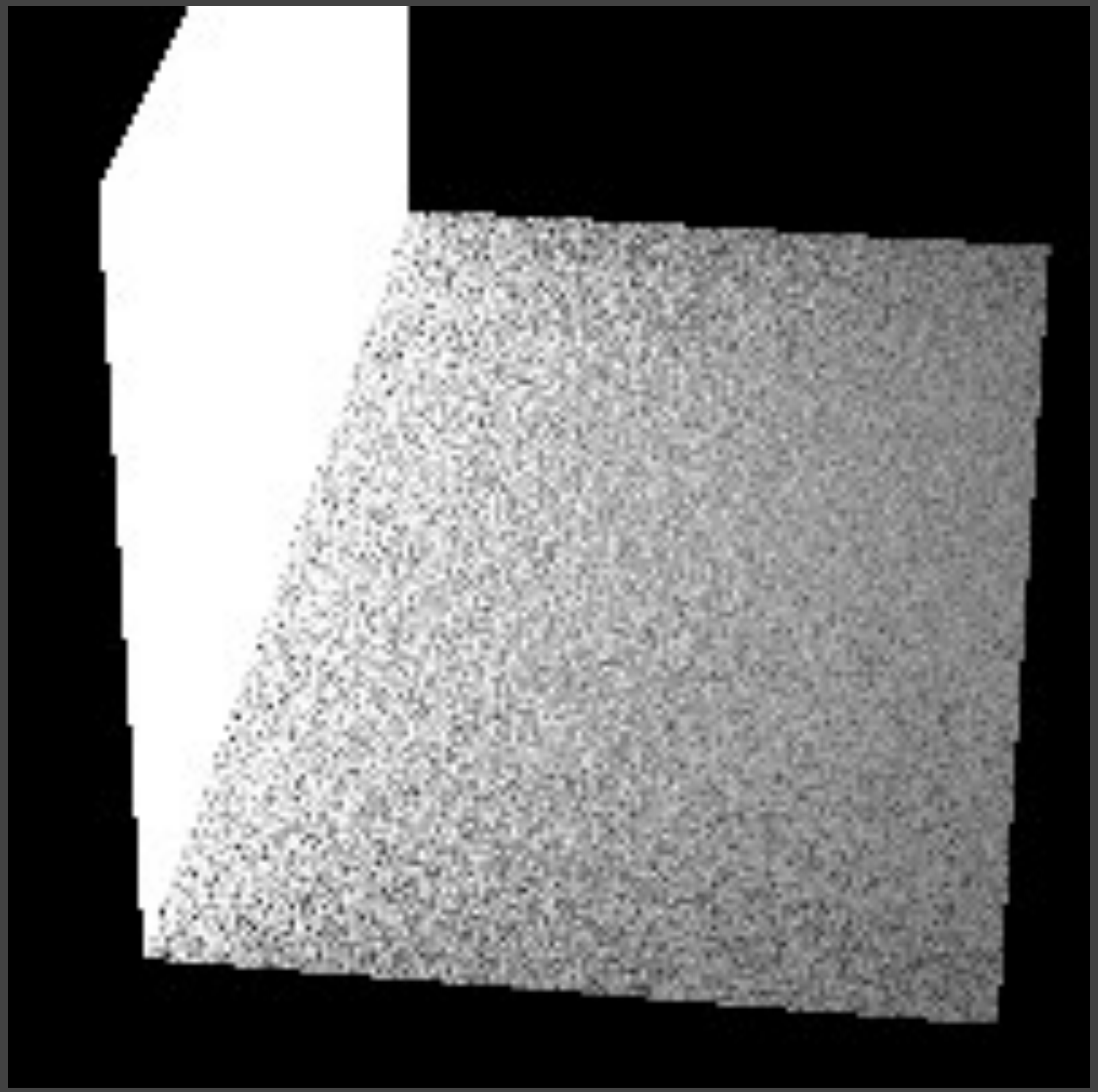
source area sampling

r^{-2} term causes high variance at left



cosine sampling

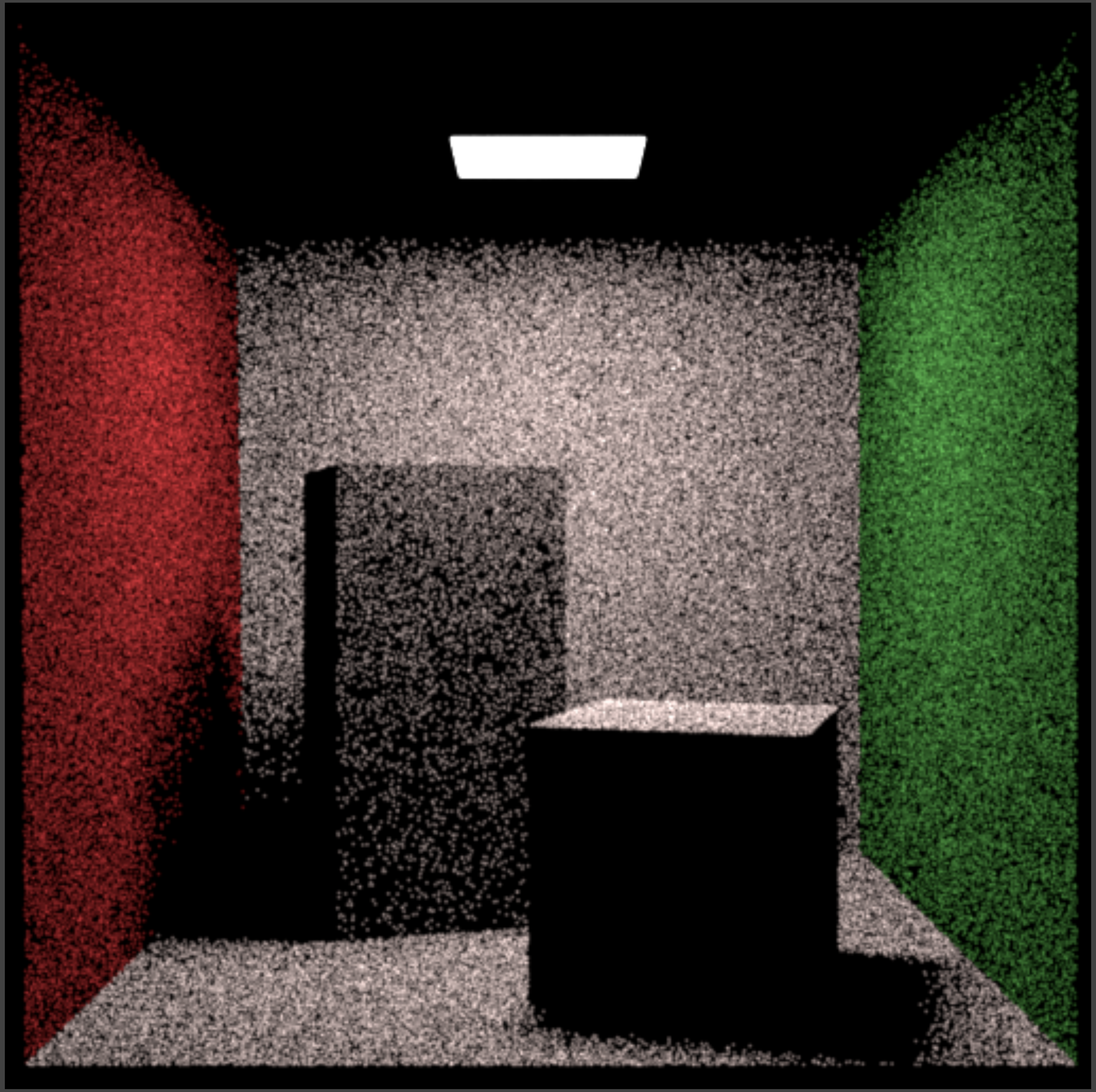
source solid angle causes high variance at right



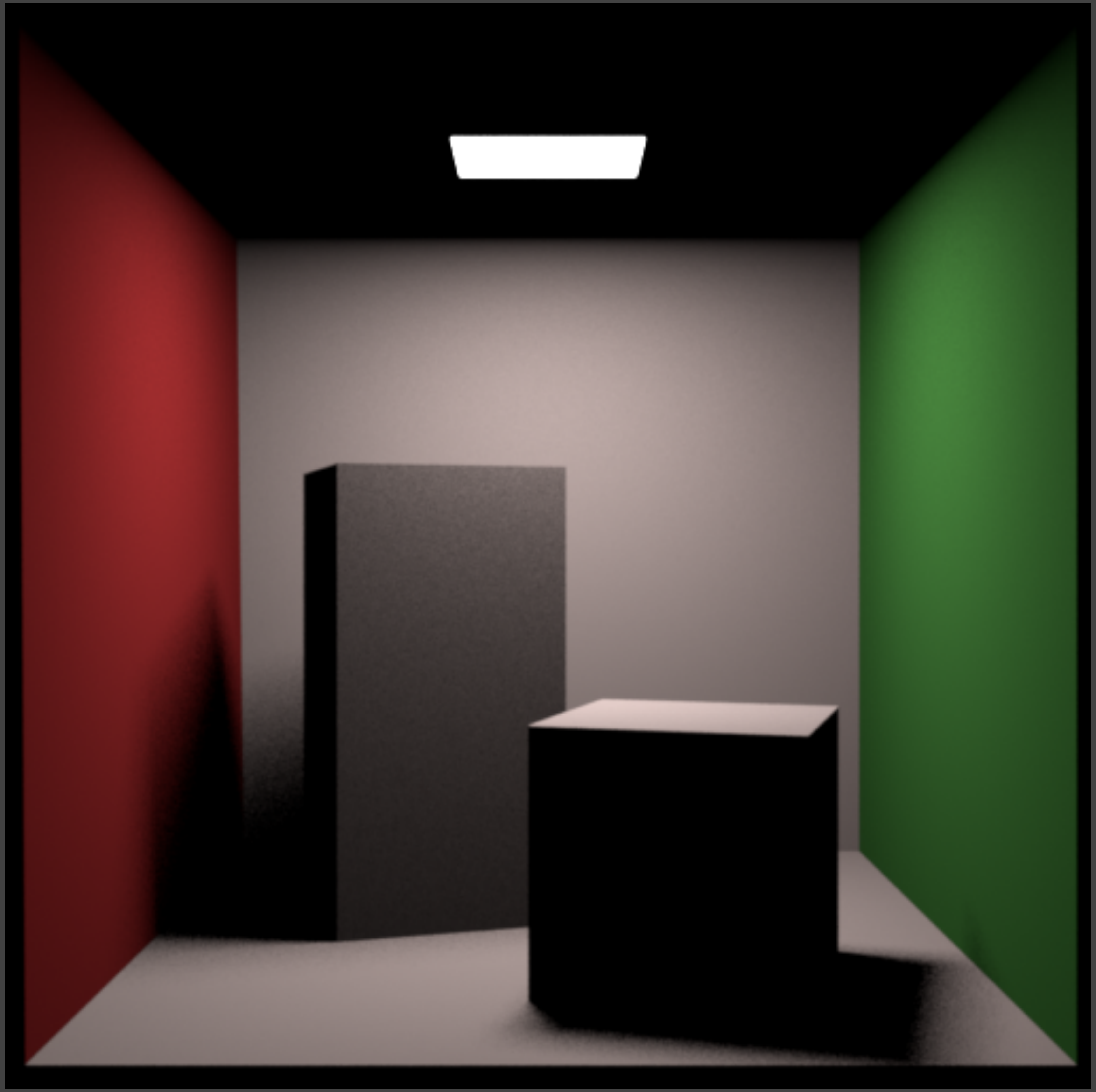
MIS

moderate variance everywhere

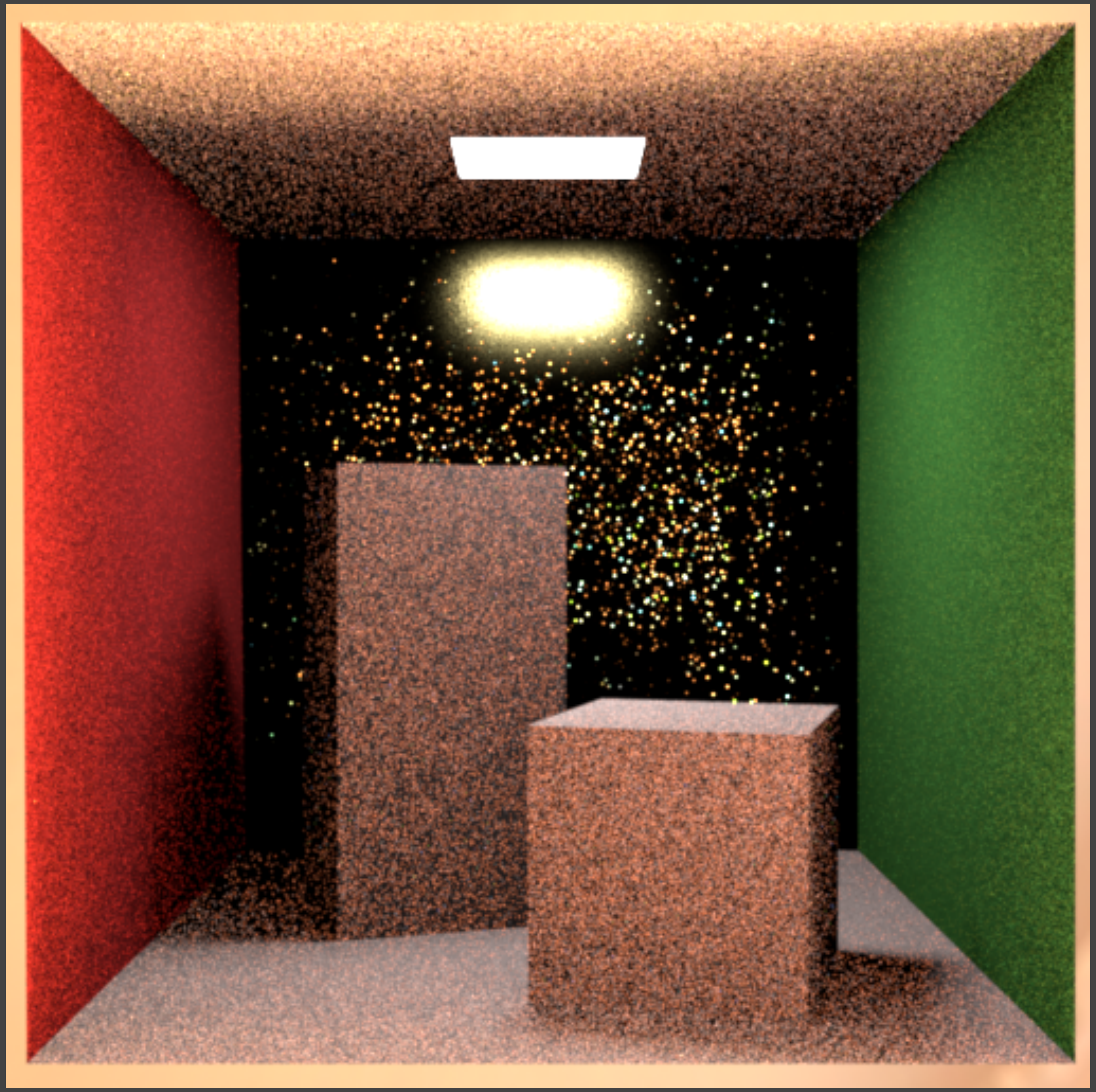
Veach thesis, 1997



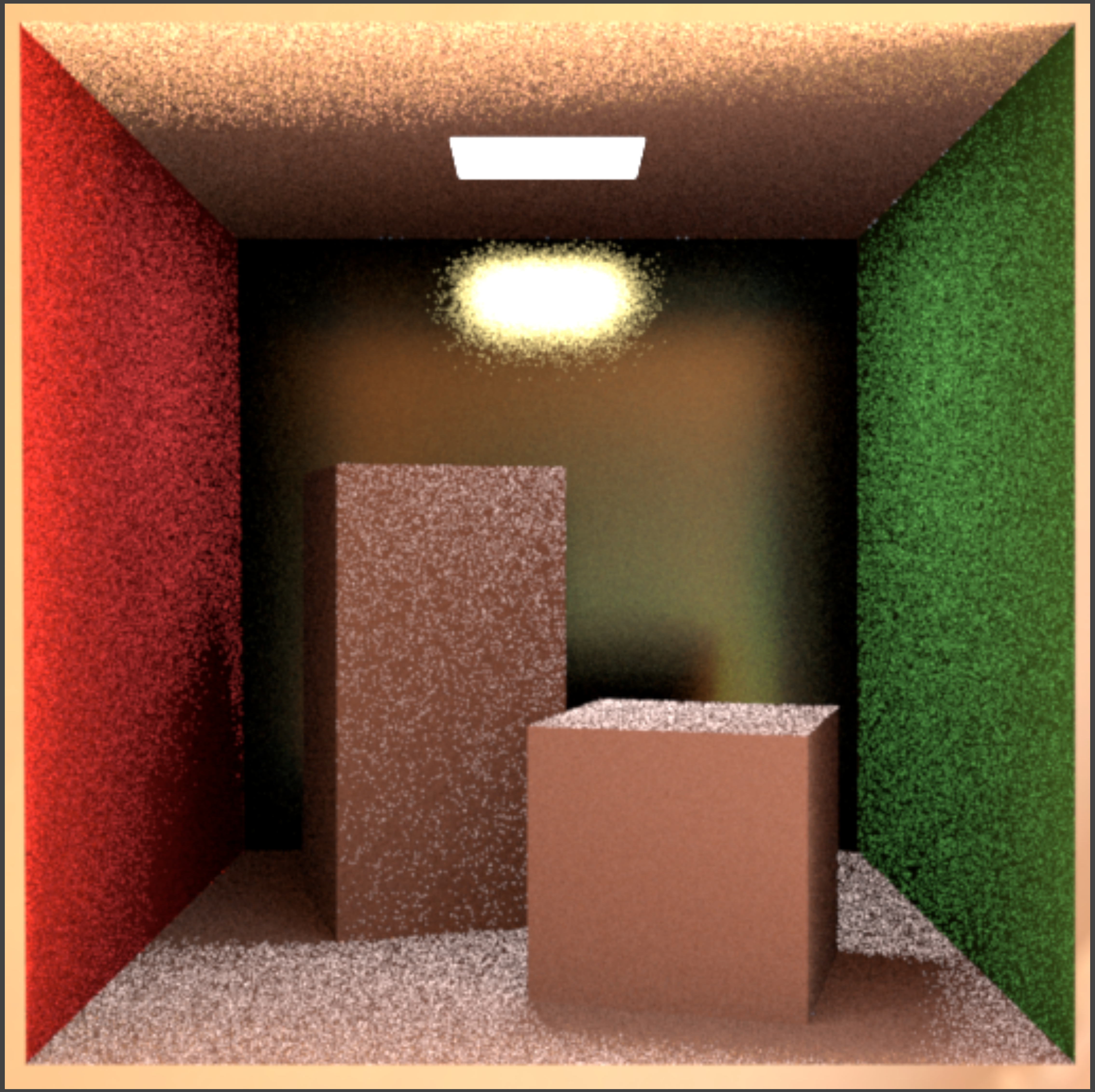
sampling the BRDF



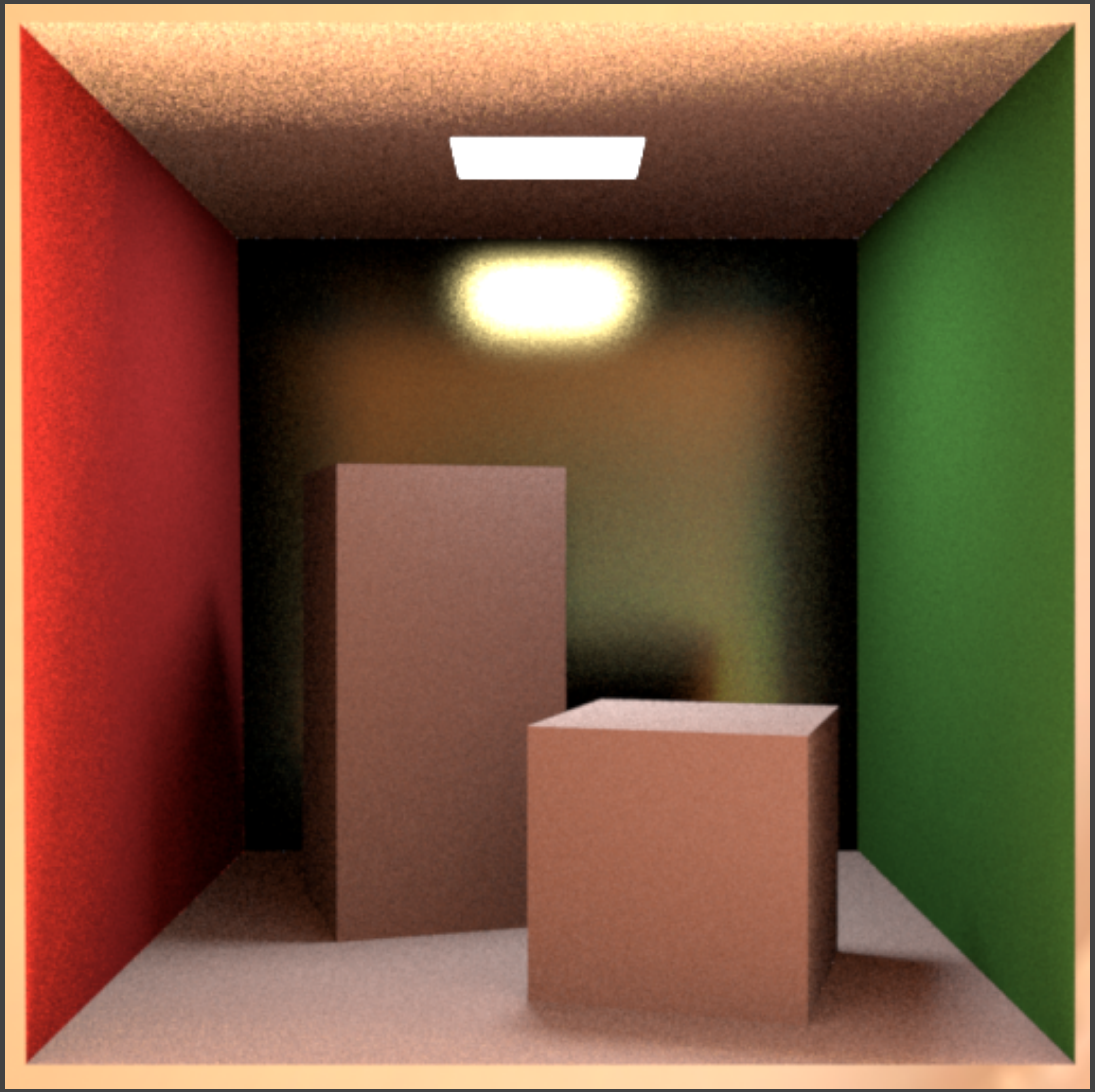
sampling the luminaires



sampling the luminaires



sampling the BRDF



combining samples with power heuristic