A stochastic path tracer implementation

Andrés Solís Montero

uOttawa

University of Ottawa

Outline

Camera Model

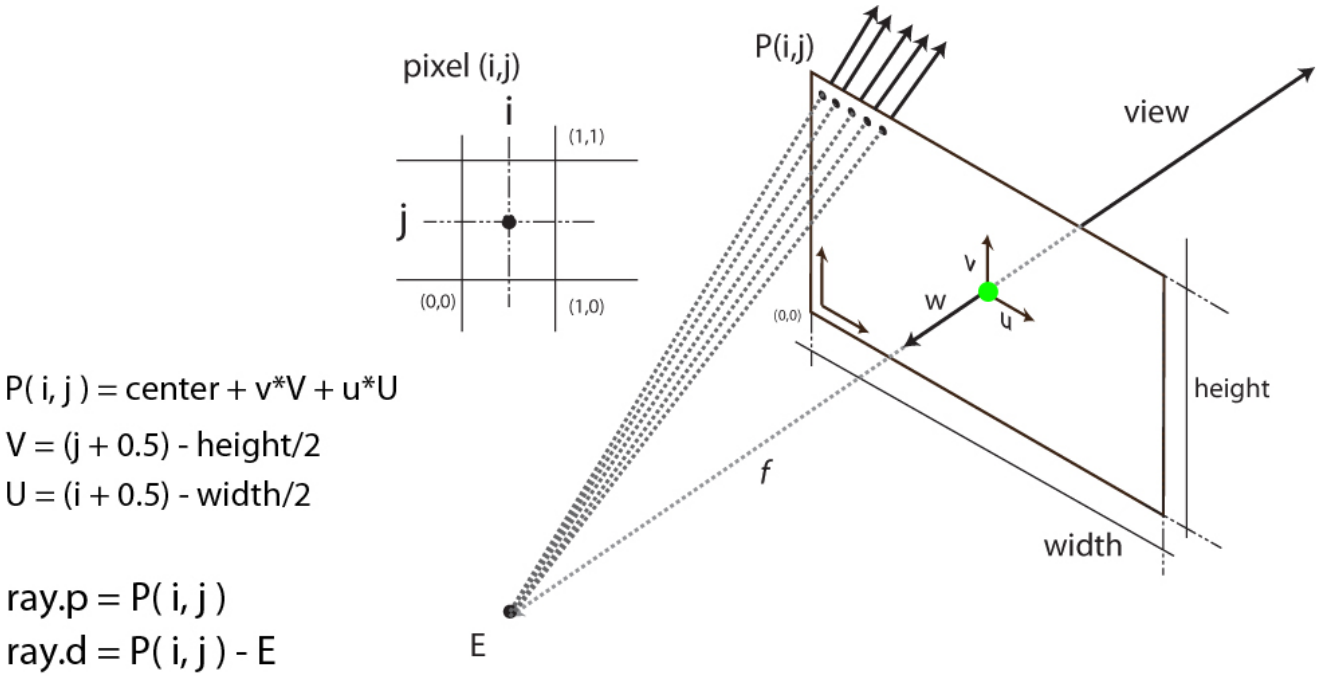
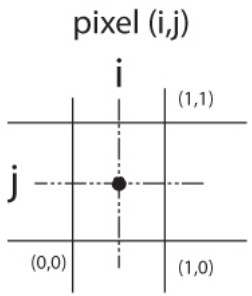
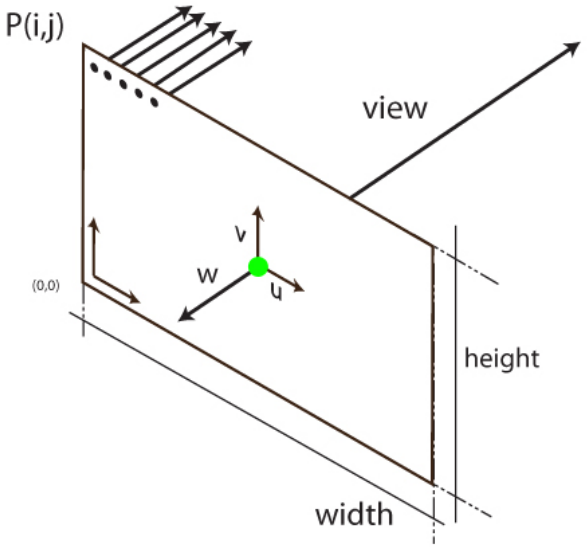


Figure : Perspective projection.

Camera Model





$P(i, j) = center + v*V + u*U$

$V = (j + 0.5) - height/2$

$U = (i + 0.5) - width/2$

$ray.p = P(i, j)$

$ray.d = -1 * w$

Figure : Parallel projection. It is used if the focus length is 0.

Camera Model - Examples

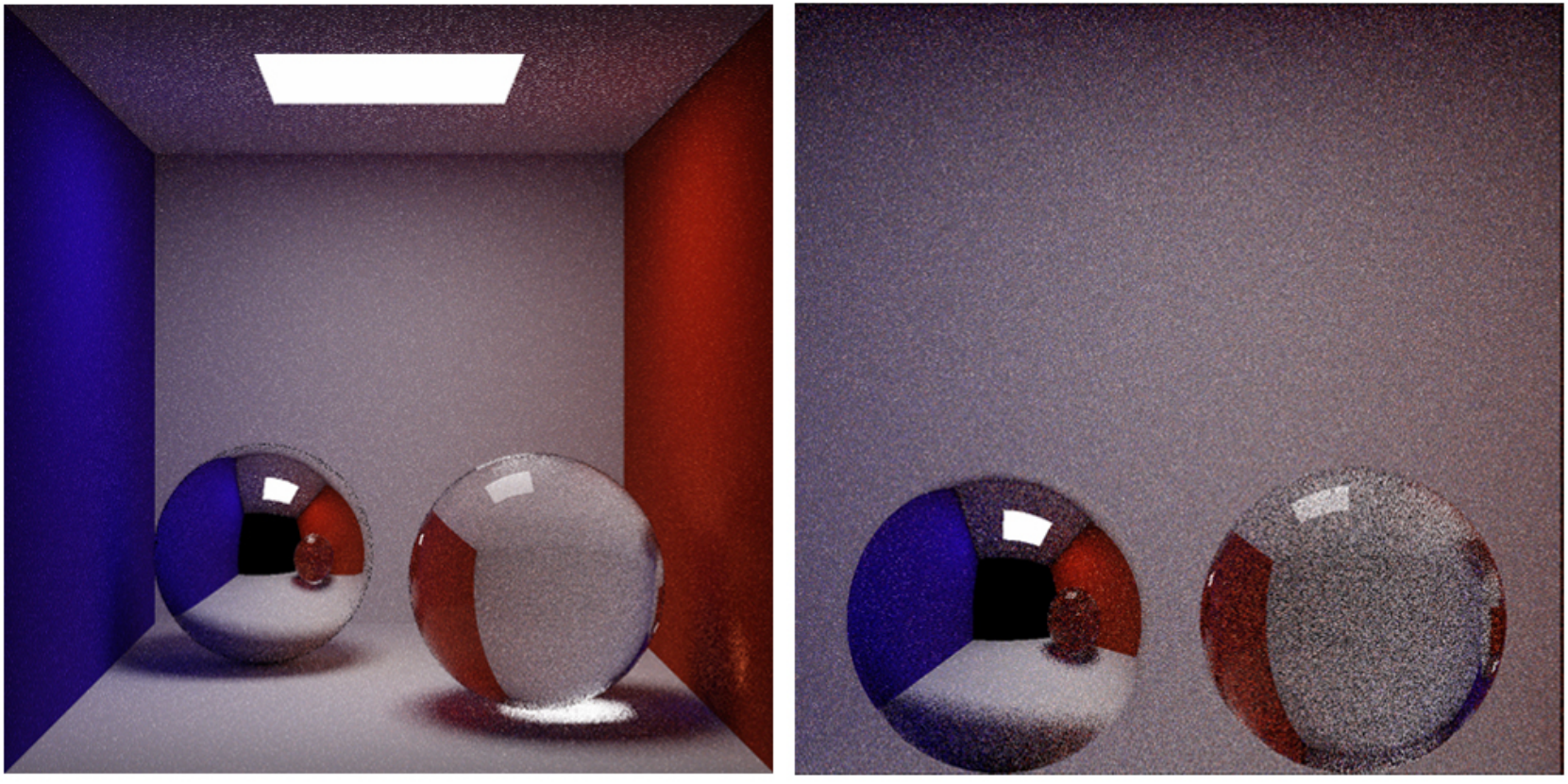


Figure : Perspective and Parallel projection examples.

Camera Model - Examples

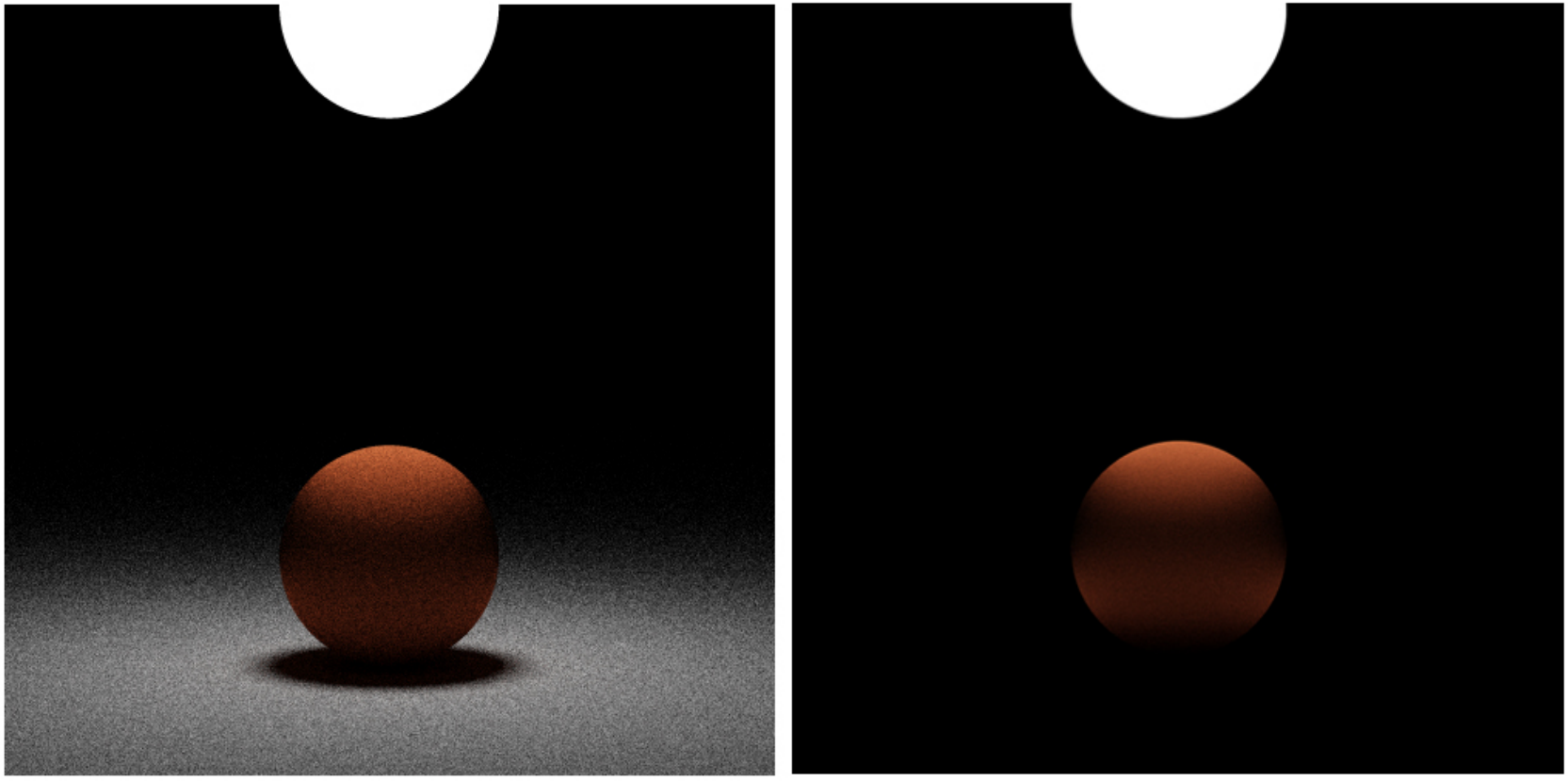


Figure : Perspective and Parallel projection examples.

Camera Model - Examples

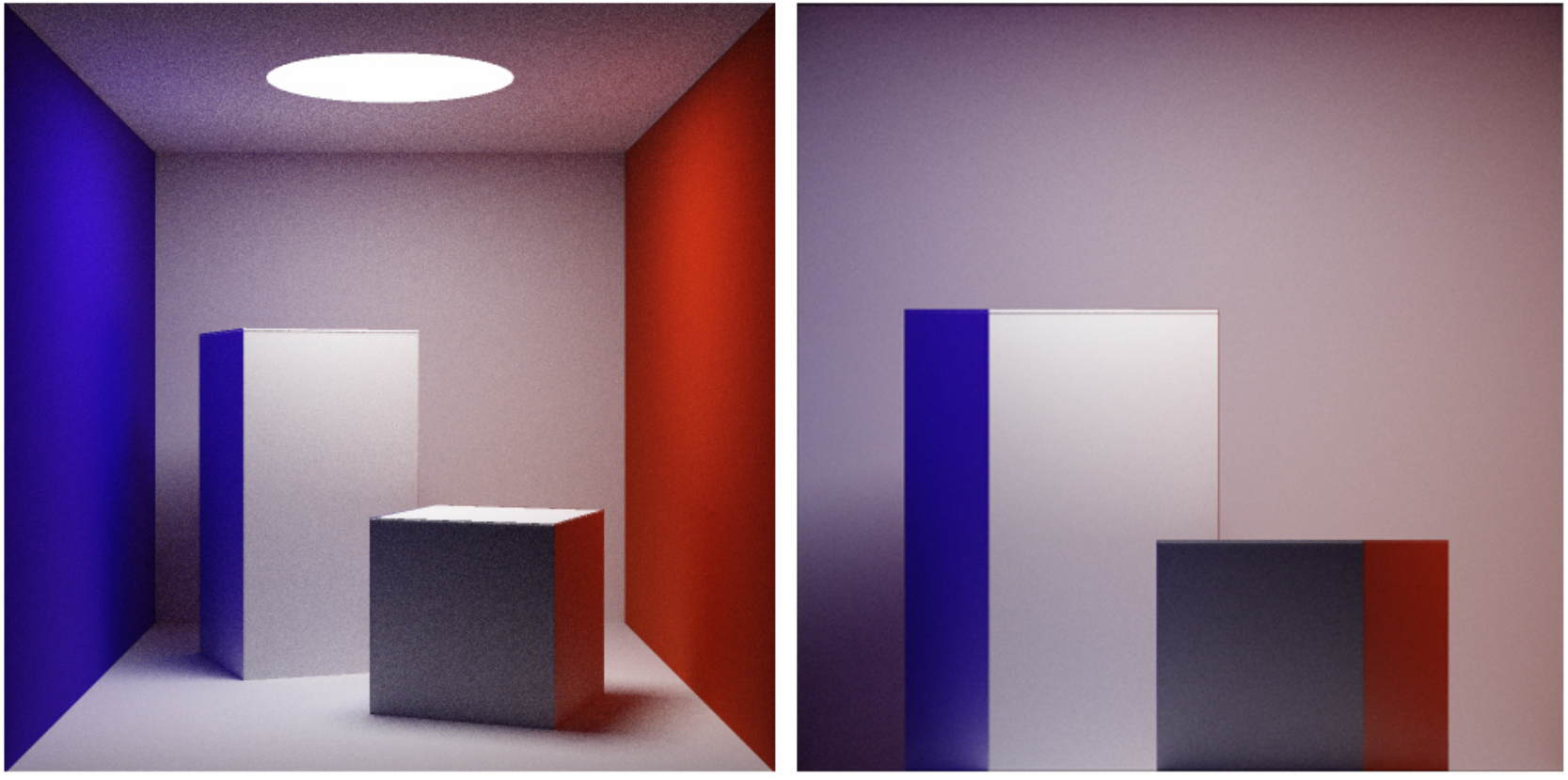


Figure : Perspective and Parallel projection examples.

Luminaires

Light emitting objects

- Only diffuse luminaires were implemented.
- All luminaries have an area (A) and generate points on the surface with density probability p=1/A.
- Only object luminaries emitted radiance $L_e > 0$.
- The light emission value is a RGB vector (affects all wavelengths).

Luminaires - Examples

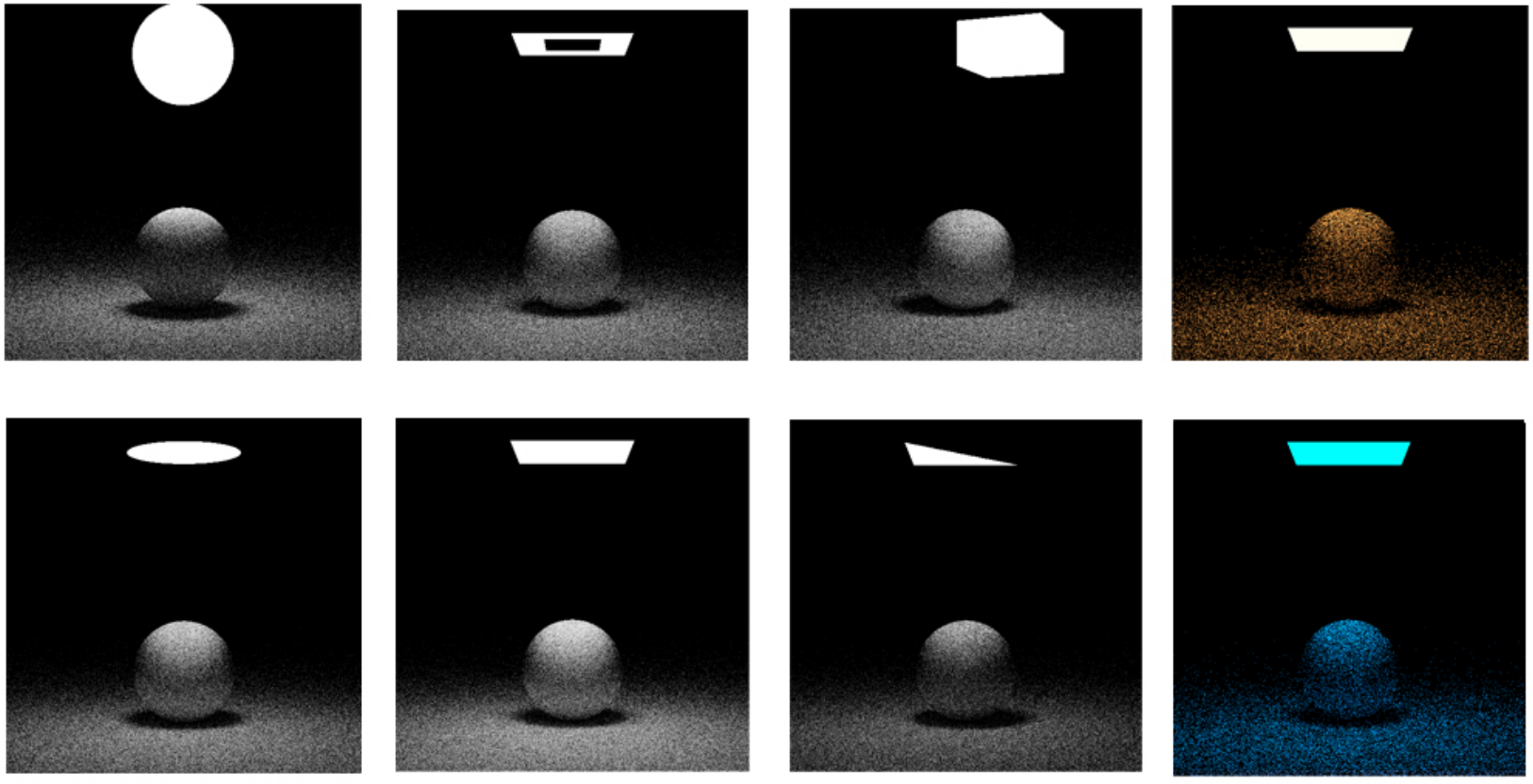


Figure : Object luminaires and different emitted radiance.

The rendering equation

Approximation with Monte Carlo integration

$$L_s(k_0) \approx \frac{\rho(k_i,k_o)L_f(k_i)cos(\theta_i)}{p(k_i)}$$

- L_s is surface outgoing radiance at a given point
- L_f is the incoming radiance at the point.
- k_i is randomly choosen with density p.
- Using $p(k_i) = \frac{\rho(k_i,k_o)cos(\theta_i)}{R(k_o)}$
- R is the directional hemispherical reflectance.

Radiance of a Surface

$$L_s(k_0) \approx R(k_0)L_f(k_i)$$

The rendering equation

Radiance seen along a ray

$$L = L_{e1} + R_1(L_{e2} + R_2(L_{e3} + R_3(...)))$$

- L_{ei} is the emitted radiance at x_i
- R_i is the reflectance.
- use of Russian Roulette to stop the recursion.

```
RGB radiance (ray r)
    if r hits at x then
      generate direction b  (According to surface)
      ray rn = x + tb
      return Le(x) + R(x)*radiance(rn)
    else
      return background(r)
```

The rendering equation

```
for each pixel(i,j):
  RGB r = (0,0,0).
    for s = 0; s < spp (samples per point); s ++:
      r +=  radiance (ray r)/spp ;
```

Indirect Illumination - Examples

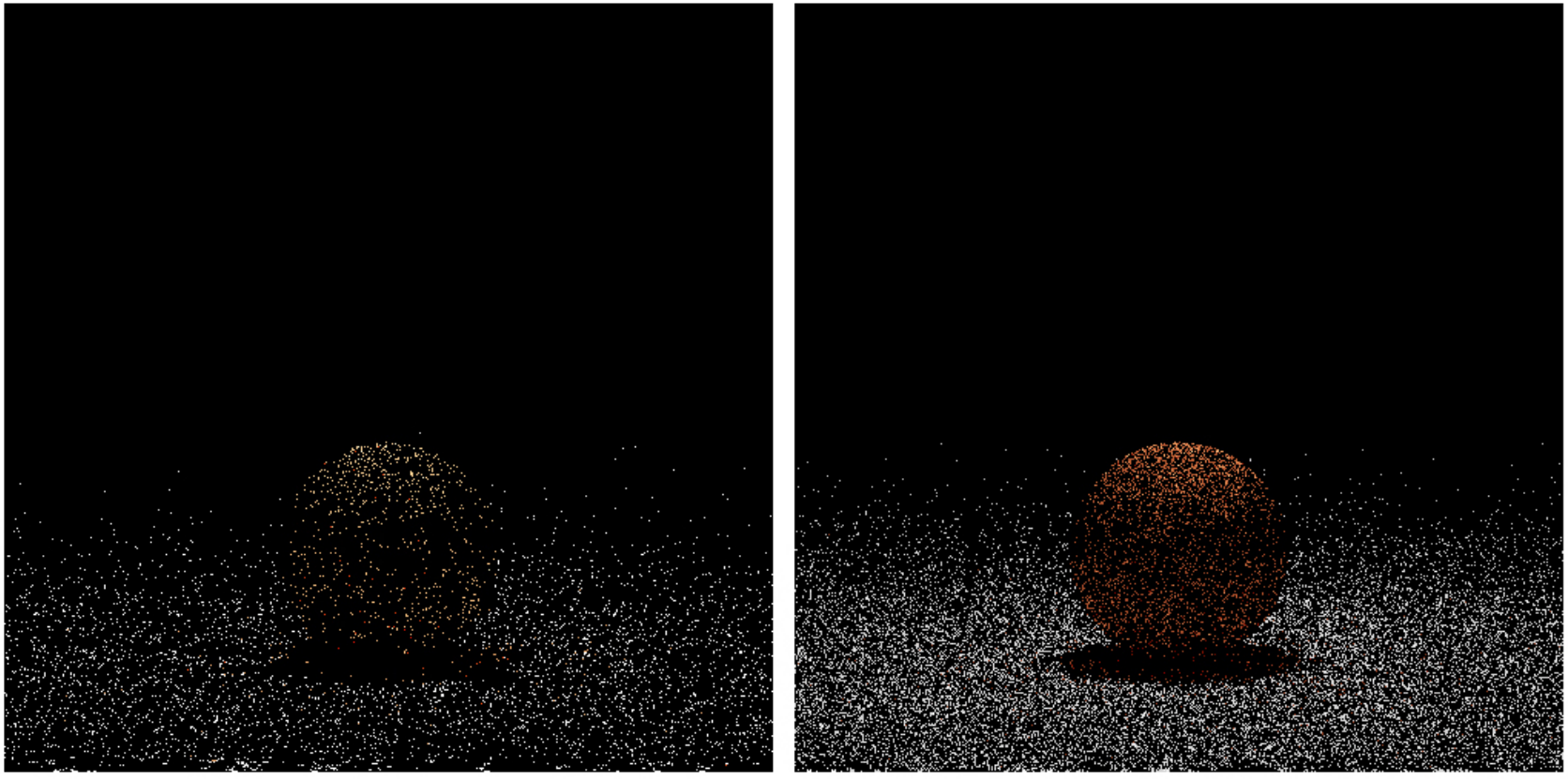


Figure : Lambertian surfaces (left: 5 spp. right: 10 spp).

Indirect Illumination - Examples

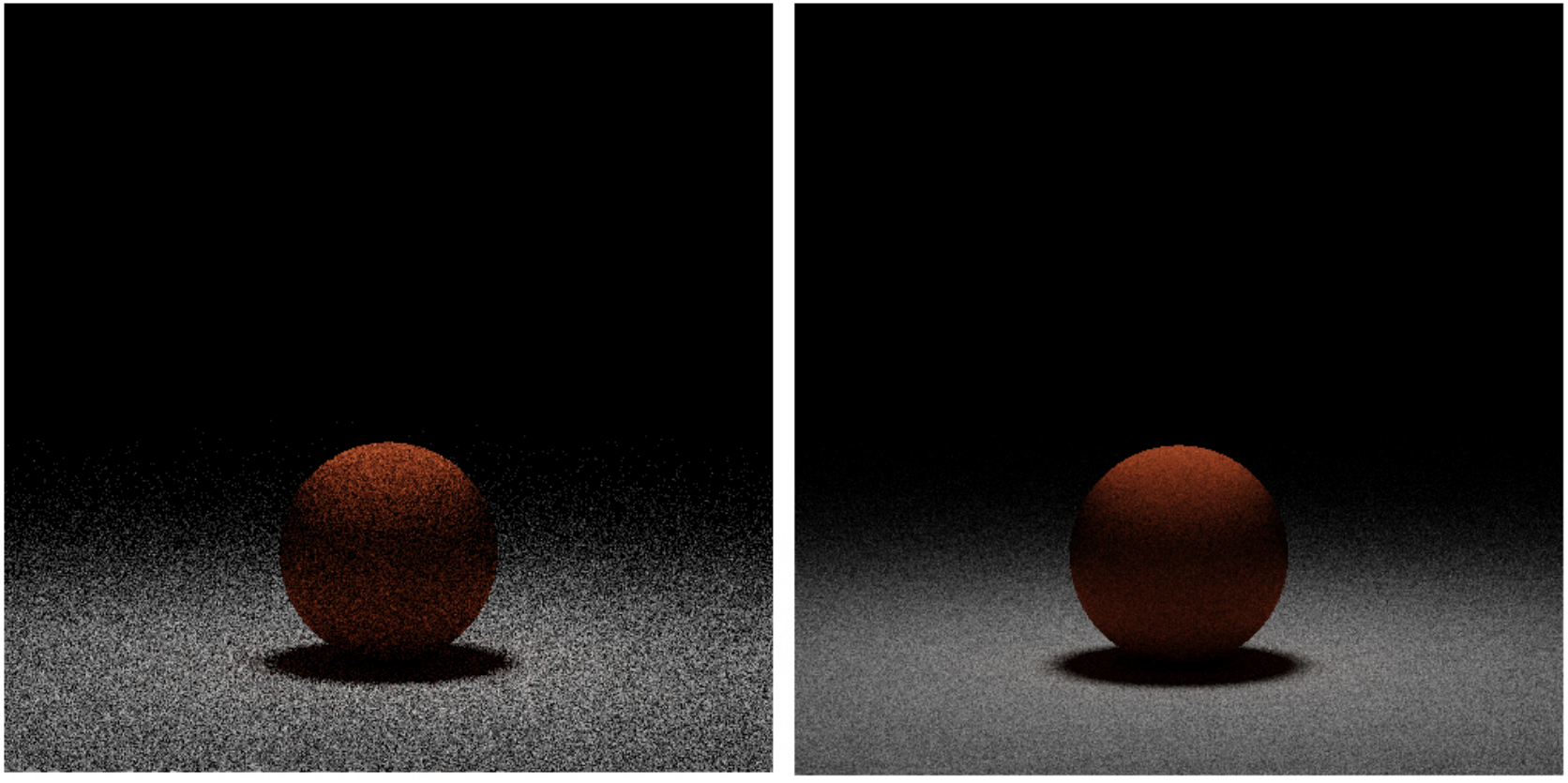


Figure : Lambertian surfaces (left: 200 spp. right: 1600 spp).

Indirect Illumination - Examples

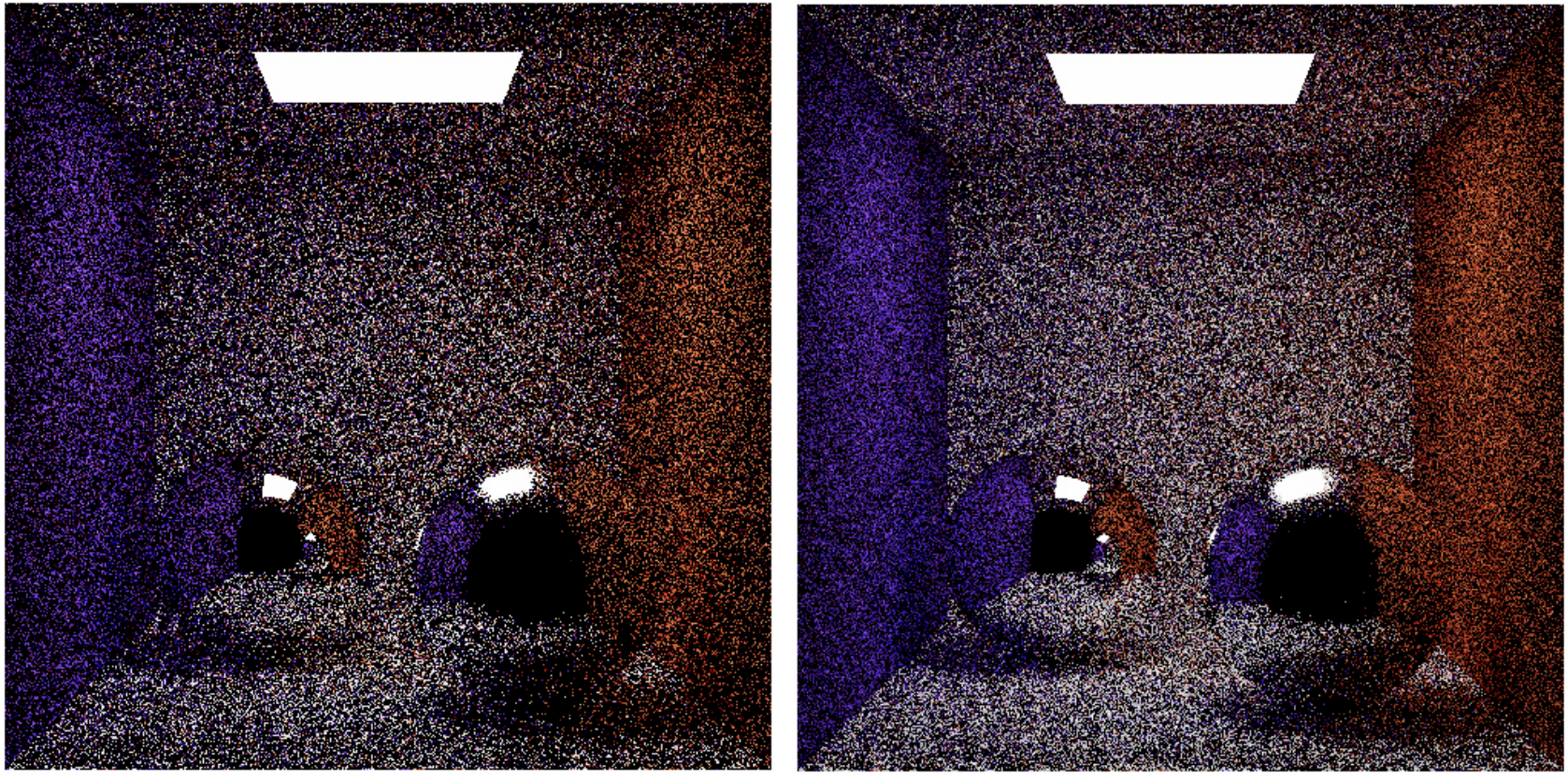


Figure : Lambertian and Specular surfaces (left: 5 spp. right: 10 spp).

Indirect Illumination - Examples

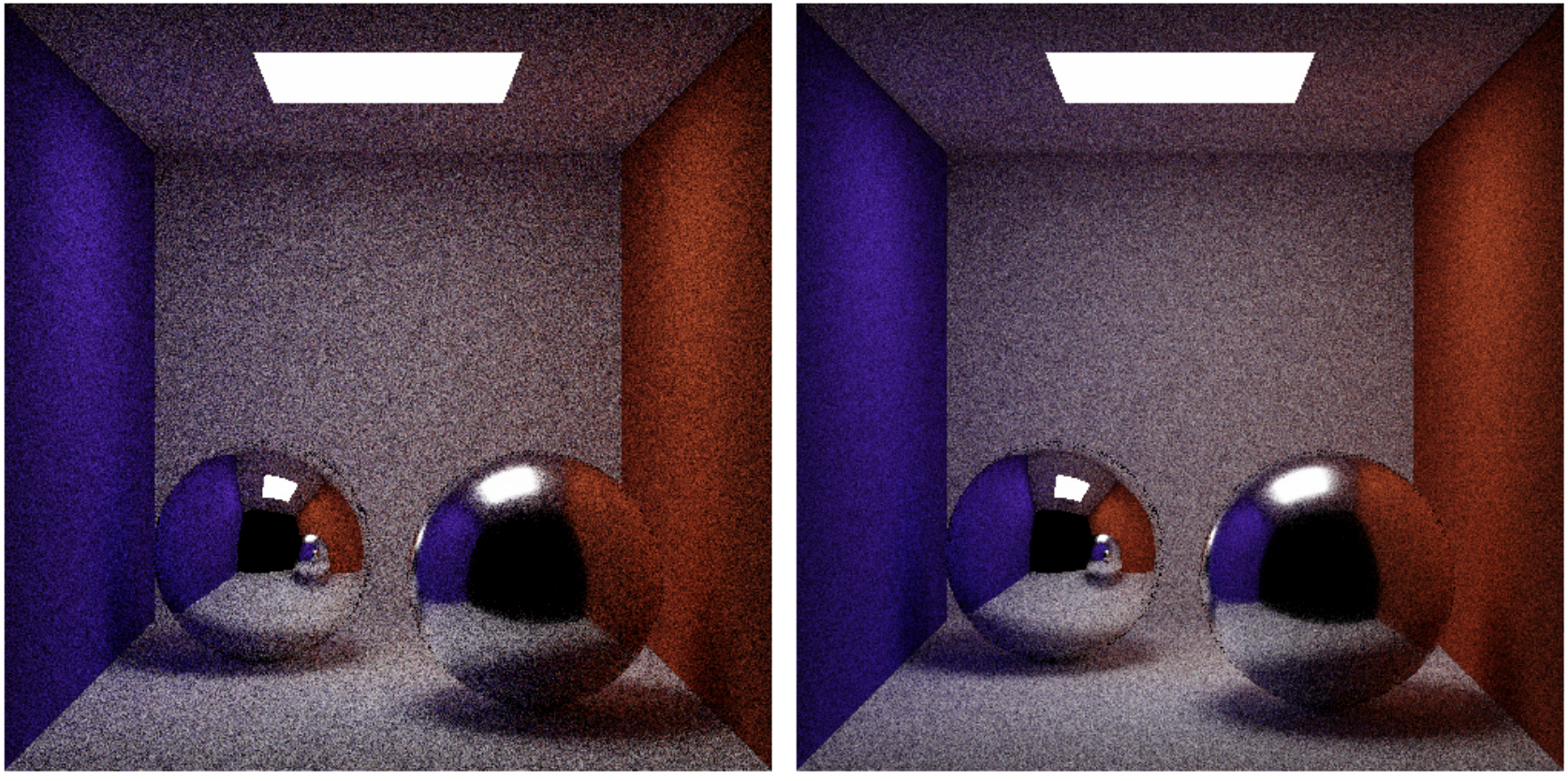


Figure : Lambertian and Specular surfaces (left: 100 spp. right: 200 spp).

Russian Roulette - Examples

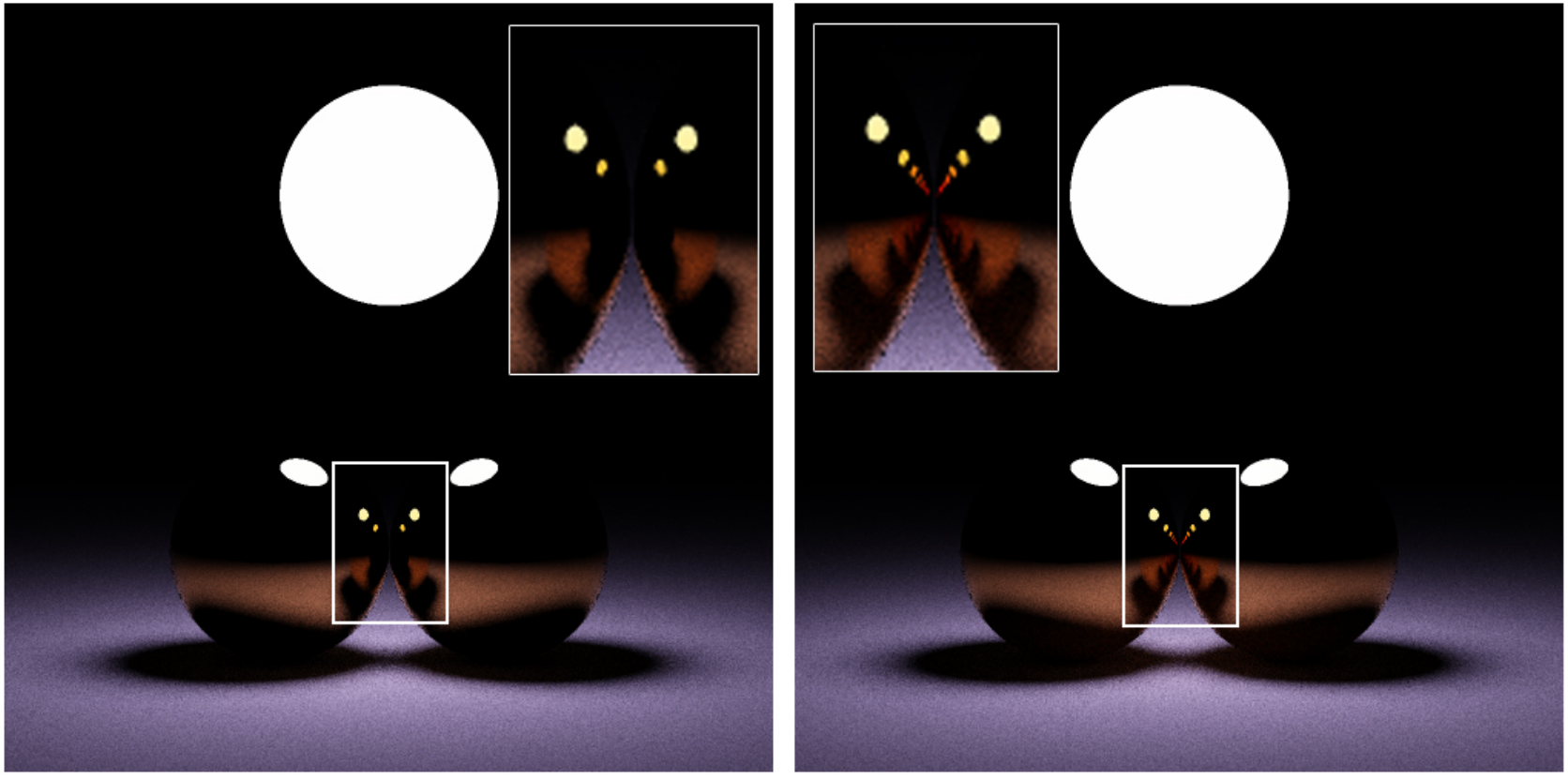


Figure : Specular surfaces at 400spp (left: 2 bounces. right: Russian Roulette).

Russian Roulette - Examples

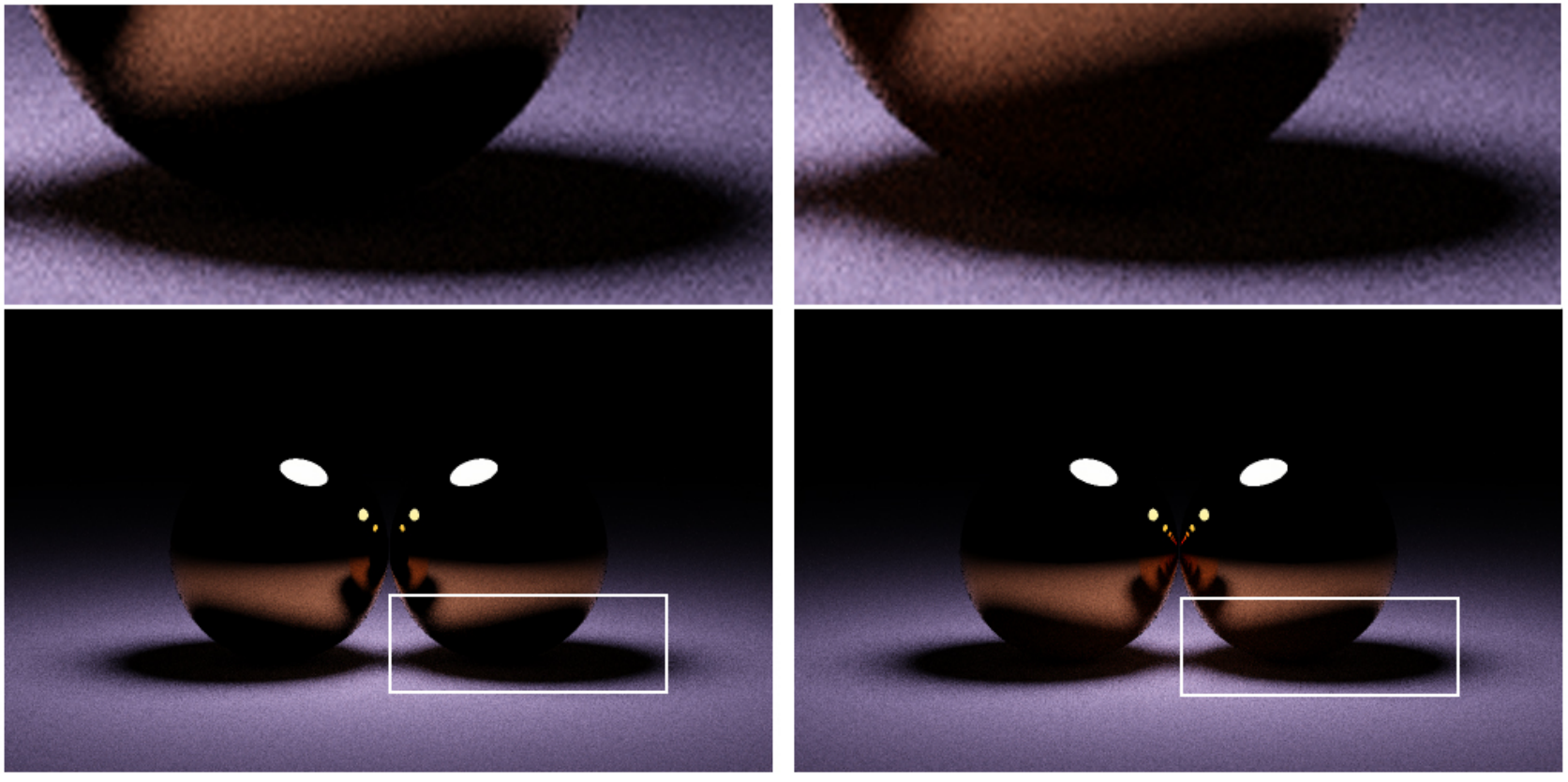
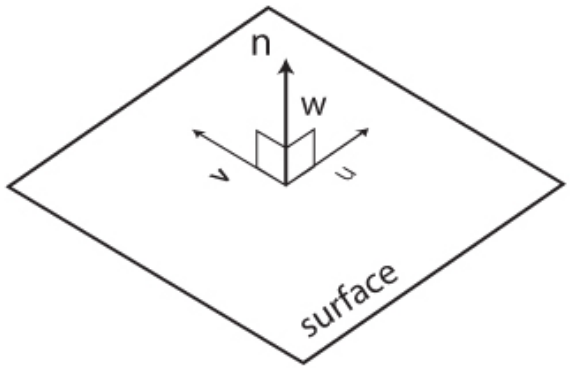


Figure : Specular surfaces at 400spp (left: 2 bounces. right: Russian Roulette).

Creating ONB from outward normal vector

Creating an orthonormal base **uvw**, from an outward normal vector. (**u,v,w** are unit length vectors, mutually perpendicular,and right handed $(u \times v = w)$).



ONB from w

$w = n$

$m = (0, 1, 0)$

$n = (1, 0, 0)$

$u = wxm$

if $||u||^2 < \epsilon$

$u = wxn$

$v = wxn$

Lambertian Surfaces

Cosine density relative to the surface normal at hit point

- Using: $\rho(\theta, \phi) = \frac{1}{\pi} cos(\theta)$.
- Generate: r_1 and r_2 canonical random numbers.
- $cos(\theta) = \sqrt{1 - r_1}$.
- $\phi = 2\pi r_2$.
- Generate orthonormal basis **uvw** for the surface where w points in the direction of the outward surface.

Direction of the ray

$$r_n.b = cos(\phi)sin(\theta)u + sin(\phi)sin(\theta)v + cos(\theta)w.$$

Lambertian Surfaces - Examples

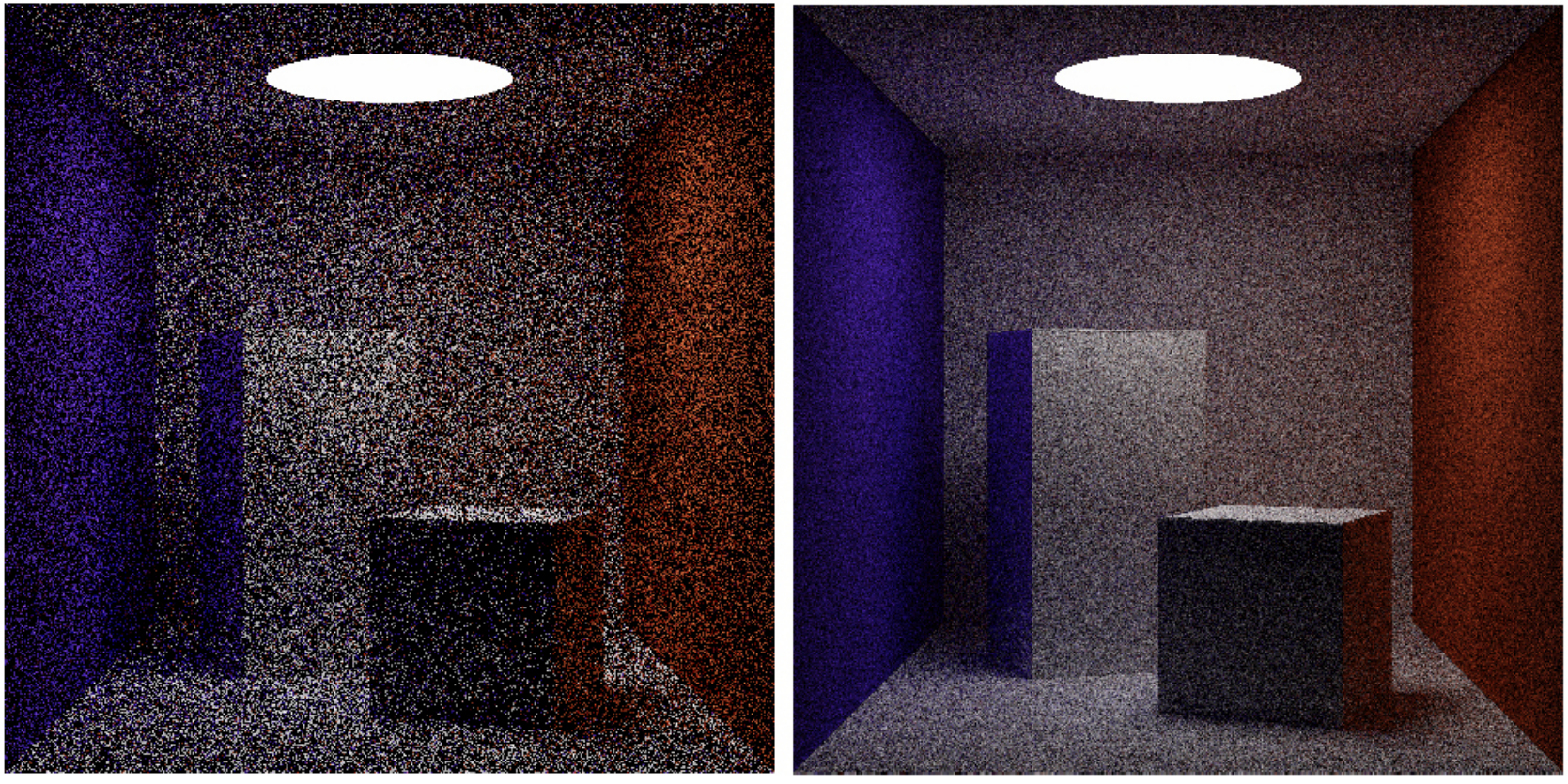


Figure : Lambertian surfaces (left: 10 spp. right: 100 spp).

Lambertian Surfaces - Examples

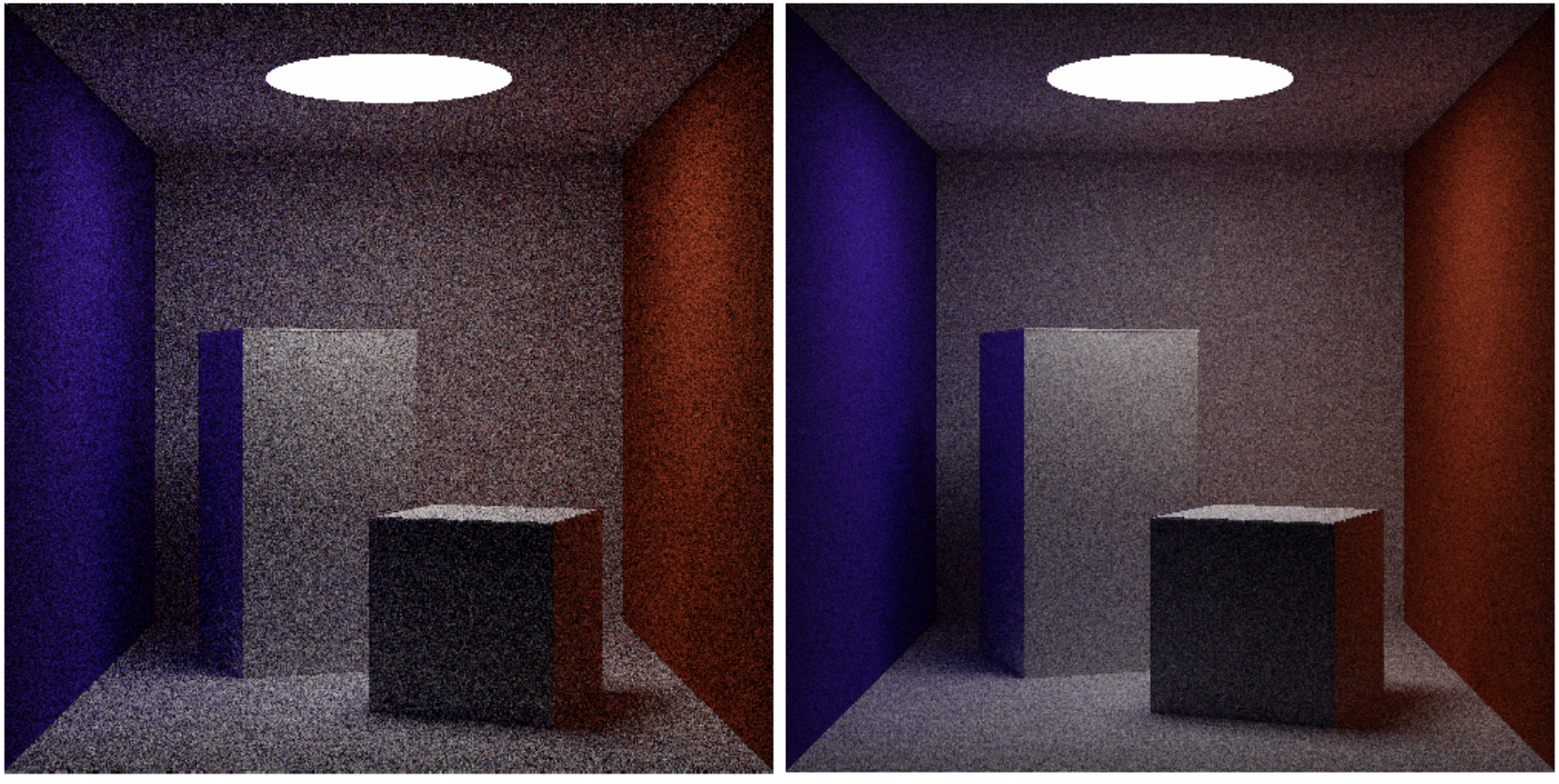


Figure : Lambertian surfaces (left: 100 spp. right: 400 spp).

Specular Surfaces

Computing the ray direction in specular surfaces (mirrors)

- Law of reflection .
- Fresnel equation (Schlick Approx.)
  $R(\theta) \approx R_0 + (1 - R_0)(1 - cos(\theta))^5$.

Direction of the ray

$$r_n.b = d - 2(d \bullet n)n.$$

Specular Surfaces - Examples

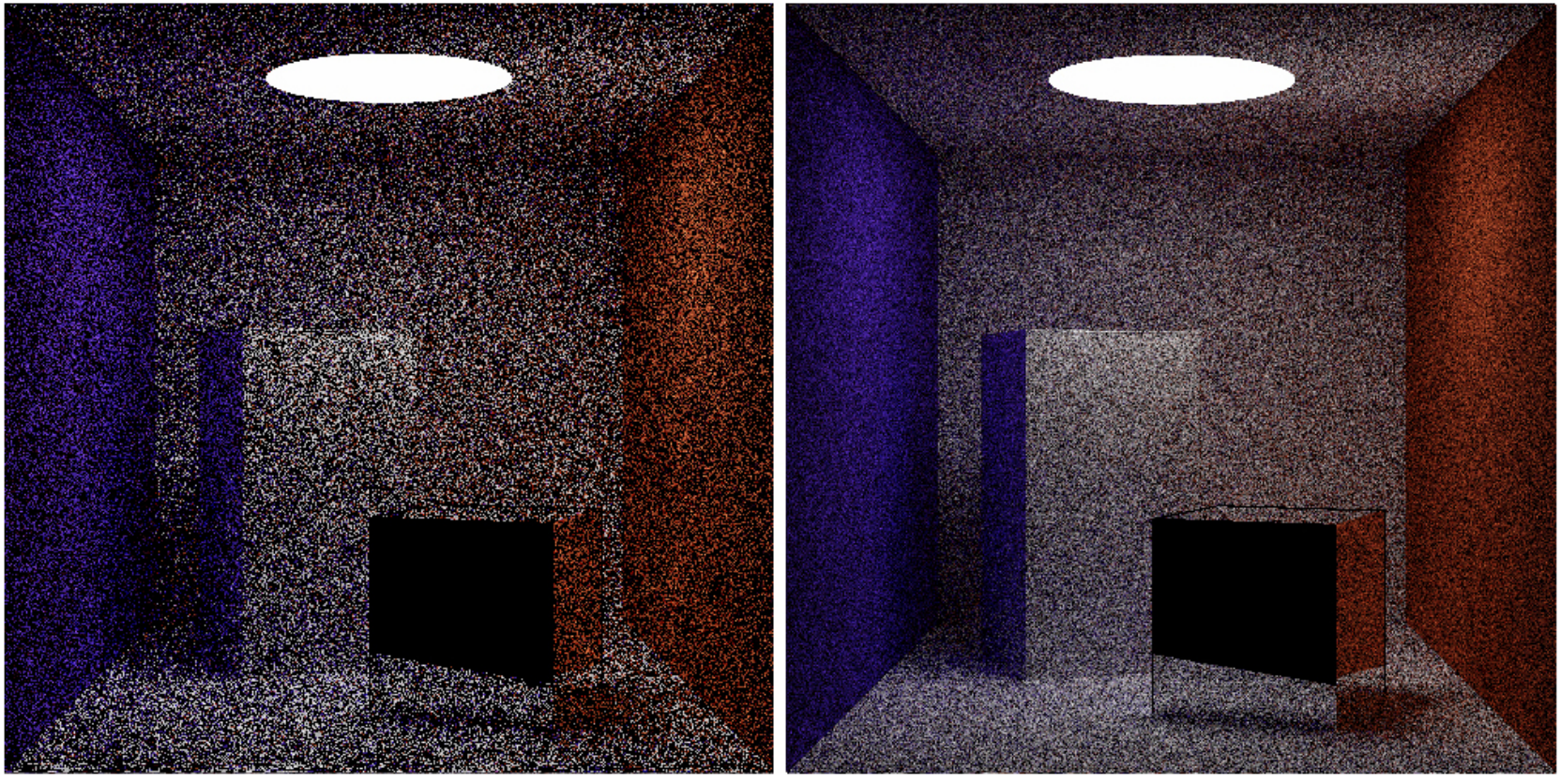


Figure : Specular surfaces (left: 10 spp. right: 100 spp).

Specular Surfaces - Examples

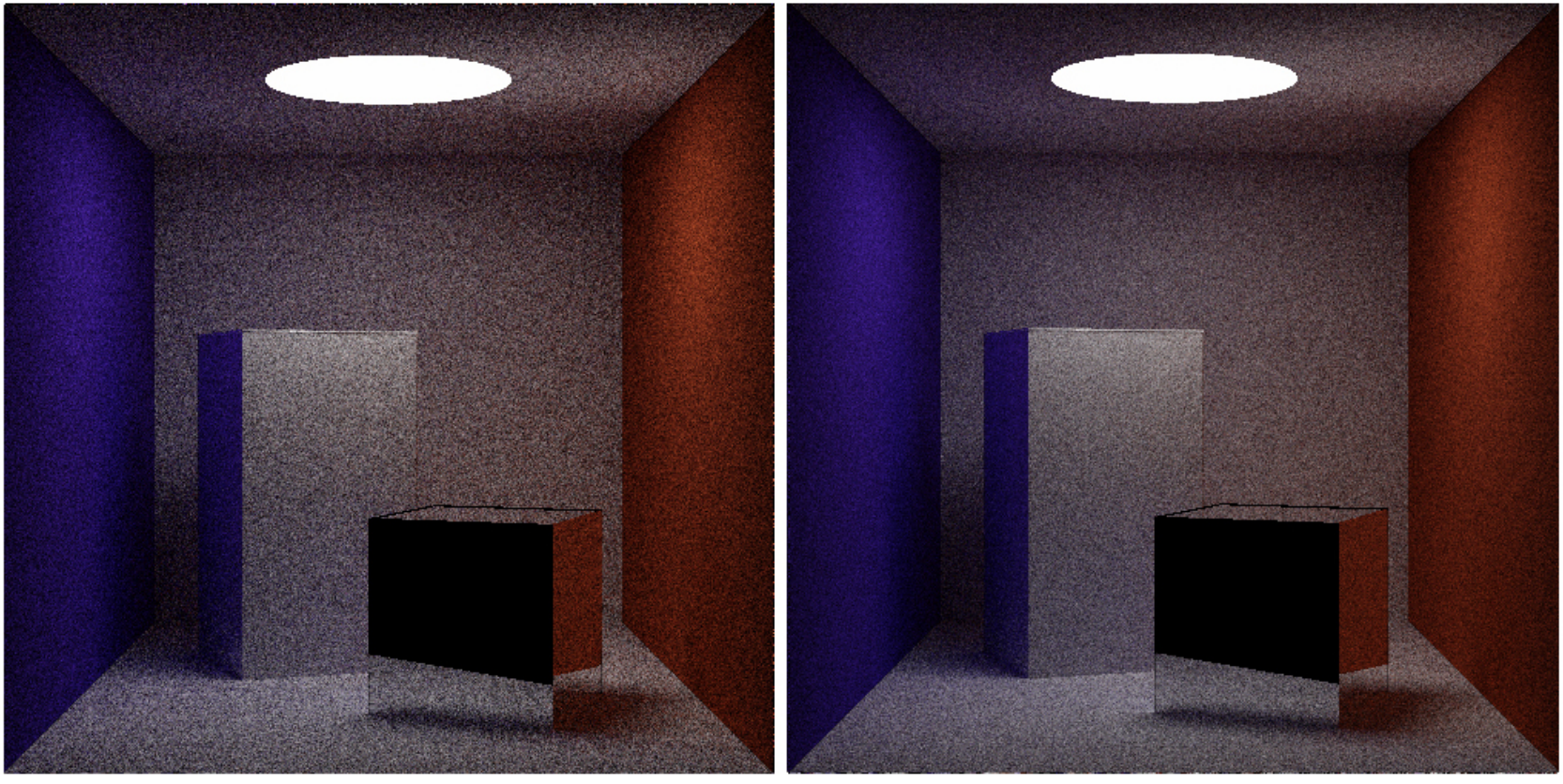


Figure : Specular surfaces (left: 200 spp. right: 400 spp).

Imperfect Specular Surfaces

Cosine density relative to the surface normal at hit point

- Using: $\rho(k) = \frac{n+1}{2\pi}(r \bullet k)^n$. r: mirror direction normalized.
- Generate: r_1 and r_2 canonical random numbers.
- $cos(\theta) = r \bullet k = (1 - r_1)^{\frac{1}{n+1}}$.
- $\phi = 2\pi r_2$.
- Generate orthonormal basis **uvw** for the surface where w points in the direction of the outward surface.

Direction of the ray

$$r_n.b = cos(\phi)sin(\theta)u + sin(\phi)sin(\theta)v + cos(\theta)w.$$

check $cos(\theta) < 0$ regenerate ray above the surface.

Imperfect Specular Surfaces - Examples

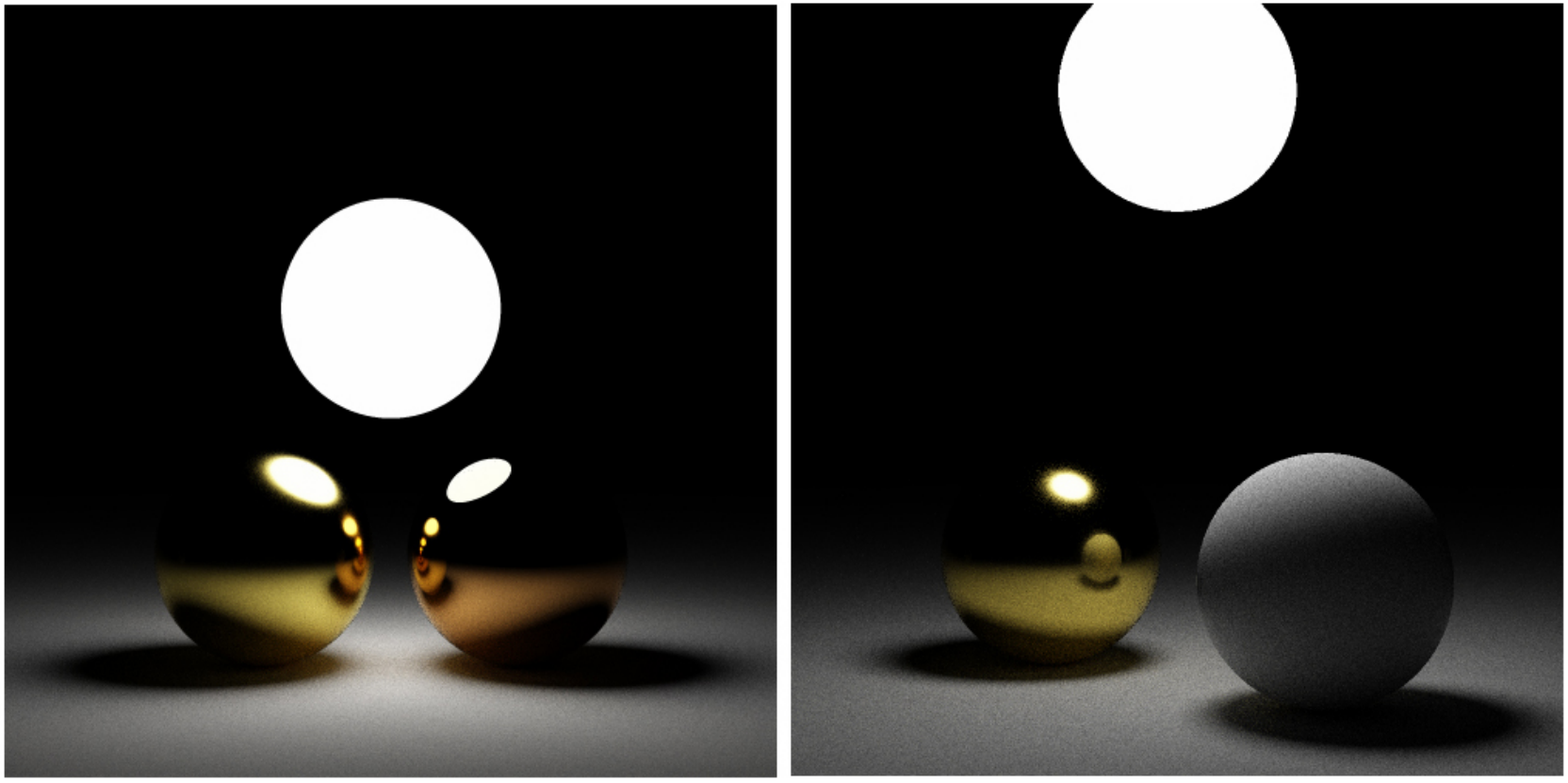


Figure : Imperfect specular gold, specular copper and lambertian surfaces (left: 800 spp. right: 400 spp).

Imperfect Specular Surfaces - Examples

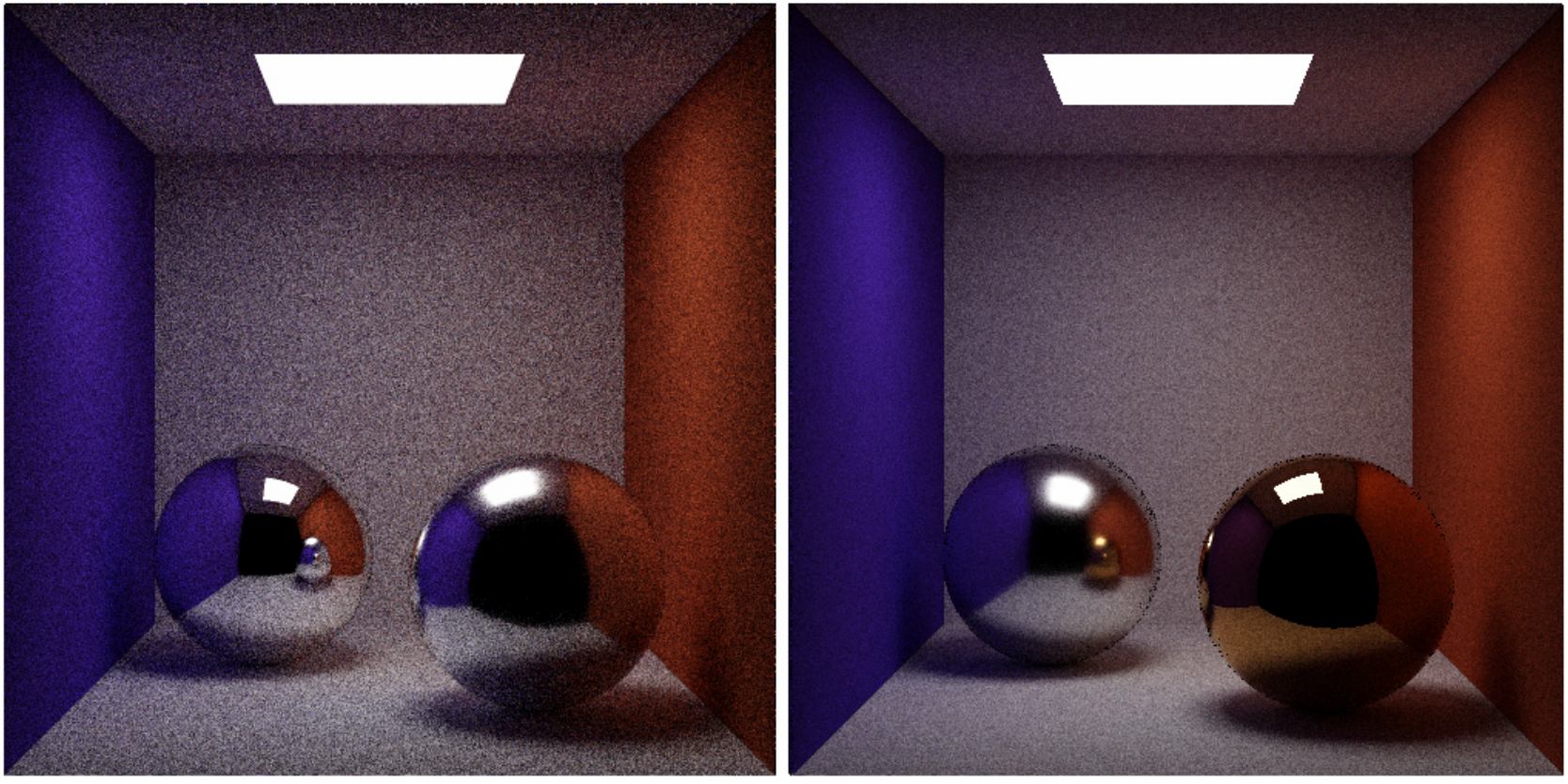


Figure : Imperfect specular and specular surfaces (left: 400 spp. right: 800 spp).

Imperfect Specular Surfaces - Examples

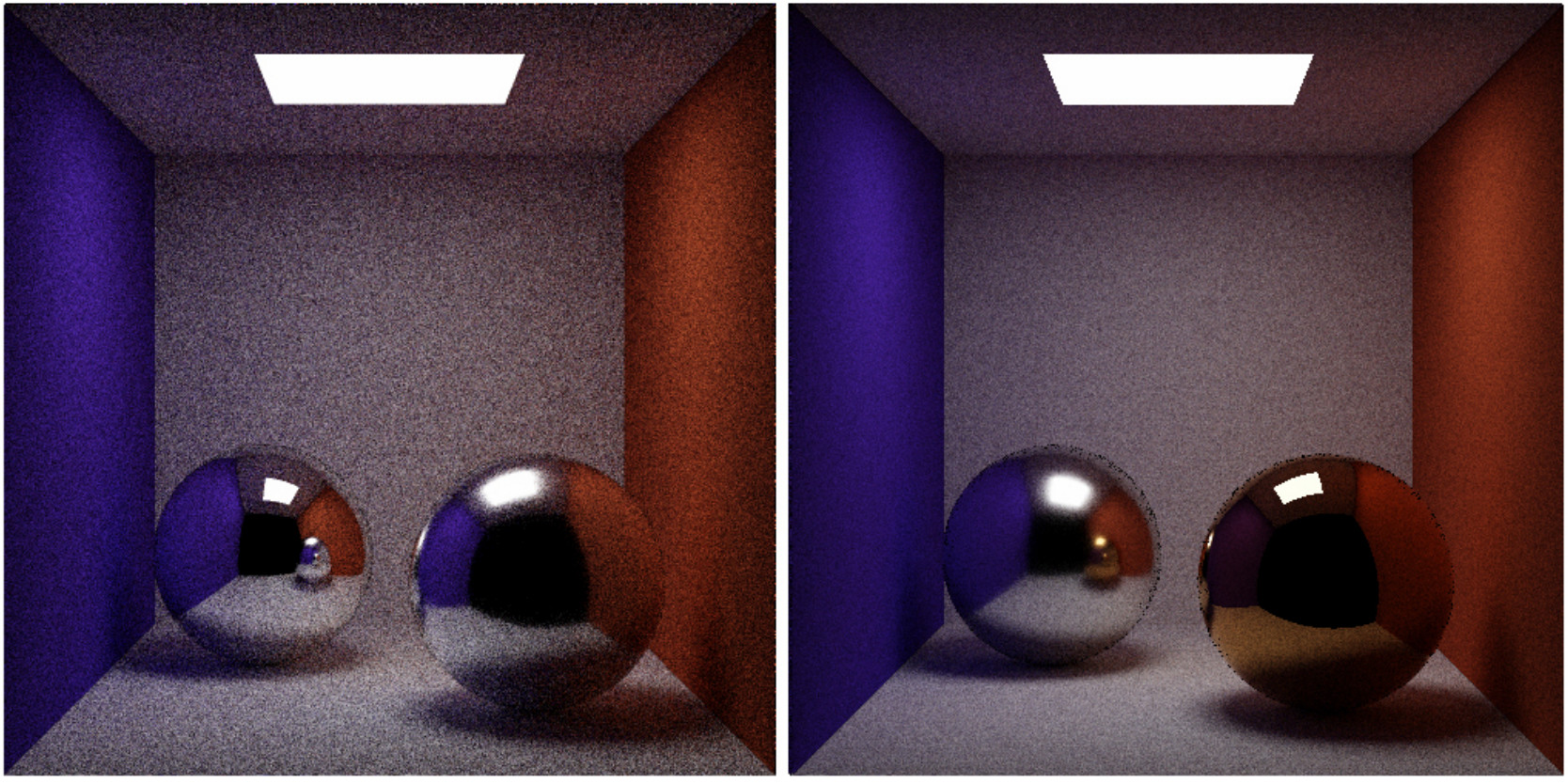


Figure : Imperfect specular and specular surfaces (left: 400 spp. right: 800 spp).

Other Surfaces - Examples

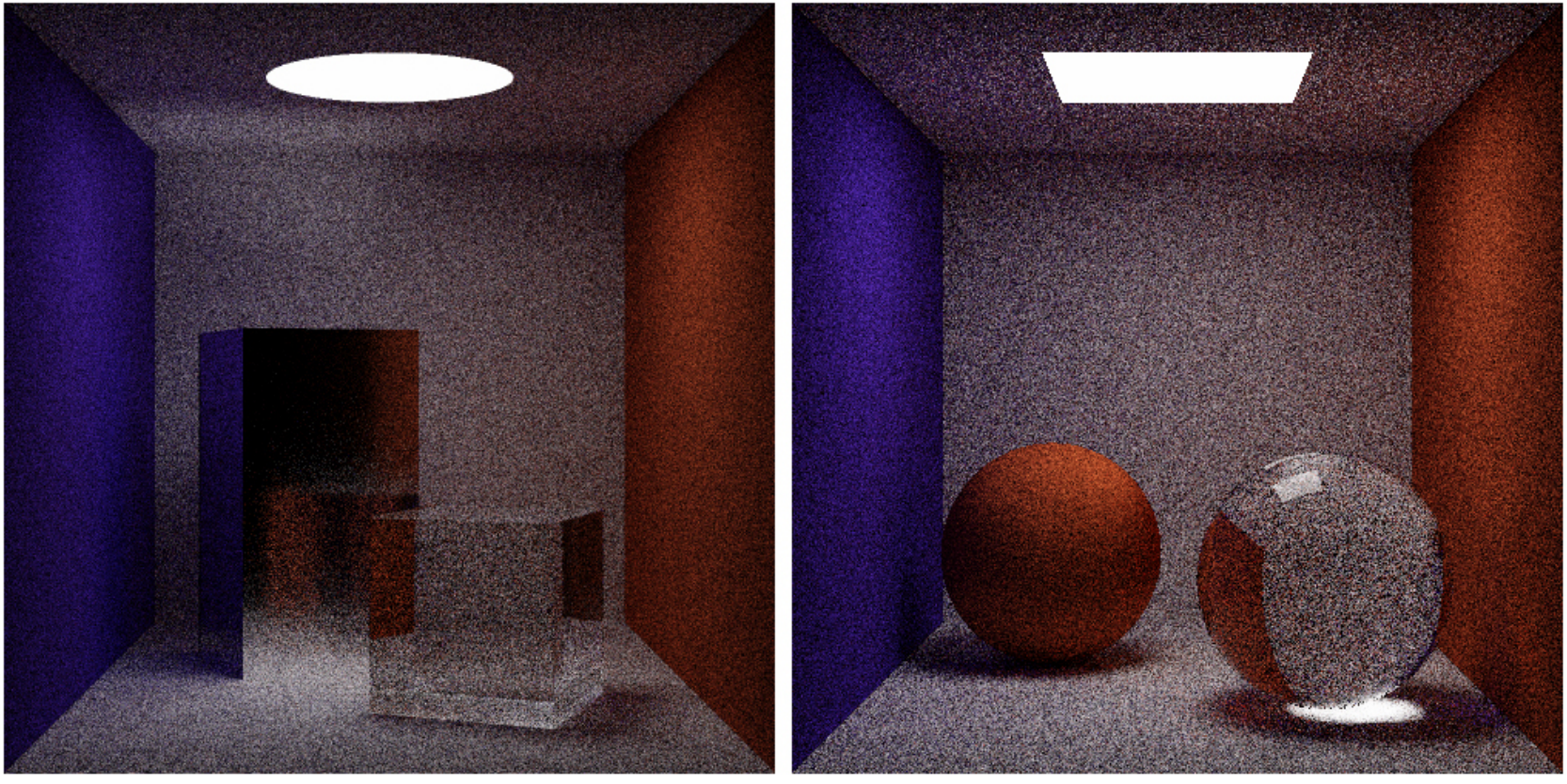


Figure : Smooth Dielectrics (refractive index: 1.5).

Direct Illumination

Approximation with Monte Carlo integration

$$L_s(x, k_0) \approx \frac{\rho(k_i, k_o) L_e(x', x - x') cos(\theta_i) cos(\theta')}{p(x') ||x - x'||^2}$$

Direct Illumination - Examples

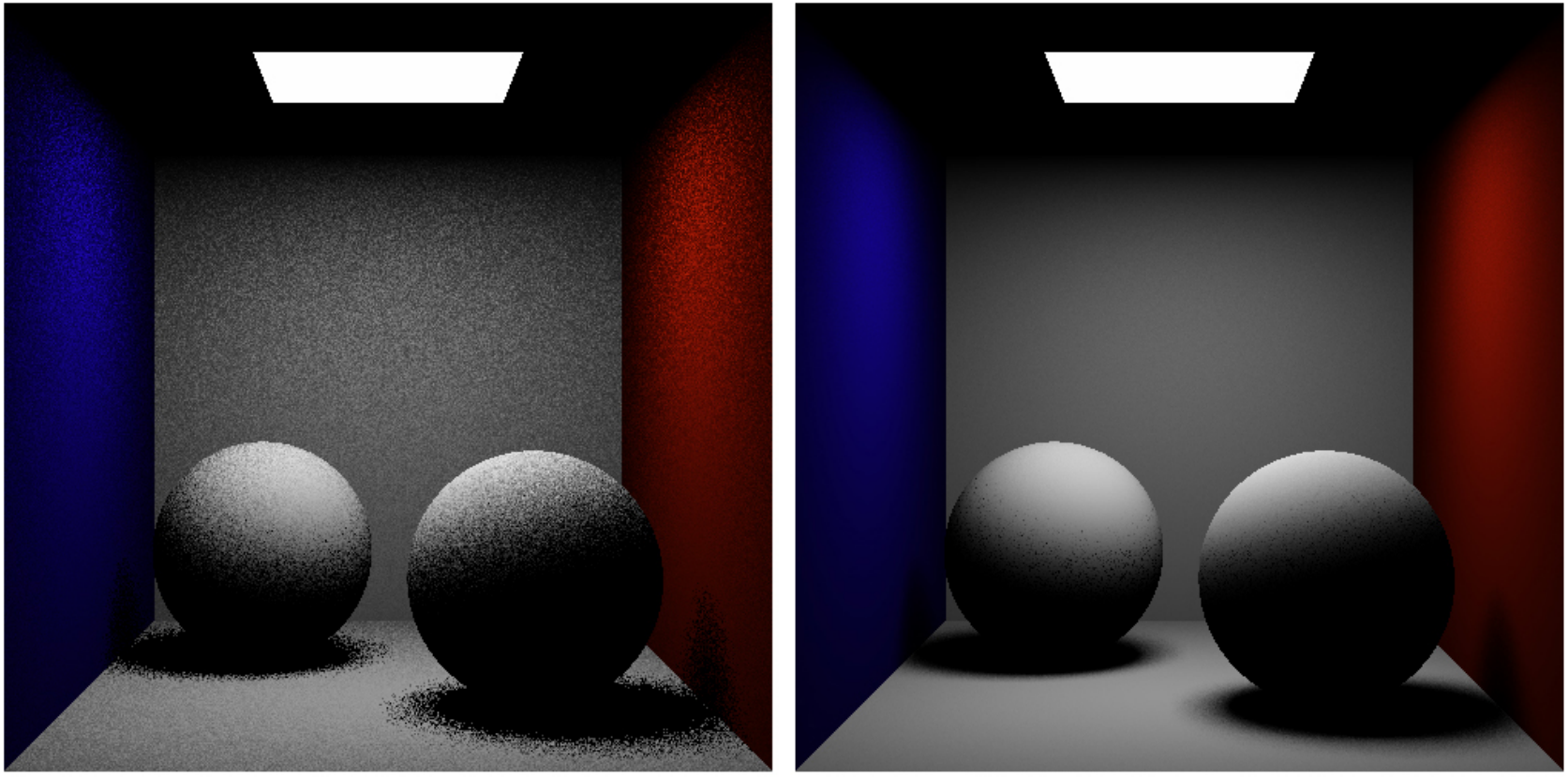


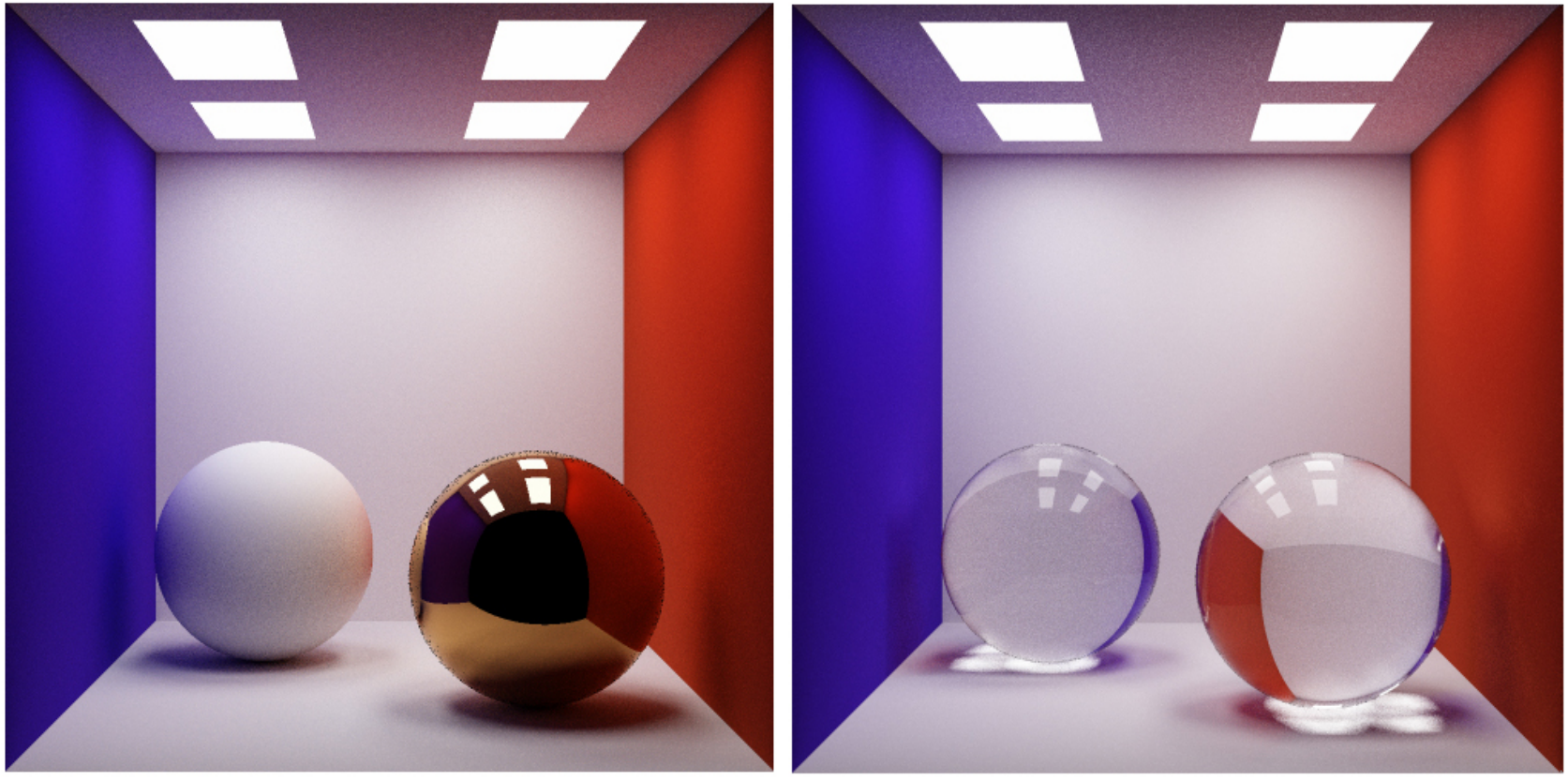
Figure : Lambertian surfaces (Left: 2spp , Right: 200spp).

Multiple Lights

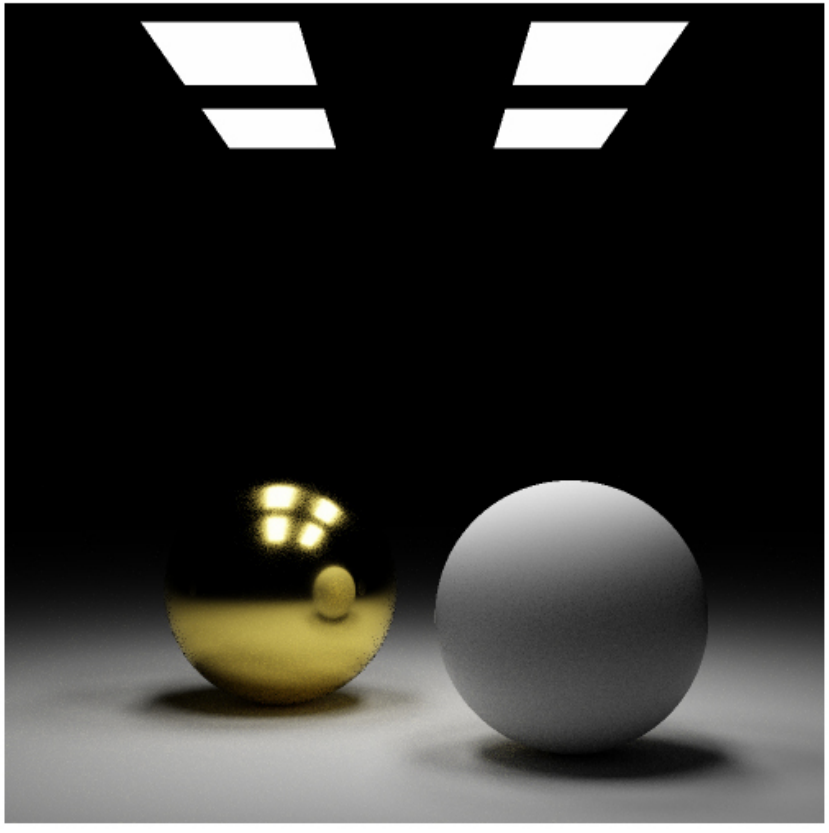
Choosing weights

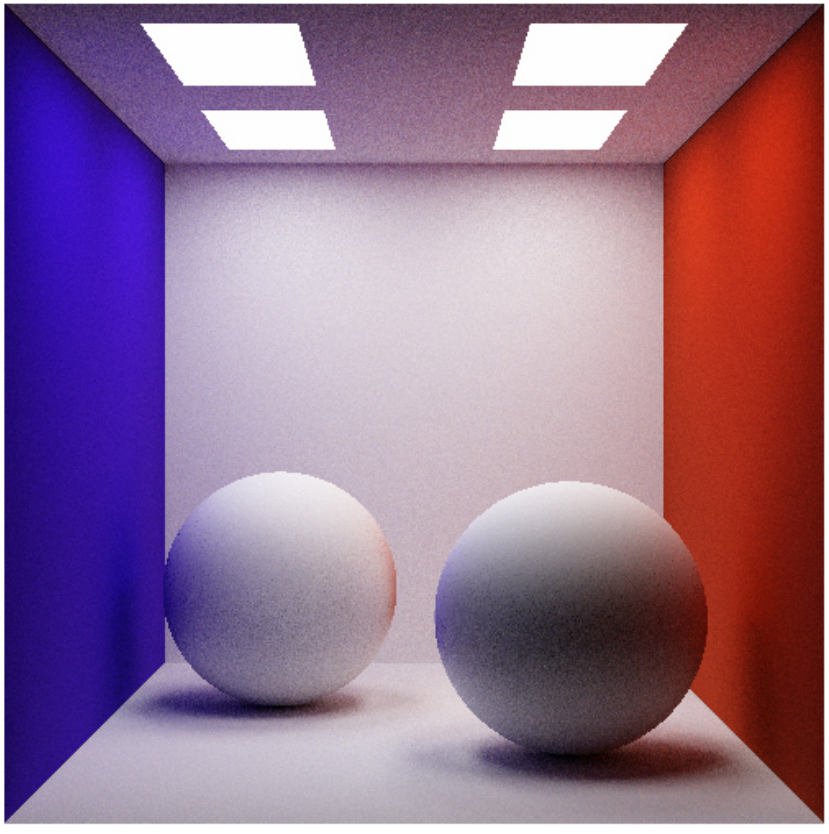
In scenes with multiple lights, I assumed all the lights have the same area and same L_i (emission).
The contribution of each light was $\frac{1}{N_l}$ (N_l is the amount of light objects).

Multiple Lights - Examples



Multiple Lights - Examples





Supersampling

A simple approach to reduce aliasing

Take more samples per pixel.
Split pixel in equally spaced cells (vertically and horizontally).
The pixel radiance is the mean of the values.

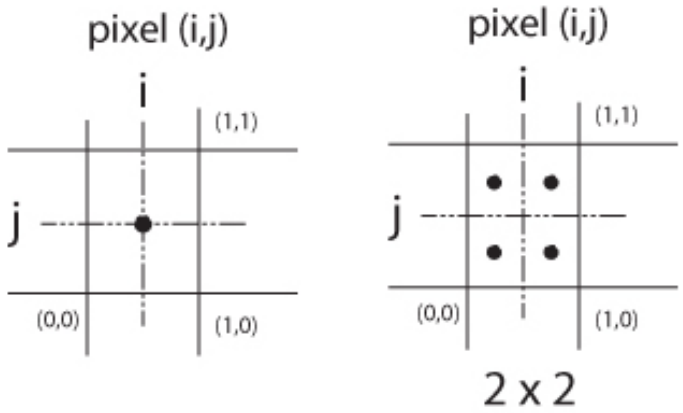


Figure : 4x supersampling.

Antialiasing - Examples

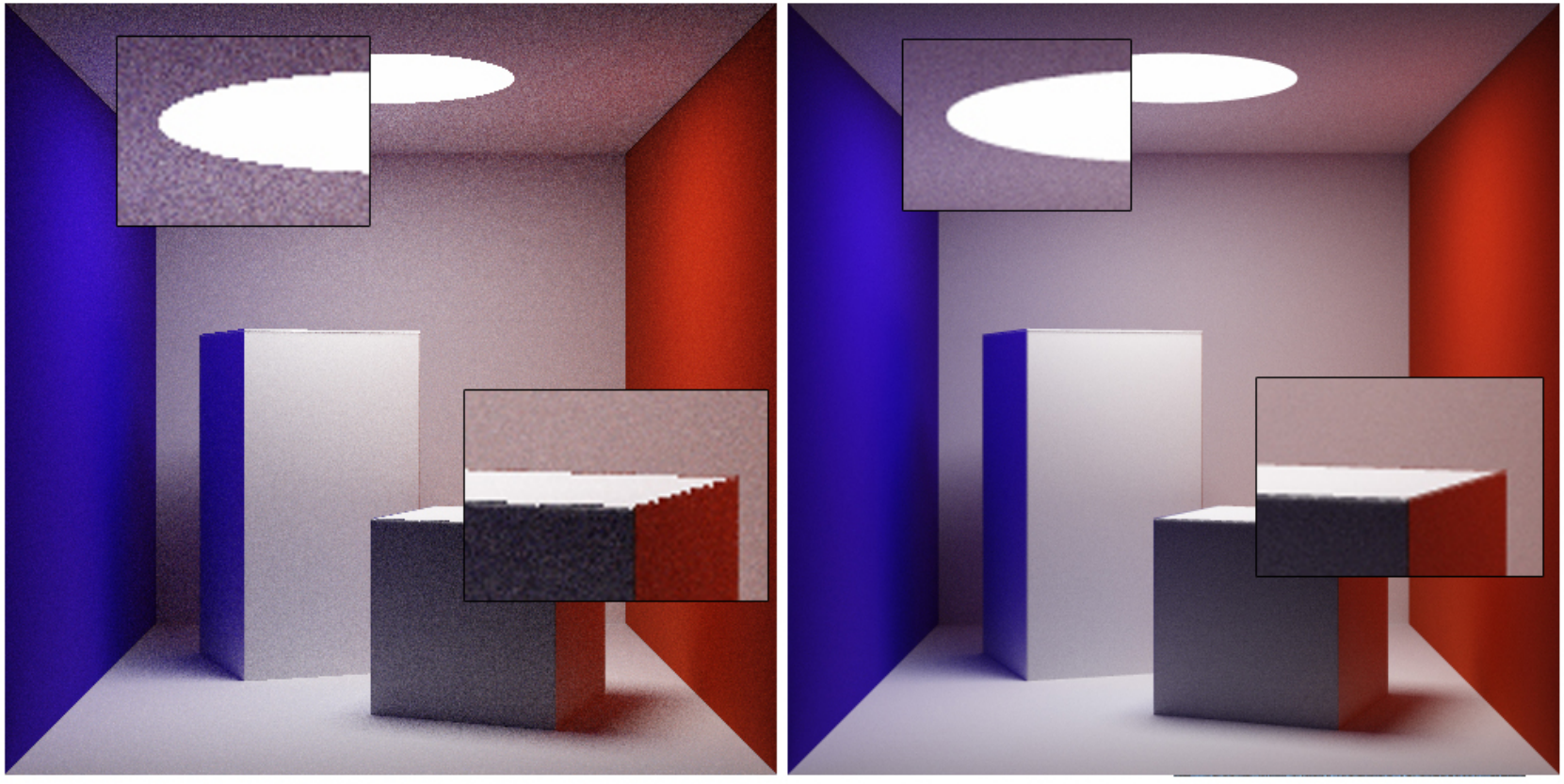


Figure : 16x (4x4) supersampling (Left: 100spp , Right: 50spp).

Antialiasing - Examples

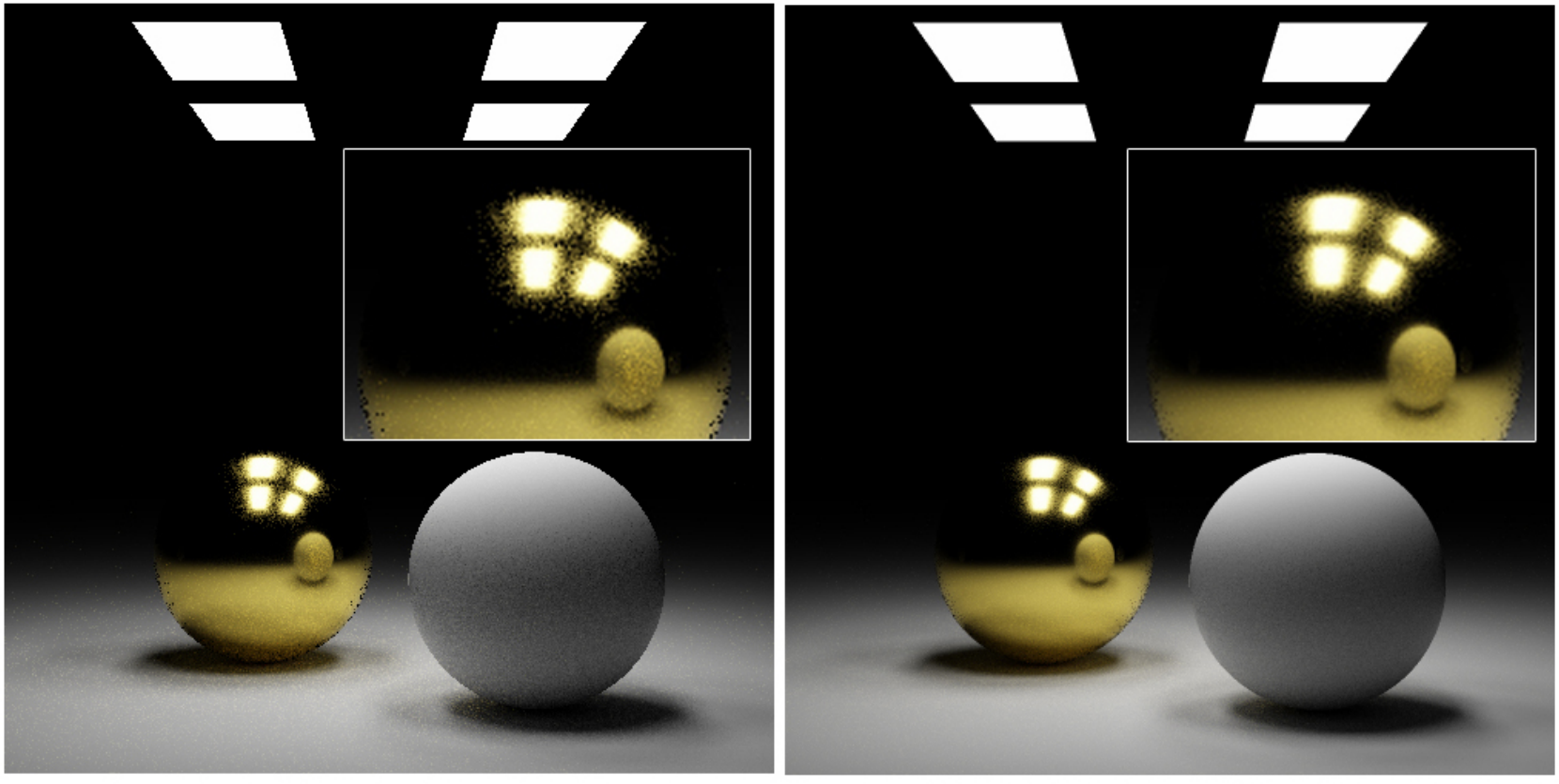


Figure : 4x (2x2) supersampling (Left: 100spp , Right: 100spp).

Parallel Programming - OpenMP

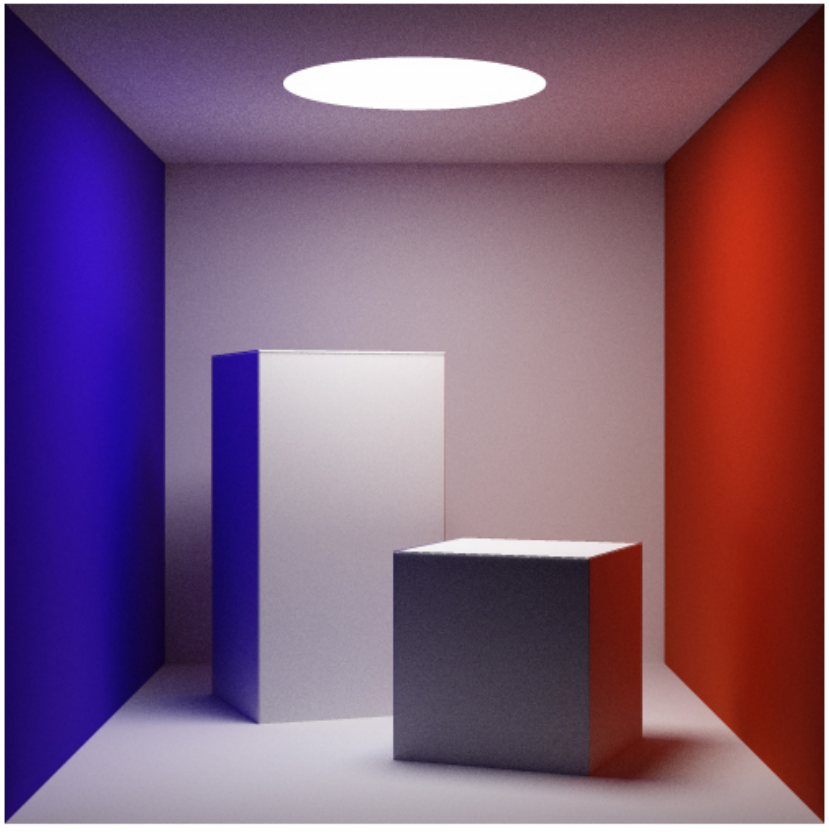
OpenMP

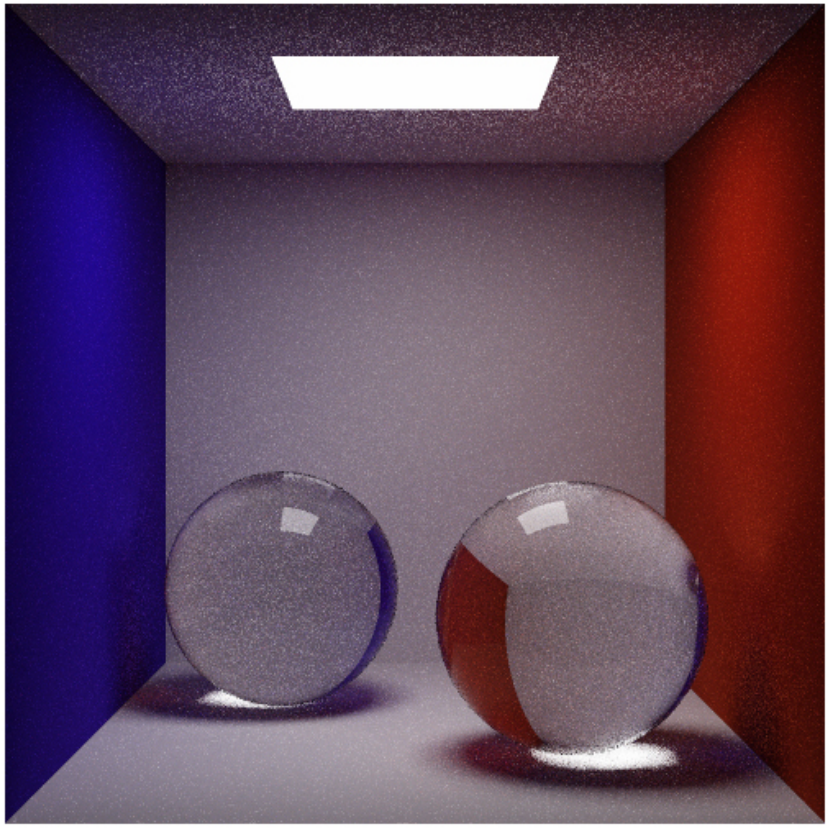
Path tracing is a highly parallel algorithm.

- GNU-Linux: compile and link with -fopenmp
- Windows: compile and link with /openmp
- Mac: I wish I had one :) , should be like GNU-Linux

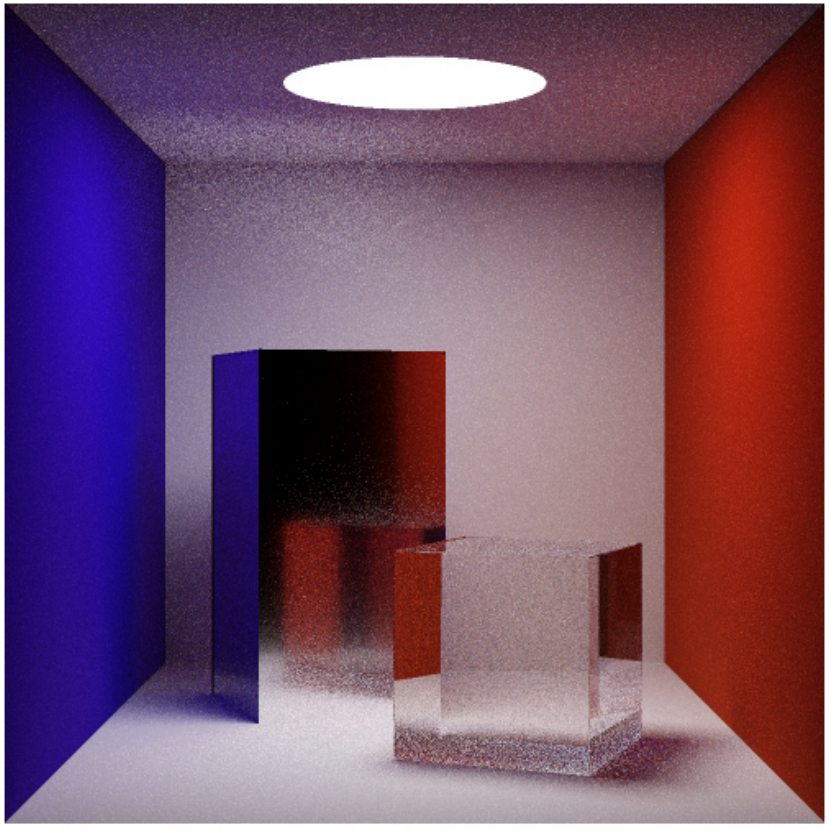
```
for (int y = 0; y < HEIGHT; y++)
    #pragma omp parallel for schedule(dynamic, 1)
    for (int x = 0; x < WIDTH; x++)
     ....
```

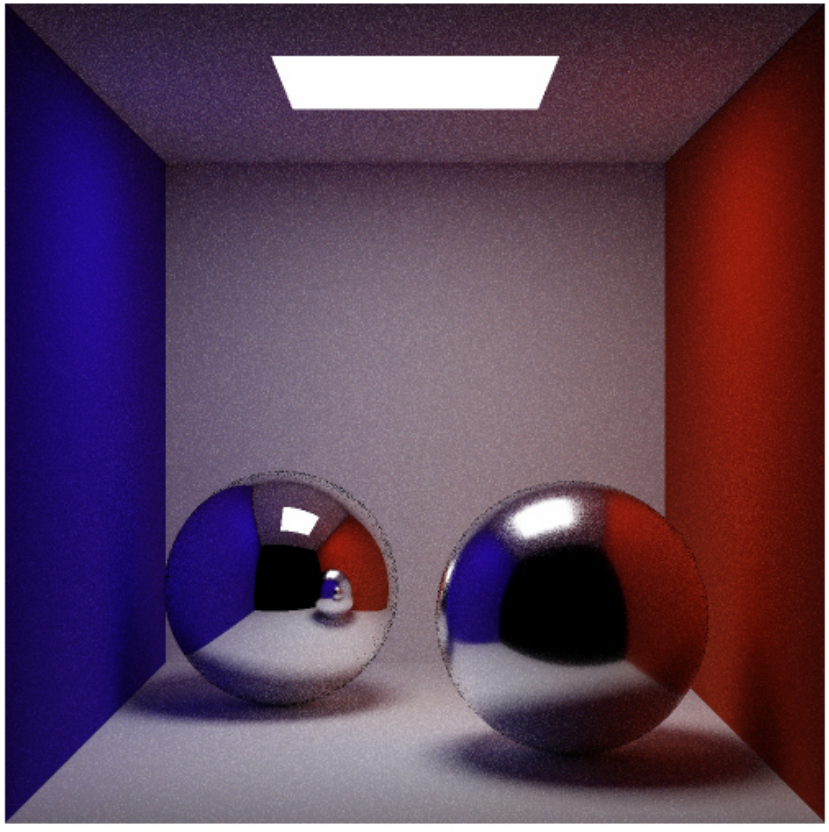
More examples



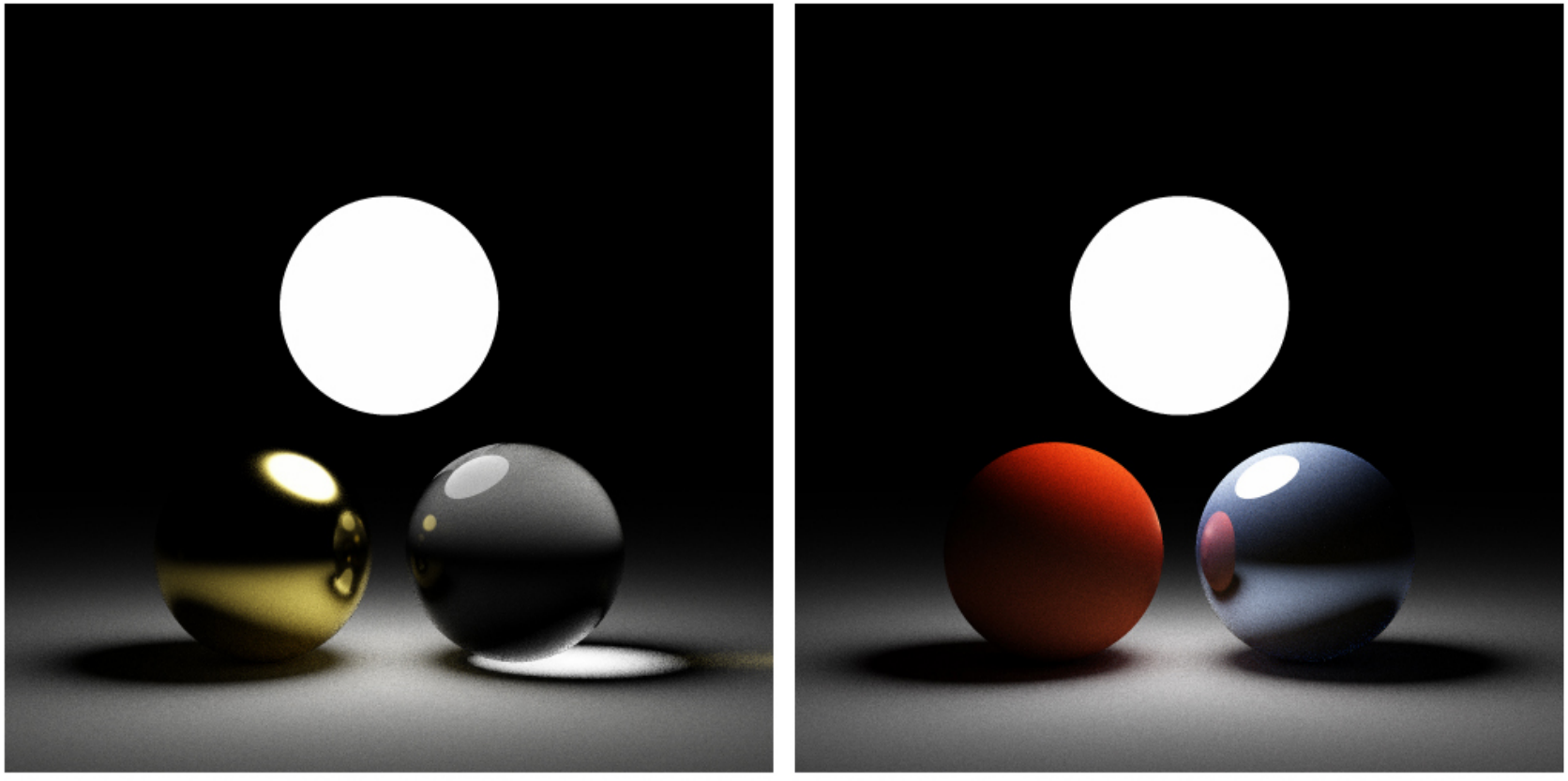


More examples





More examples




Peter Shirley and R. Keith Morley (2003). "Realistic Ray Tracing", *A K Peters/CRC Press*,2nd edition, 2003.


Peter Shirley and Steve Marschner (2009). "Fundamentals of Computer Graphics" , *AK Peters Ltd.*, 3rd edition, 2009.


Philip Dutré, Kavita Bala, and Philippe Bekaert. (2006). "Advanced Global Illumination", *AK Peters Ltd.*,2nd edition, 2006.

Thank You!!

Questions??