VSB TECHNICAL UNIVERSITY OF OSTRAVA
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
DEPARTMENT OF COMPUTER SCIENCE

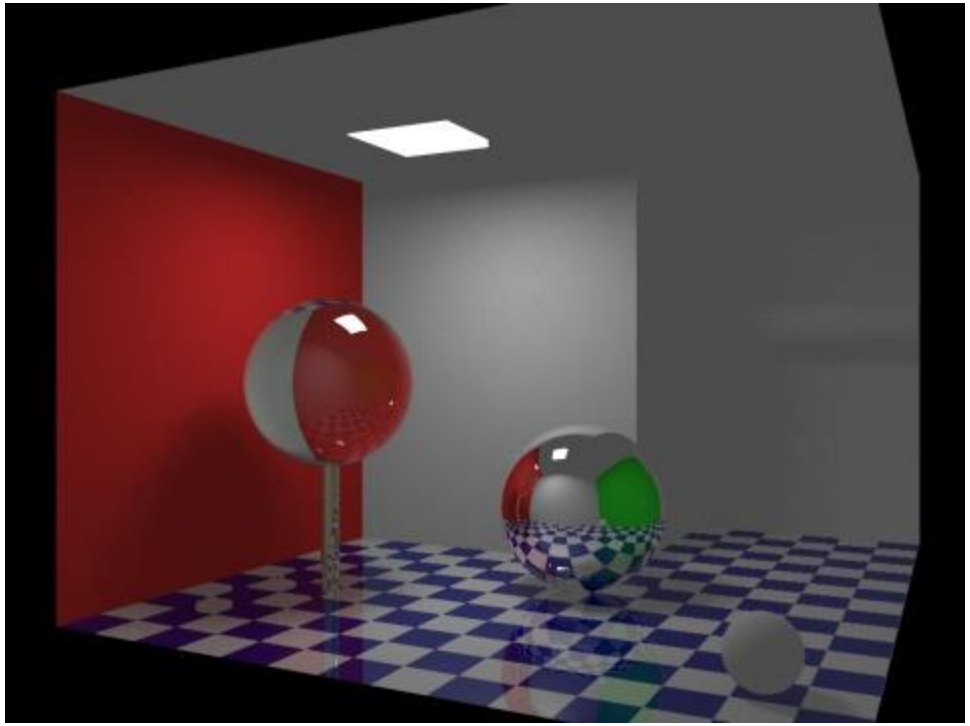
Computer Graphics I

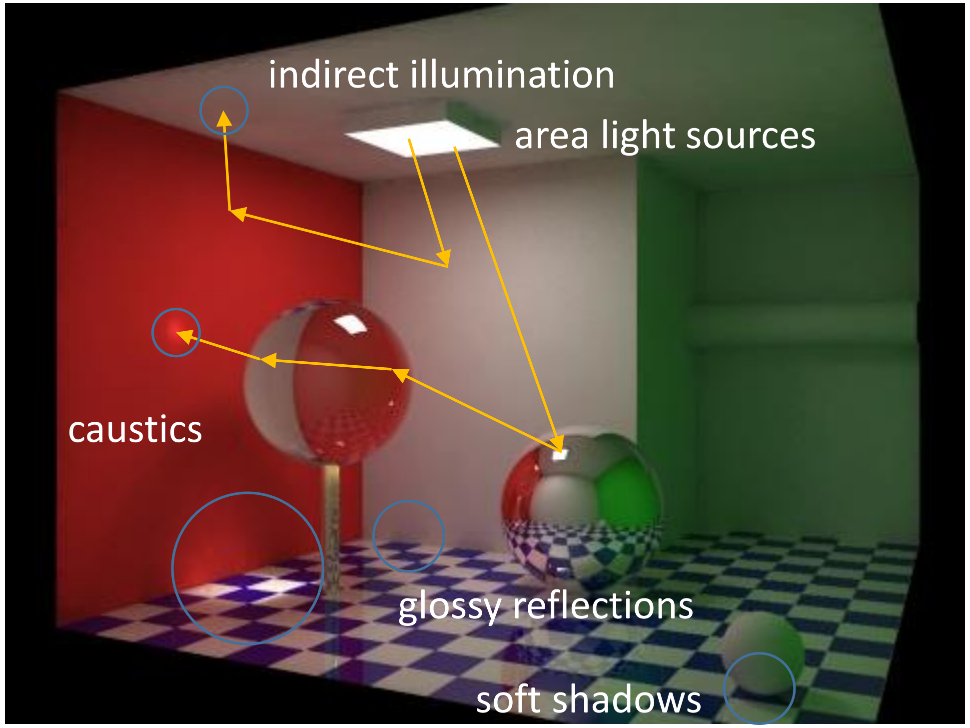
460-4078

Fall 2024

Last update 31. 10. 2024

Global Illumination = Direct I. + Indirect I.





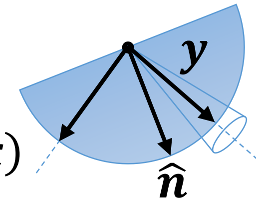
Global Illumination

- The light transport equation (LTE) is the governing equation that describes the equilibrium distribution of radiance in a scene.
- What makes evaluating the LTE difficult is the fact that incident radiance at a point is affected by the geometry and scattering properties of all of the objects in the scene. Rendering algorithms that account for this complexity are often called global illumination algorithms.
- The key principle behind the LTE is energy balance

$$\Phi_o - \Phi_i = \Phi_e - \Phi_a$$

- The difference between the power leaving an object, Φ_o, and the power entering it, Φ_i, is equal to the difference between the power it emits, Φ_e, and the power it absorbs, Φ_a.

Rendering Equation



Angle between incoming direction and normal

Scattering (BRDF) function

Angle between incoming direction and normal

- Reflection equation in angular form

$$L_o(\boldsymbol{x}, \omega_0) = L_e(\boldsymbol{x}, \omega_0) + \underbrace{\int_{H(\boldsymbol{x})} L_i(\boldsymbol{x}, \omega_i)\, f_r(\boldsymbol{x}, \omega_i, \omega_0) \cos\theta_i \, \mathrm{d}\omega_i}_{L_r(\boldsymbol{x}, \omega_0)}$$

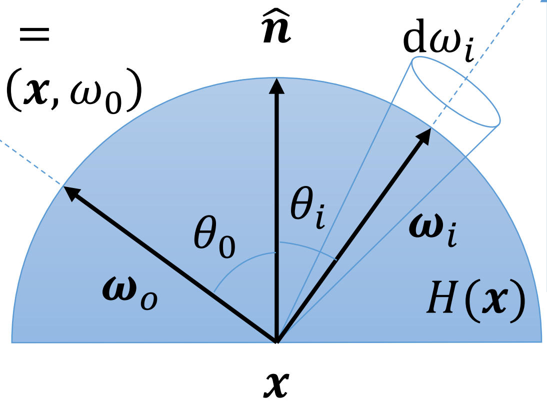
Outgoing radiance

Emitted radiance

Incoming radiance $L_i(\boldsymbol{x}, \omega_i) = L_o(\boldsymbol{r}(\boldsymbol{x}, \omega_i), -\omega_i)$

$L_r(\boldsymbol{x}, \omega_0)$

$L_o(\boldsymbol{x}, \omega_0) =$
$L_e(\boldsymbol{x}, \omega_0) + L_r(\boldsymbol{x}, \omega_0)$



Ray casting function

Reflected radiance as the sum of contributions over the hemisphere

This equality holds due to the constancy of the radiance along the ray ($L_i(\boldsymbol{x}, \boldsymbol{y} \to \boldsymbol{x})$ and $L_o(\boldsymbol{y}, \boldsymbol{y} \to \boldsymbol{x})$ must be the same provided that $\boldsymbol{y} = \boldsymbol{r}(\boldsymbol{x}, \omega_i)$)

Rendering Equation

- Note that the integral without the BRDF function represents irradiance

$$E(\boldsymbol{x}) = \int_{H(\boldsymbol{x})} L_i(\boldsymbol{x}, \omega_i) \cos\theta_i \, \mathrm{d}\omega_i$$

- The irradiance of surface patch $\boldsymbol{x}$ is given by integrating over all incoming radiance weighted by the projected area of the receiving surface in the direction of the incoming light directions

Rendering Equation

- Rendering equation in angular form ($L = L_o$)

$$L(\boldsymbol{x}, \omega_0) = L_e(\boldsymbol{x}, \omega_0) + \int\limits_{H(\boldsymbol{x})} L(\boldsymbol{r}(\boldsymbol{x}, \omega_i), -\omega_i)\, f_r(\boldsymbol{x}, \omega_i, \omega_0) \cos\theta_i \, \mathrm{d}\omega_i$$

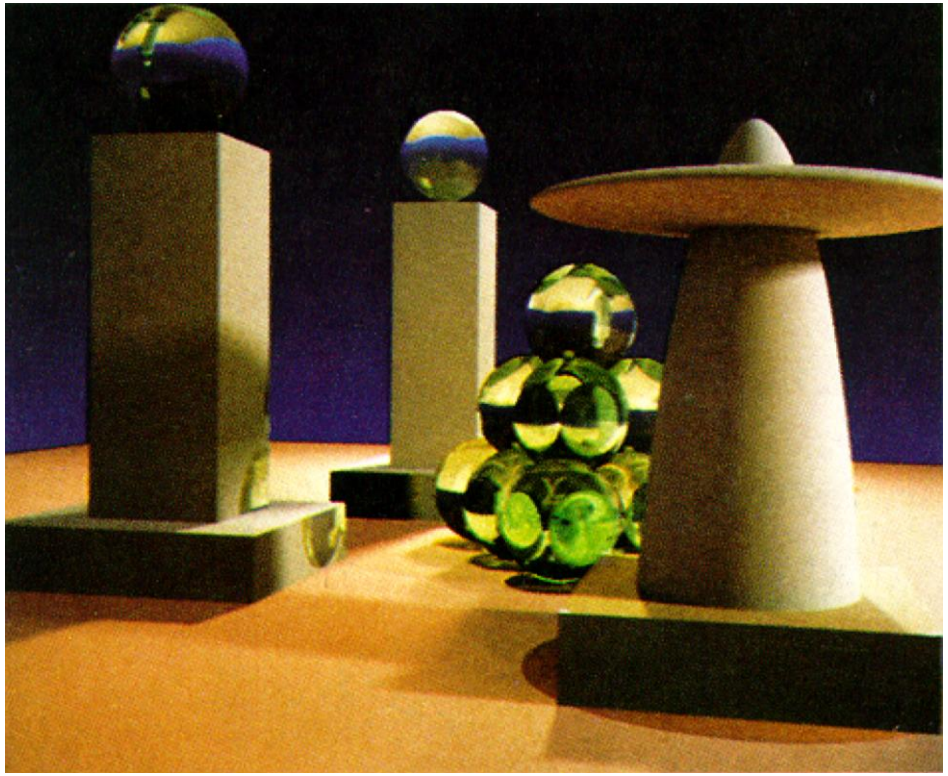
constant limits of integration

- Fredholm integral equation of the second kind (unknown radiance L appears both inside and outside of the integral)
- Describes the steady state

$$\varphi(t) = f(t) + \lambda \int_a^b K(t,s)\varphi(s)\mathrm{d}s$$

Given the kernel $K(t,s)$, and the function $f(t)$, the problem is typically to find the function $\varphi(t)$

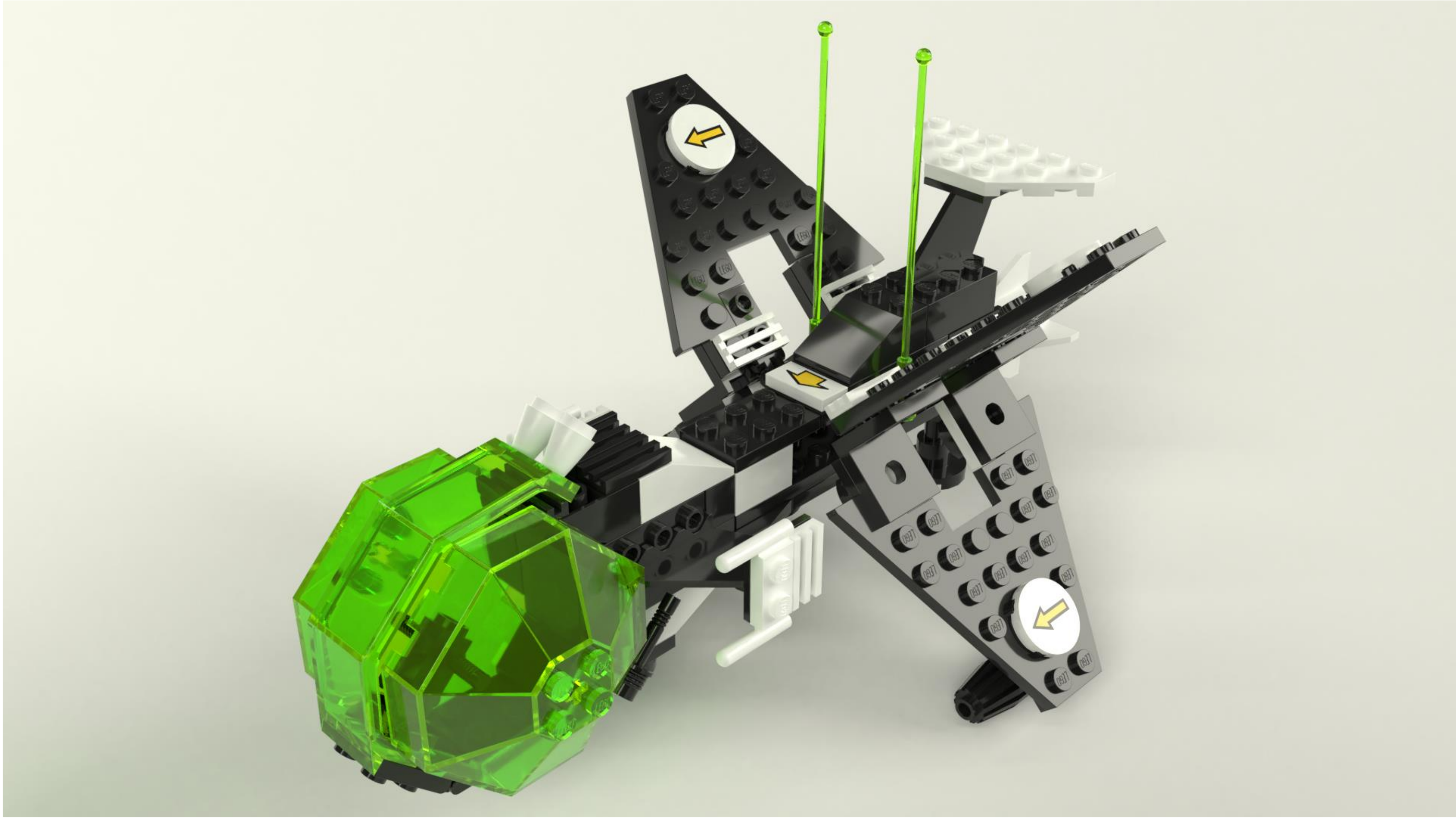
Examples of Rendering Equation Solution



Source: KAJIYA, James T. The rendering equation. In: *ACM Siggraph Computer Graphics*. ACM, 1986. p. 143-150.

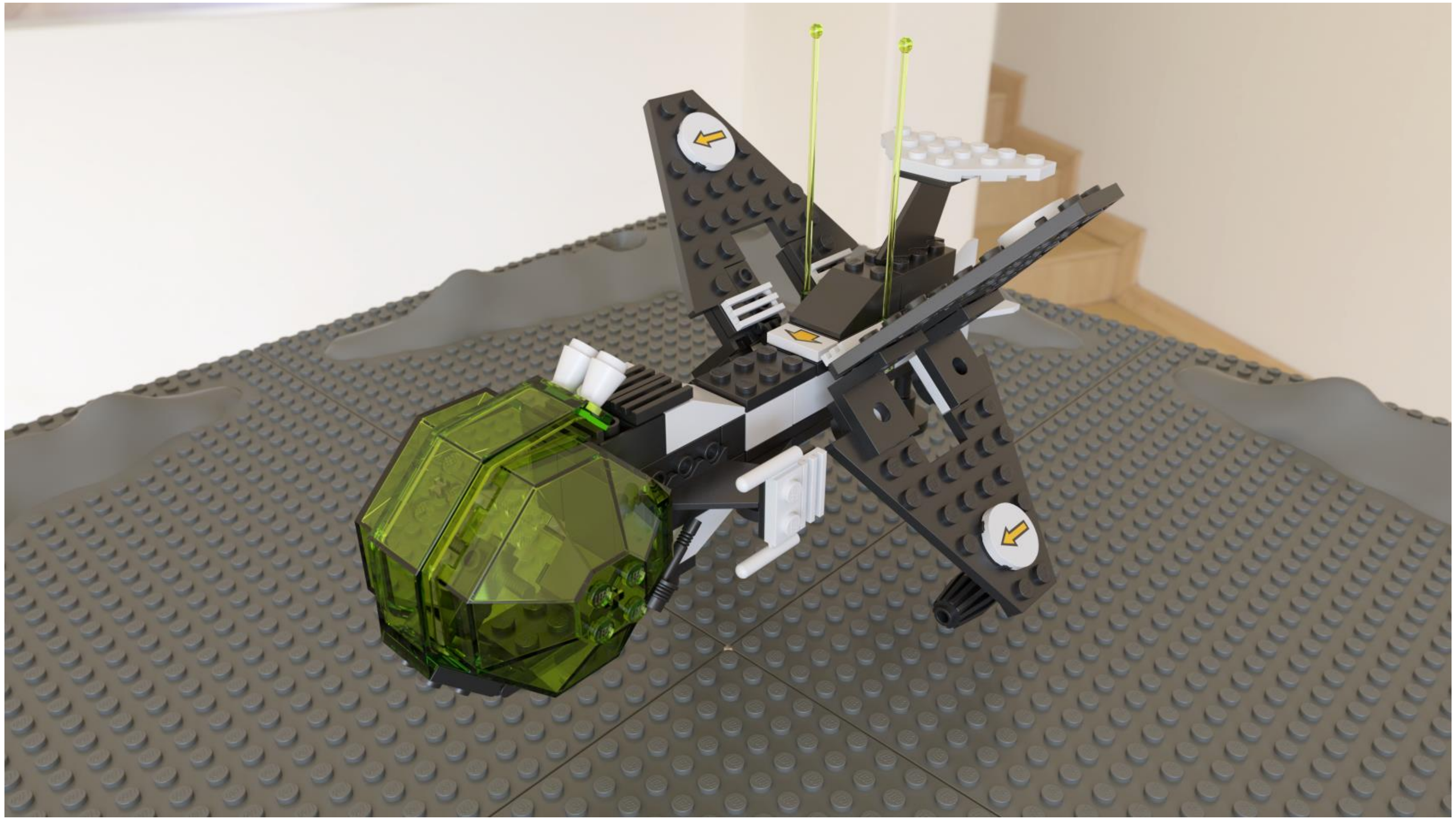
„Figure 6 is a 512 by 512 pixel image with 40 paths per pixel. It was computed on an IBM 3081 and consumed 1221 minutes of CPU time.“

Examples of Rendering Equation Solution



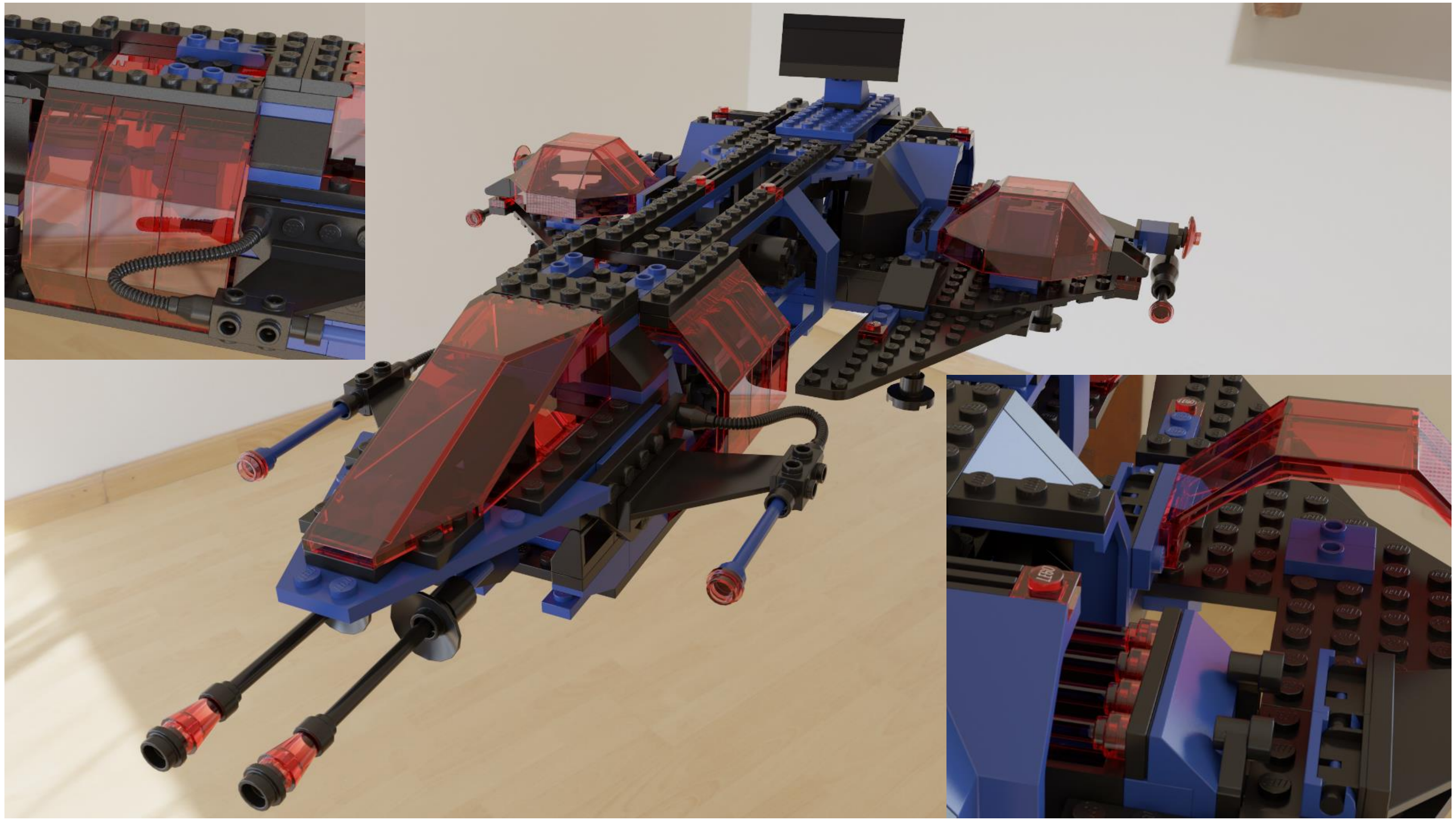
Created with Cycles, an physically based production renderer natively integrated in Blender, Poser, and Rhino in less than a few tens of seconds on common GPU

Examples of Rendering Equation Solution



Created with PG1 Renderer (path tracing)

Examples of Rendering Equation Solution



Created with PG1 Renderer (path tracing)

Examples of Rendering Equation Solution

Created with PG1 Renderer
(path tracing)

Light Transport Approximation Assumptions

- Geometrical optics
  - No diffraction, no polarization, no interference
- Discrete-wavelength approximation of color
  - Quantized approx. of dispersion (rainbows) and fluorescence (emission of light by a substance that has absorbed light or other electromagnetic radiation)
- No propagation media
  - No atmospheric scattering (fog, clouds) or refraction (mirages)
- Light travels in a straight line
  - No gravity lenses
- Superposition (adding lights)
  - No non-linear reflecting materials
  - Non-linearity of observer or display will be handled separately

Source: Marc Levoy, Computer Graphics: Image Synthesis Techniques Notes

Rendering Equation

- Rendering equation in **angular form** (int. over hemisphere)

$$L(\boldsymbol{x}, \omega_0) = L_e(\boldsymbol{x}, \omega_0) + \int_{H(\boldsymbol{x})} L(\boldsymbol{r}(\boldsymbol{x}, \omega_i), -\omega_i)\, f_r(\boldsymbol{x}, \omega_i, \omega_0) \cos\theta_i \, \mathrm{d}\omega_i$$

- Substituting $\mathrm{d}\omega_i = \frac{\mathrm{d}A_y \cos\theta_y}{r^2}$ yields **area form** (int. over surface)

$$L(\boldsymbol{x}, \omega_0) = L_e(\boldsymbol{x}, \omega_0) + \int_A L(\boldsymbol{y} \to \boldsymbol{x})\, f_r(\boldsymbol{y} \to \boldsymbol{x} \to \omega_0) G(\boldsymbol{x}, \boldsymbol{y}) V(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}A_y$$

Geometry term $G(\boldsymbol{x}, \boldsymbol{y}) = \frac{\cos\theta_i \cos\theta_y}{\|\boldsymbol{x}-\boldsymbol{y}\|^2}$

Visibility term {0, 1}

Scene surface

Angular and Area Form of Rendering Equation

Angular Form

- Find the closes intersection p
- For each direction from p, find the nearest surface or background
- Return the outgoing radiance at that point as the sum of radiance contributions multiplied with the cosine-weighted BRDF and average the result over the hemisphere
- General use case

Area Form

- Find the closes intersection p
- Find all other points on the scene surface (mutually) visible from p
- Return the outgoing radiance at that point as the sum of radiance contributions multiplied with the geometry term-weighted BRDF and average the result over the visible surface points
- Calculating DI for area light sources

Operator form of Rendering Equation

- Rendering equation in angular form (int. over hemisphere)

$$(T \circ L)(\boldsymbol{x}, \omega_0) = \int_{H(\boldsymbol{x})} L(\boldsymbol{r}(\boldsymbol{x}, \omega_i), -\omega_i)\, f_r(\omega_i, \omega_0) \cos\theta_i \,\mathrm{d}\omega_i$$

- Rendering equation rewritten with linear transport operator T

$L = L_e + T \circ L$, formal solution $L = (I - T)^{-1} \circ L_e$ (practically unusable)

- Recursive substitution of L yields the Neumann series

$$L = L_e + T \circ L = L_e + T \circ (L_e + T \circ L) = \cdots$$

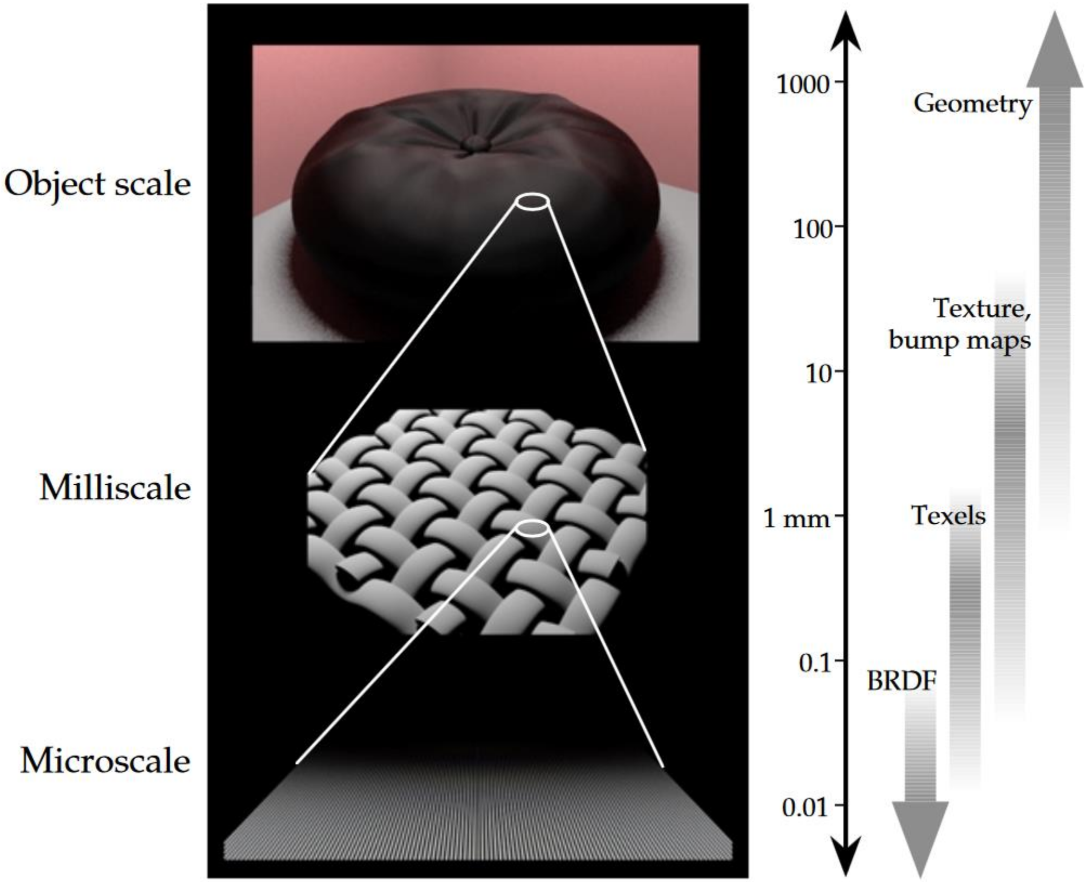
1st bounce 2nd bounce Negligible contribution

$$= \overbrace{L_e + T \circ L_e}^{\text{OpenGL}} + \underbrace{T^2 \circ L_e + T^3 \circ L_e + \cdots + T^\infty \circ L_e}_{\text{Indirect illumination}}$$

Emission only Direct illumination

Representations Used in Realistic Rendering

- At the smallest scale reflectance function accurately captures the appearance of a surface
- As individual surface features become larger than one pixel, texture maps, bump maps, and texels can be used to show surface features
- At the largest scale, the geometry must be modeled explicitly



Source: WESTIN, Stephen H.; ARVO, James R.; TORRANCE, Kenneth E. *Predicting reflectance functions from complex surfaces*. ACM, 1992.

Light Interaction with Surfaces

- Absorption
- **Reflection**
- Transmission or refraction

- Reflection is the relation of reflected radiance L_r to incoming radiance L_i
- Determines the appearance of objects on microscopic level

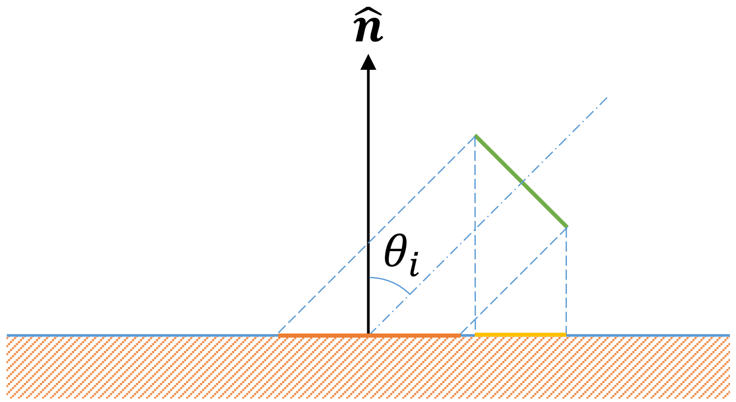
Geometry → Bump maps → Texels → BRDF

Macrostructures *Microstructures*

Source: MATUSIK, Wojciech. *A data-driven reflectance model*. 2003. PhD Thesis. Massachusetts Institute of Technology.

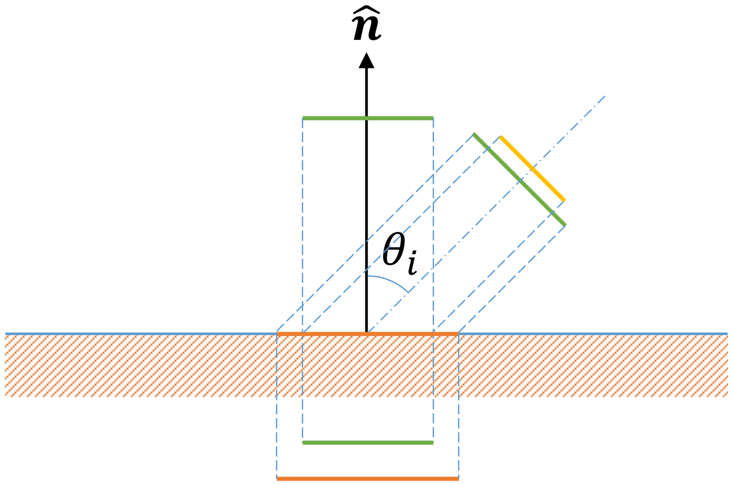
Weakening Factor

- What does the cosinus term stand for?

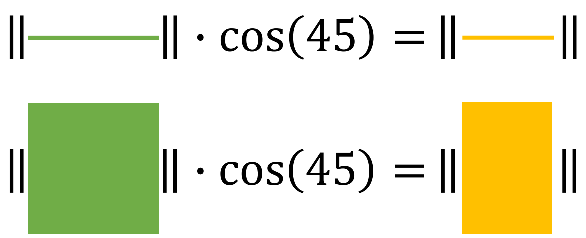


Projected area for basic shapes

Shape	Area	Projected area
Flat rectangle	$A = L \times W$	$A_{proj} = L \times W \cos\beta$
Circular disc	$A = \pi r^2$	$A_{proj} = \pi r^2 \cos\beta$
Sphere	$A = 4\pi r^2$	$A_{proj} = \frac{A}{4} = \pi r^2$



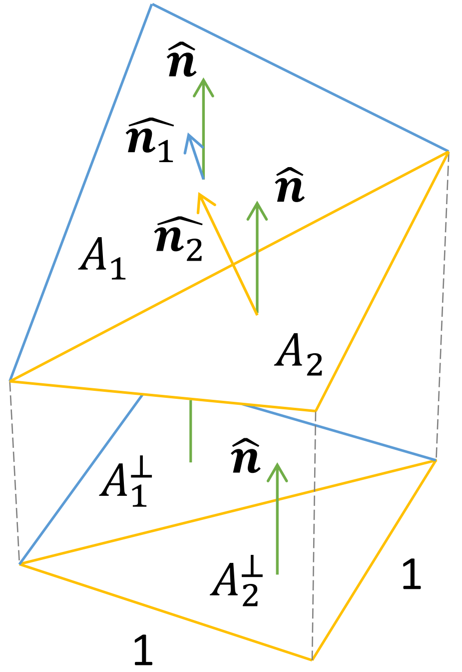
$$\cos(\theta_i = 45) = 0.71$$



$$A_{\text{projected}} = \int_A \cos\theta_i \, \mathrm{d}A$$

Weakening Factor

- What does the cosinus term stand for?
- Here we have another example: What will happen to the areas A_1 and A_2 of triangles projected onto a plane given by its normal vector $\hat{\boldsymbol{n}}$?



$$n := \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$n1 := \begin{bmatrix} -0.2357 \\ -0.2357 \\ 0.9428 \end{bmatrix} \qquad n2 := \begin{bmatrix} 0.1961 \\ -0.5883 \\ 0.7845 \end{bmatrix}$$

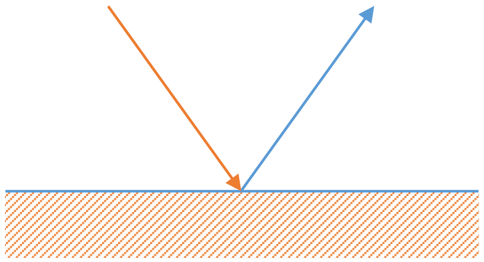
$$A1 := 0.5303 \qquad A2 := 0.6374$$

$$\theta1 := \mathrm{acos}(n \cdot n1) = 19.473\ \mathbf{deg} \qquad \theta2 := \mathrm{acos}(n \cdot n2) = 38.326\ \mathbf{deg}$$

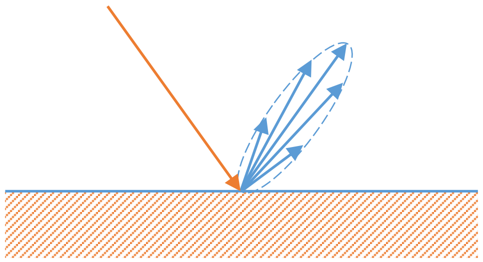
$$\boxed{A1p := A1 \cdot \cos(\theta1) = 0.5} \qquad \boxed{A2p := A2 \cdot \cos(\theta2) = 0.5}$$

$$A1p + A2p = 1$$

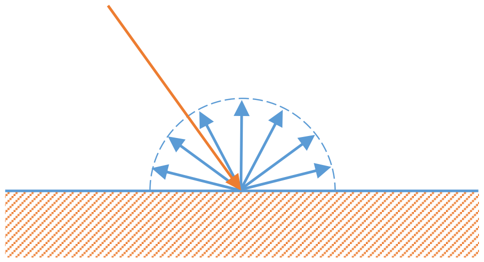
Types of (Idealized) Reflections



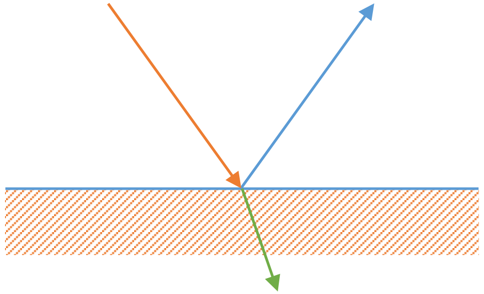
specular reflection



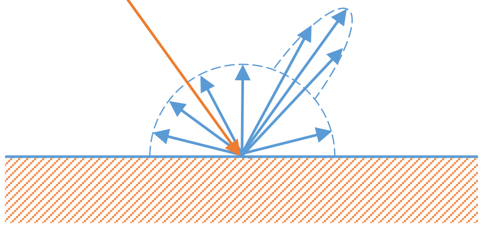
glossy reflection



diffuse reflection



specular reflection + refraction



diffuse + glossy reflection

Types of Materials

- Metals (conductors)

  In the case of metals, free electrons prevent light from penetrating the metal surface, so scattering cannot occur and metallic substances show only specular reflection and no diffuse reflection

- Dielectrics (insulators)

  Insulators exhibit significant amount of diffuse reflectivity while the specular reflectance is limited

- Semiconductors

  We will not deal with them here

BRDF (sr^{-1})

Radiance and irradiance relation

$$L_r(\boldsymbol{x}, \omega_o) \overset{\text{e.g.}}{=} f_{Lambert} \int_{H(\boldsymbol{x})} L_i(\boldsymbol{x}, \omega_i) \cos\theta_i \, \mathrm{d}\omega_i = \frac{\rho_d}{\pi} E(\boldsymbol{x})$$

"total" irradiance of point $\boldsymbol{x}$ from all directions given by $H(\boldsymbol{x})$

- Bidirectional Reflectance Distribution Function

part of the reflected radiance from point $\boldsymbol{x}$ in direction ω_o

$$f_r(\omega_i, \omega_o) = \overbrace{f_r(\omega_i \to \omega_o) = f_r(\omega_o \to \omega_i)}^{\textbf{I.}\text{ Helmholz reciprocity}} = \frac{\mathrm{d}L_r(\boldsymbol{x}, \omega_o)}{\underbrace{L_i(\boldsymbol{x}, \omega_i) \cos\theta_i \, \mathrm{d}\omega_i}_{\mathrm{d}E(\boldsymbol{x}, \omega_i)}}$$

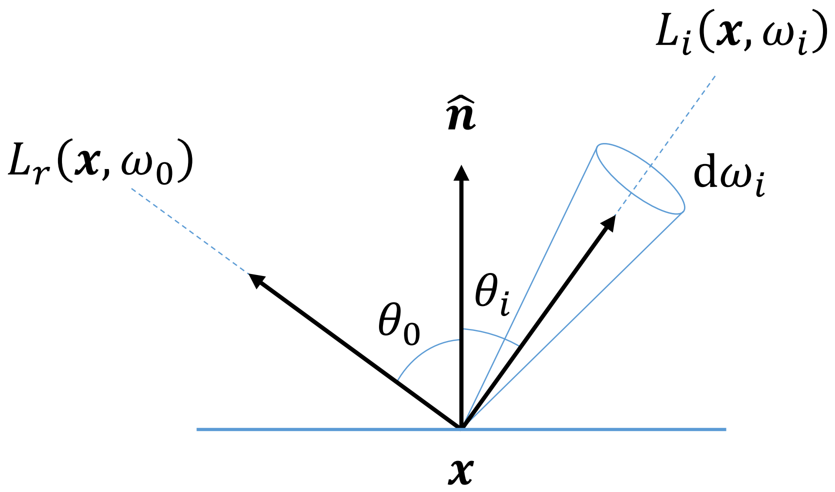
… induced by …

incoming "partial" irradiance of point $\boldsymbol{x}$ from direction ω_i

II. Energy conservation

$$\int_H f_r(\omega_i, \omega_o) \cos\theta_i \, \mathrm{d}\omega_i \le 1$$

Surface cannot reflect more energy than it receives



III. Positivity

Range $f_r \in \langle 0, \infty)$

BRDF

- Validation criteria (actually very weak conditions)
  - positivity
  - reciprocity
  - energy conservation
- Those criteria are not sufficient to validate a new model, because they are not restrictive enough. Intuitively, one could come up with some random BRDF model that easily satisfies those three conditions, and yet fails to relate to any meaningful physical model.

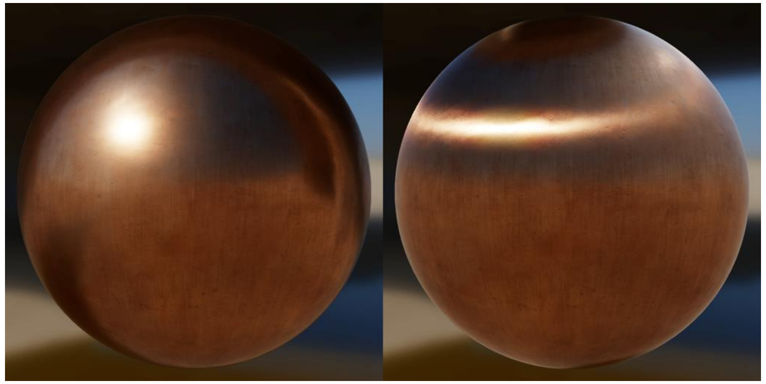
Source: E. Heitz, Understanding the Masking-Shadowing Function in Microfacet-Based BRDFs, 2014.

(An)isotropic BRDF

- Isotropic BRDF is invariant to a rotation around surface normal

$$f_r(\theta_i, \varphi_i, \theta_o, \varphi_o) = f_r(\theta_i, \varphi_i + \boldsymbol{\varphi}, \theta_o, \varphi_o + \boldsymbol{\varphi}) = f_r(\theta_i, \theta_o, \boldsymbol{\varphi_i} - \boldsymbol{\varphi_o})$$

- Only 3 instead of 4 directional degrees of freedom

isotropic BRDF  anisotropic BRDF

Source: https://google.github.io/filament/images/material_anisotropic.png

BRDF Components

Ideal diffuse reflection (Lambert)

+

Ideal specular reflection (mirror)

+

Glossy reflections (directional diffuse)

=

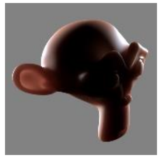
General BRDF

Taxonomy of Reflectance Functions

General scattering function: 14 D
$(x_i, y_i, z_i, \theta_i, \phi_i, \lambda_i, t_i, x_o, y_o, z_o, \theta_o, \phi_o, \lambda_o, t_o)$

16D with additional θ_t, ϕ_t (transmittance)

↓ Ignore time, wavelength dependence; Assume free space



BSSRDF / 8D-RF: 8D
$(x_i, y_i, \theta_i, \phi_i, x_o, y_o, \theta_o, \phi_o)$

↙ Ignore subsurface scattering

↘ Ignore directional dependence

BRDF + texture

SVBRDF / BTF / 6D-RF: 6D
$(x, y, \theta_i, \phi_i, \theta_o, \phi_o)$

↓ Ignore spatial variation

BRDF: 4D
$(\theta_i, \phi_i, \theta_o, \phi_o)$

↓ Assume isotropy

Isotropic BRDF: 3D
$(\theta_i, \theta_o, |\phi_i - \phi_o|)$

Heterogeneous Scattering: 4D
(x_i, y_i, x_o, y_o)

↓ Ignore spatial variation

Homogeneous Scattering: 1D
$(\|\boldsymbol{x}_i - \boldsymbol{x}_o\|)$

Source: WEYRICH, Tim, et al. *Principles of appearance acquisition and representation*. Now Publishers Inc, 2009.

Basic BRDFs

- Perfect mirror

$$f_r^{Mirror}(\omega_i, \omega_o) = \rho_s \cdot \begin{cases} \infty \text{ if } \theta_i = \theta_o \\ 0 \text{ otherwise} \end{cases} = \rho_s \frac{\delta(\cos\theta_i - \cos\theta_o)\delta(\varphi_o - \varphi_i + \pi)}{\cos\theta_i}$$

All incident radiance is reflected in a single specular direction and scaled by factor ρ_s (specular albedo)

where $\rho_s \leq 1$

- Perfect diffusor (Lambertian surface)

$$f_r^{Lambert}(\omega_i, \omega_o) = \frac{\rho_d}{\pi}$$

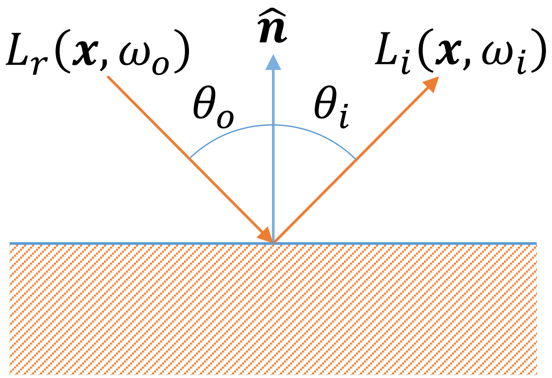
ρ_d (albedo) is the ration of outgoing and incoming flux

- Modified Phong (physically correct but still empirical model)

$$f_r^{Phong}(\omega_i, \omega_o) = \frac{\rho_d}{\pi} + \frac{\rho_s(\gamma+2)}{2\pi}(cos\,\theta_r)^\gamma$$ where $\rho_d + \rho_s \leq 1$ and

θ_r is angle between ω_o and perfect specular reflection of ω_i or vice versa

Pure Specular Reflector (Mirror)



- From energy conservation criterium we know that

$$\int_0^{2\pi} \int_0^{\frac{\pi}{2}} \frac{\delta(\cos\theta_i - \cos\theta_o)\delta(\varphi_o - \varphi_i + \pi)}{\cos\theta_i} \cos\theta_i \sin\theta_i \, \mathrm{d}\theta_i \mathrm{d}\varphi_i = 1$$

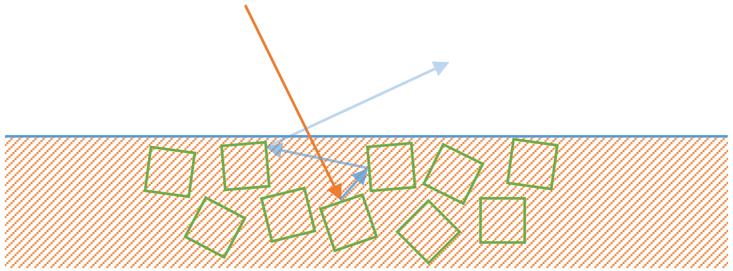
- $pdf(\omega_i) = \delta(\cos\theta_i - \cos\theta_o)\delta(\varphi_o - \varphi_i + \pi)$
- Let's check that $\int_H \; pdf(\omega_i)\mathrm{d}\omega_i = 1$ (pdf must sum up to 1)

$$\int_0^{2\pi} \int_0^{\frac{\pi}{2}} \delta(\cos\theta_i - \cos\theta_o)\delta(\varphi_o - \varphi_i + \pi) \sin\theta_i \, \mathrm{d}\theta_i \mathrm{d}\varphi_i = 1$$

- Recall that δ is the Dirac delta distribution defined such that $\delta(x) = 0$ for all $x \neq 0$ and $\int_{-\infty}^{\infty} \delta(x) \, \mathrm{d}x = 1$

Diffuse Material

- Incident ray is scattered at many angles
- Ideal diffuse material is said to be Lambertian = equal luminance (radiance) when viewed from all directions lying in „upper“ hemisphere
- Good examples of solid diffuse reflectors are plaster, paper, or polycrystalline materials (exhibit subsurface scattering mechanism caused by internal subdivisions)
- Few materials do not cause diffuse reflection: metals (do not allow light to enter), gases, liquids, glass, and transparent plastics



Diffuse BRDF

- For any combination of input and output directions, we want the surface reflectance to be a constant and energy conserving

$$1 \geq \int_{H(\boldsymbol{x})} f_r(\boldsymbol{x}, \omega_i, \omega_o) \cos(\theta_i) \, \mathrm{d}\omega_i = f_r \int_0^{2\pi} \int_0^{\pi/2} \cos(\theta_i) \sin(\theta_i) \, \mathrm{d}\theta_i \mathrm{d}\varphi_i = \underbrace{}_{\begin{bmatrix} u = \cos(\theta_i) \\ \mathrm{d}u = -\sin(\theta_i)\mathrm{d}\theta_i \end{bmatrix} \text{ substitution}}$$

$$= -f_r \int_0^{2\pi} \int_1^0 u \, \mathrm{d}u \, \mathrm{d}\varphi_i = f_r \int_0^{2\pi} \left[\frac{u^2}{2}\right]_0^1 \mathrm{d}\varphi_i = \frac{f_r}{2} \int_0^{2\pi} \mathrm{d}\varphi_i = \frac{2\pi f_r}{2} = \pi f_r$$

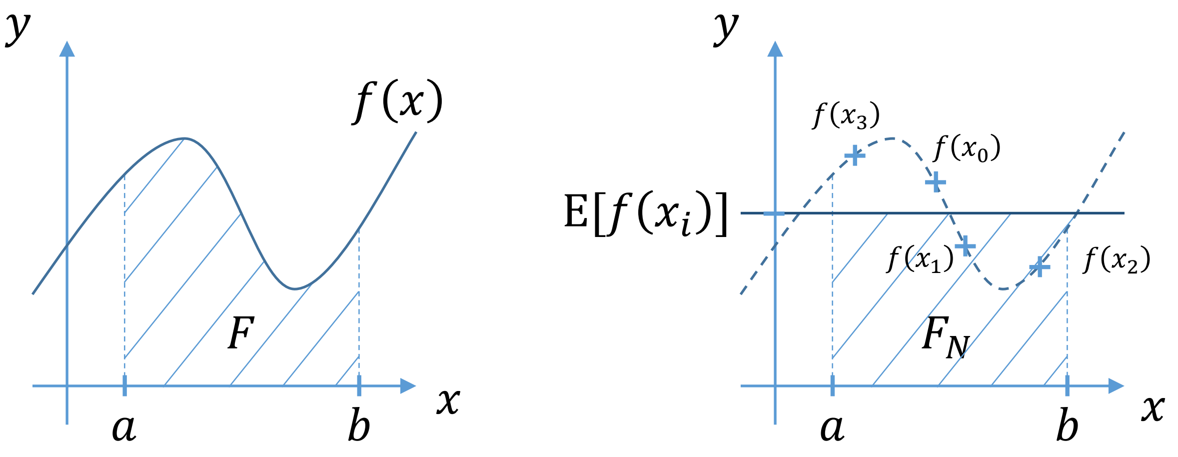
- To conclude, it must hold that $1 \geq \pi f_r \Longrightarrow f_r = \frac{albedo}{\pi}$ where $albedo \in \langle 0,1 \rangle^3$

Monte Carlo Estimator

- Suppose we want to evaluate 1D integral with MC estimator given uniform random samples $x_i \in \langle a, b \rangle$

$$F = \int_a^b f(x)\mathrm{d}x$$

$$F \approx F_N = (b-a)\mathrm{E}[f(x_i)] = \frac{b-a}{N}\sum_{i=0}^{N-1} f(x_i)$$



$$\mathrm{E}[f(x_i)] = \frac{1}{N}\sum_{i=0}^{N-1} f(x_i)$$

Monte Carlo Estimator

- Expected value of the etimator is equal to the integral of f

$$
\begin{aligned}
\mathrm{E}[F_N] &= \mathrm{E}\left[\frac{b-a}{N}\sum_{i=0}^{N-1} f(x_i)\right] \\
&= \frac{b-a}{N}\sum_{i=0}^{N-1} \mathrm{E}[f(x_i)] \\
&= \frac{b-a}{N}\sum_{i=0}^{N-1} \int_a^b f(x)p(x)\mathrm{d}x \\
&= \frac{1}{N}\sum_{i=0}^{N-1} \int_a^b f(x)\mathrm{d}x \\
&= \int_a^b f(x)\mathrm{d}x
\end{aligned}
$$

Note that $p(x) = \frac{1}{b-a}$

Monte Carlo Estimator

Note that $p(x)$ must be nonzero for all x where $f(x) \neq 0$

- Same holds for estimators with an arbitrary PDF $p(x)$

$$
\begin{aligned}
\mathrm{E}[F_N] &= \mathrm{E}\left[\frac{1}{N}\sum_{i=0}^{N-1}\frac{f(x_i)}{p(x_i)}\right] \\
&= \frac{1}{N}\sum_{i=0}^{N-1}\int_a^b \frac{f(x)}{p(x)}p(x)\mathrm{d}x \\
&= \frac{1}{N}\sum_{i=0}^{N-1}\int_a^b f(x)\mathrm{d}x \\
&= \int_a^b f(x)\mathrm{d}x
\end{aligned}
$$

Monte Carlo Estimator For Mirror BRDF

- Recall that for mirror BRDF we have

$$f_r(\omega_i, \omega_o) = \rho_s \frac{\delta(\cos\theta_i - \cos\theta_o)\delta(\varphi_o - \varphi_i + \pi)}{\cos\theta_i}$$

and

$$pdf(\omega_i) = \delta(\cos\theta_i - \cos\theta_o)\delta(\varphi_o - \varphi_i + \pi)$$

That gives us the (specularly) reflected (estimated) radiance as follows

Exactly the same result as in Whitted model for reflection of light

$$L_r(\boldsymbol{x}, \omega_o) = \frac{1}{N}\sum_{i=0}^{N-1} \frac{L_i(\boldsymbol{x}, \omega_i) f_r(\omega_i, \omega_o)\cos\theta_i}{p(\omega_i)} = \frac{1}{N}\sum_{i=0}^{N-1} \boxed{L_i(\boldsymbol{x}, \omega_i)\rho_s}$$

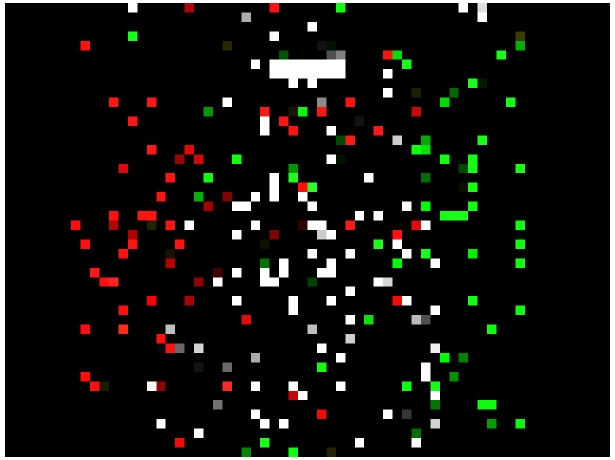
$$I = I_a + k_d \sum_{j=1}^{j=ls} (\bar{N} \cdot \bar{L}_j) + \boxed{k_s S} + k_t T$$

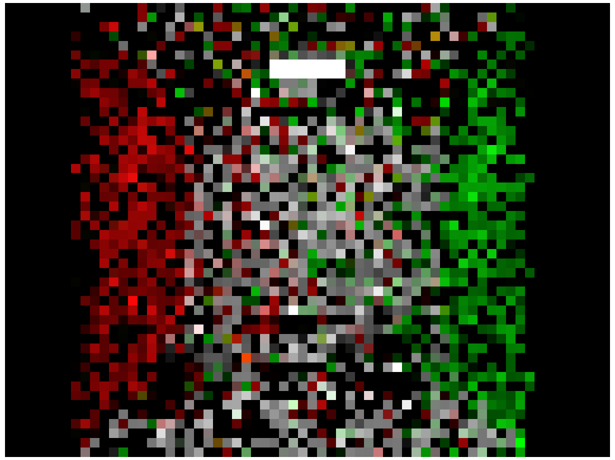
Path Tracing

```
for each pixel in the image:
    pixel := 0
    for each sample:
        ray := make_primary_ray(pixel)
        pixel += trace_ray(ray, 0)
    pixel := to_srgb(pixel / number_of_samples) // back to nonlinear space

trace_ray(ray, depth):
    if ( depth > max_depth ) return 0
    p := find_closest_intersection(ray, scene)
    if ( no hit ) return background
    L_e := get_emission(p, omega_o)   // note that omega_o = -ray.dir
    if L_e ≠ 0: return L_e            // we hit a source and stopped our light path here
    omega_i, pdf := sample_hemisphere(normal)
    L_i := trace_ray(make_secondary_ray(p, omega_i), depth + 1)
    f_r := Albedo / π                 // e.g. Lambert BRDF
    L_r := L_i * f_r * (omega_i · normal) / pdf
    return L_r
```

Convergence


10 spp


100 spp

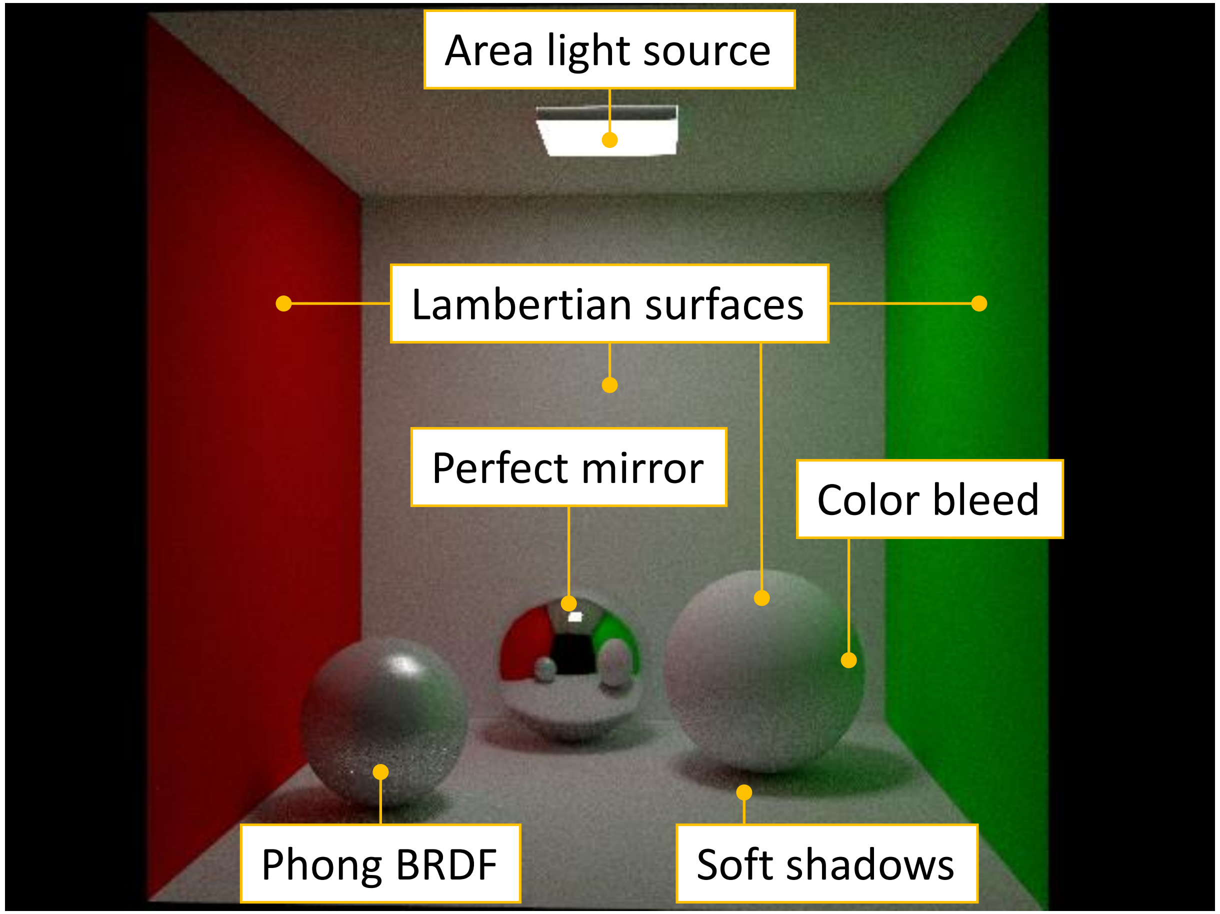

1k spp


10k spp


100k spp


1M spp

Path Tracing



Path traced Cornell Box

Naive implementation without using variance reduction techniques or direct illumination sampling.

20.000 spp, render time ≈ 1 hour

(Hemi)sphere Sampling

- Random point on a unit sphere (i.e. direction) $\omega_i = (x, y, z)$ where

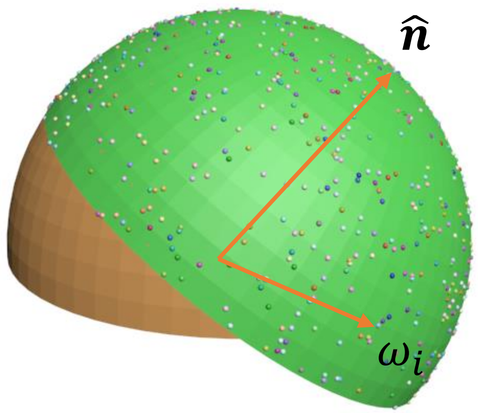
$$x = 2\cos(2\pi\xi_1)\sqrt{\xi_2(1-\xi_2)}$$

$$y = 2\sin(2\pi\xi_1)\sqrt{\xi_2(1-\xi_2)}$$

$$z = 1 - 2\xi_2$$

$$\mathrm{pdf}(\omega_i) = \frac{1}{4\pi}$$

ξ_1 and ξ_2 are pseudo random uniform variables in the range $\langle 0, 1)$



- Hemisphere sampling is almost the same: accept sample with $\omega_i \cdot \boldsymbol{n} \geq 0$ otherwise flip the sample (i.e. $-\omega_i$)

$$\mathrm{pdf}(\omega_i) = \frac{1}{2\pi}$$

Hemisphere Sampling

- Random point on a unit hemisphere (i.e. direction) $\omega_i = (x, y, z)$ where

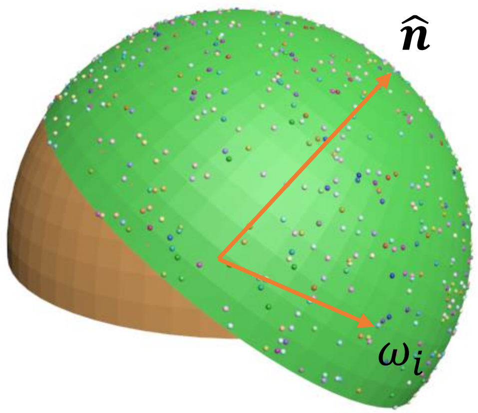
$$x = \cos(2\pi\xi_1)\sqrt{1-\xi_2^2}$$

$$y = \sin(2\pi\xi_1)\sqrt{1-\xi_2^2}$$

$$z = \xi_2$$

$$\mathrm{pdf}(\omega_i) = \frac{1}{2\pi}$$

ξ_1 and ξ_2 are pseudo random uniform variables in the range $\langle 0, 1)$



- Note that we need to transform the generated sample from the local reference frame to the world space of the scene

Hemisphere Cosine-weighted Sampling

- Random point on a unit sphere (i.e. direction) $\omega_i = (x, y, z)$ where

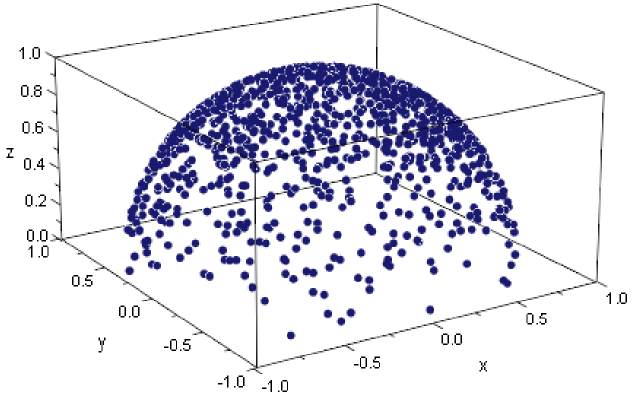
$$x = \cos(2\pi\xi_1)\sqrt{1-\xi_2}$$

$$y = \sin(2\pi\xi_1)\sqrt{1-\xi_2}$$

$$z = \sqrt{\xi_2}$$

$$\mathrm{pdf}(\omega_i) = \frac{\cos(\theta)}{\pi}$$

ξ_1 and ξ_2 are pseudo random uniform variables in the range $\langle 0, 1)$

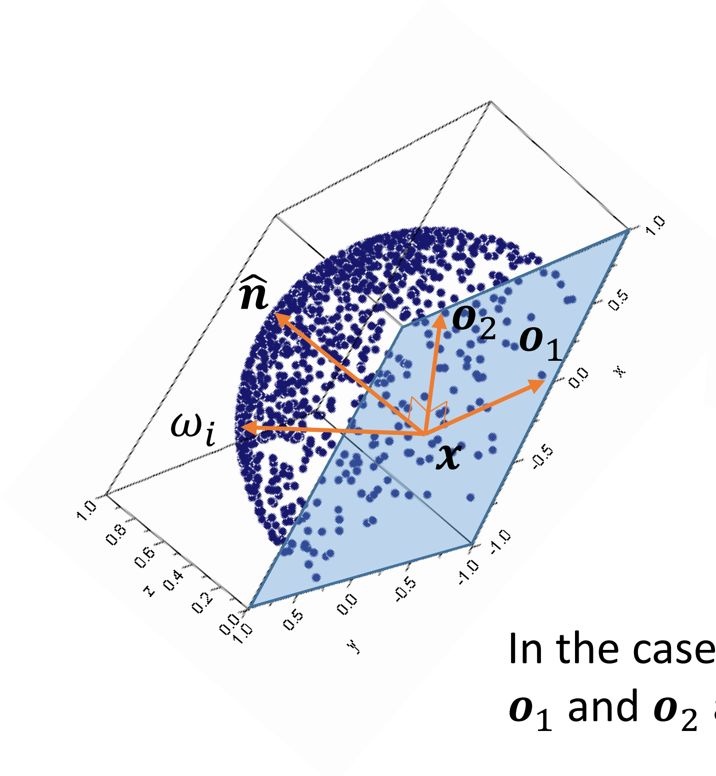


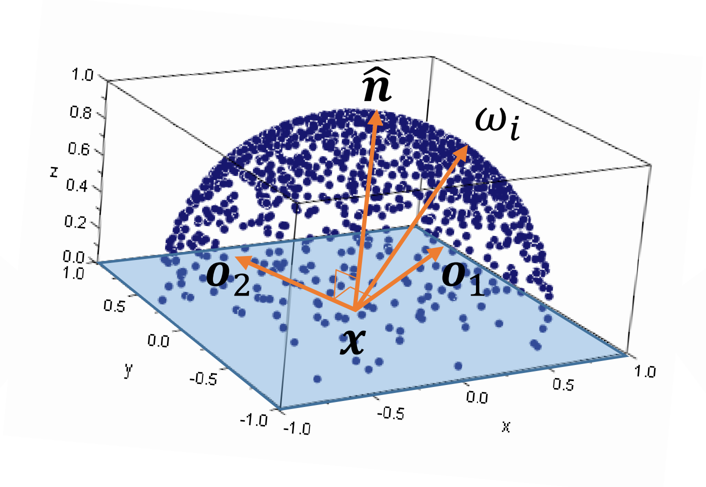
- Note that we need to transform the generated sample from the local reference frame to the world space of the scene

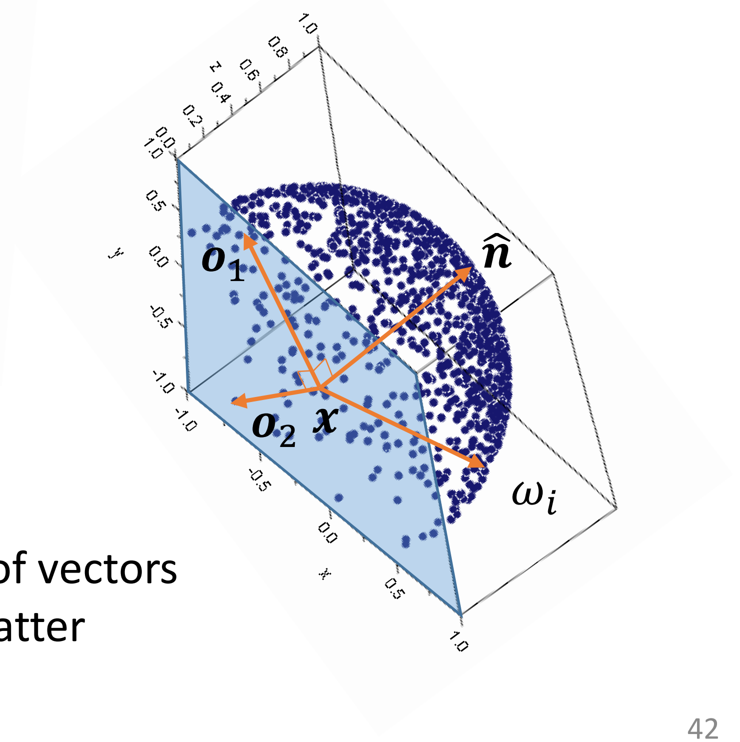
Local Reference Frame

- Hemisphere samples generated in RS must be (at some point in the ray tracing pipeline) transformed to WS

$$\omega_i^{WS} = T_{RS \rightarrow WS} \omega_i^{RS}$$







In the case of isotropic BRDFs, the rotation of vectors $\boldsymbol{o}_1$ and $\boldsymbol{o}_2$ around the normal $\hat{\boldsymbol{n}}$ does not matter

Local Reference Frame

- Derive transformation matrix (RS → WS) from surface normal $\hat{\boldsymbol{n}}$
  - Vector $\hat{\boldsymbol{n}}$ and **any non-parallel** vector $\boldsymbol{a}$ define a plane (we assume that the plane is passing through the origin)
  - This plane has normal $\hat{\boldsymbol{o}}_2$ such that $\hat{\boldsymbol{o}}_2 = \hat{\boldsymbol{n}} \times \boldsymbol{a}$ and by definition, the vector $\hat{\boldsymbol{o}}_2$ is perpendicular to $\hat{\boldsymbol{n}}$. The remaining question is how to construct such a vector $\boldsymbol{a}$?

$\hat{\boldsymbol{o}}_2$ where $\boldsymbol{a} = (0,0,1)$ $\hat{\boldsymbol{o}}_2$ where $\boldsymbol{a} = (1,0,0)$

```
inline Vector3 orthogonal( const Vector3 & n )
{
    return ( abs( n.x ) > abs( n.z ) ) ? Vector3( n.y, -n.x, 0.0f ) : Vector3( 0.0f, n.z, -n.y );
}
```

  - The remaining third axis can be computed as $\hat{\boldsymbol{o}}_1 = \hat{\boldsymbol{o}}_2 \times \hat{\boldsymbol{n}}$ yelding vector perpendicular to both $\hat{\boldsymbol{o}}_2$ and $\hat{\boldsymbol{n}}$
  - Now we can construct a change-of-basis matrix $T_{RS \rightarrow WS}$ that transforms vector in the reference (local) space (RS) to the world space (WS)

Local Reference Frame

$$T_{RS \to WS} = \begin{bmatrix} \vdots & \vdots & \vdots \\ \hat{\boldsymbol{o}}_1 & \hat{\boldsymbol{o}}_2 & \hat{\boldsymbol{n}} \\ \vdots & \vdots & \vdots \end{bmatrix}$$

- Inverse transformation can be computed as follows

$$T_{WS \to RS} = T_{RS \to WS}^{-1}$$

- Moreover, the matrix $T_{RS \to WS}$ belongs to a special orthogonal group $SO(3)$, also called the 3D rotation group (matrices of orthonormal basis) for which holds that $QQ^T = I$ for every $Q \in SO(n)$. Also note that for any nonsingular A: $AA^{-1} = I$
- This property allows us to calculate the inversion of the transformation matrix using simpler (and faster) transposition

$$T_{WS \to RS} = T_{RS \to WS}^{-1} = T_{RS \to WS}^{T}$$

Tangent-Bitangent-Normal

Side note: Texture coordinates are interpolated linearly (barycentric interpolation) across the triangle. Hence, the derivatives are all constant and we can calculate tangents/bitangents per triangle.

- $P_1 - P_0 = \boldsymbol{e}_1 = \Delta u_1 \boldsymbol{t} + \Delta v_1 \boldsymbol{b}$
- $P_2 - P_0 = \boldsymbol{e}_2 = \Delta u_2 \boldsymbol{t} + \Delta v_2 \boldsymbol{b}$
- $\Delta u_1 = P_1^u - P_0^u, \Delta v_1 = P_1^v - P_0^v$
- $\Delta u_2 = P_2^u - P_0^u, \Delta v_2 = P_2^v - P_0^v$

$\boldsymbol{e}_{1,2}$ and $\boldsymbol{t}$, $\boldsymbol{b}$ are 3D row vectors

$P_i^{\{u,v\}}$ are u, resp. v, texture coordinates of i-th vertex

… and we want to solve for $\boldsymbol{t}$ and $\boldsymbol{b}$…

- $\begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \end{bmatrix} = \begin{bmatrix} \Delta u_1 & \Delta v_1 \\ \Delta u_2 & \Delta v_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{t} \\ \boldsymbol{b} \end{bmatrix}$

Transformation matrix $TBN_{TS \to WS} = \begin{pmatrix} \vdots & \vdots & \vdots \\ \hat{\boldsymbol{t}} & \hat{\boldsymbol{b}} & \hat{\boldsymbol{n}} \\ \vdots & \vdots & \vdots \end{pmatrix}$

- $\begin{bmatrix} \boldsymbol{t} \\ \boldsymbol{b} \end{bmatrix} = \begin{bmatrix} \Delta u_1 & \Delta v_1 \\ \Delta u_2 & \Delta v_2 \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \end{bmatrix} = \frac{1}{\Delta u_1 \Delta v_2 - \Delta u_2 \Delta v_1} \begin{bmatrix} \Delta v_2 & -\Delta v_1 \\ -\Delta u_2 & \Delta u_1 \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \end{bmatrix}$

Tangent-Bitangent-Normal

- It is not necessarily true that the tangent vectors $\hat{\boldsymbol{t}}$ and $\widehat{\boldsymbol{b}}$ are perpendicular to each other or to the normal vector $\widehat{\boldsymbol{n}}$
- We may assume that these three vectors will be nearly orthogonal. Use Gram-Schmidt orthogonalization proces to fix that
- To find the tangent vectors for a single vertex, we average the tangents for all triangles sharing that vertex in a manner similar to the way in which vertex normals are commonly calculated. In the case that the neighboring triangles have discontinuous texture mapping, vertices along the border are generally already duplicated since they have different mapping coordinates anyway.

The Gram–Schmidt Process

- The Gram–Schmidt process works as follows

$$
\begin{aligned}
\mathbf{u}_1 &= \mathbf{v}_1, & \mathbf{e}_1 &= \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|} \\
\mathbf{u}_2 &= \mathbf{v}_2 - \mathrm{proj}_{\mathbf{u}_1}(\mathbf{v}_2), & \mathbf{e}_2 &= \frac{\mathbf{u}_2}{\|\mathbf{u}_2\|} \\
\mathbf{u}_3 &= \mathbf{v}_3 - \mathrm{proj}_{\mathbf{u}_1}(\mathbf{v}_3) - \mathrm{proj}_{\mathbf{u}_2}(\mathbf{v}_3), & \mathbf{e}_3 &= \frac{\mathbf{u}_3}{\|\mathbf{u}_3\|} \\
\mathbf{u}_4 &= \mathbf{v}_4 - \mathrm{proj}_{\mathbf{u}_1}(\mathbf{v}_4) - \mathrm{proj}_{\mathbf{u}_2}(\mathbf{v}_4) - \mathrm{proj}_{\mathbf{u}_3}(\mathbf{v}_4), & \mathbf{e}_4 &= \frac{\mathbf{u}_4}{\|\mathbf{u}_4\|} \\
&\ \,\vdots & &\ \,\vdots \\
\mathbf{u}_k &= \mathbf{v}_k - \sum_{j=1}^{k-1} \mathrm{proj}_{\mathbf{u}_j}(\mathbf{v}_k), & \mathbf{e}_k &= \frac{\mathbf{u}_k}{\|\mathbf{u}_k\|}
\end{aligned}
$$

where $\mathrm{proj}_{\hat{\boldsymbol{u}}}(\boldsymbol{v}) = (\boldsymbol{v} \cdot \hat{\boldsymbol{u}})\hat{\boldsymbol{u}}$

Source: https://en.wikipedia.org/wiki/Gram%E2%80%93Schmidt_process

Tangent-Bitangent-Normal

- Using this process, orthogonal (but still unnormalized) tangent vectors $\boldsymbol{t}'$ and $\boldsymbol{b}'$ are given by

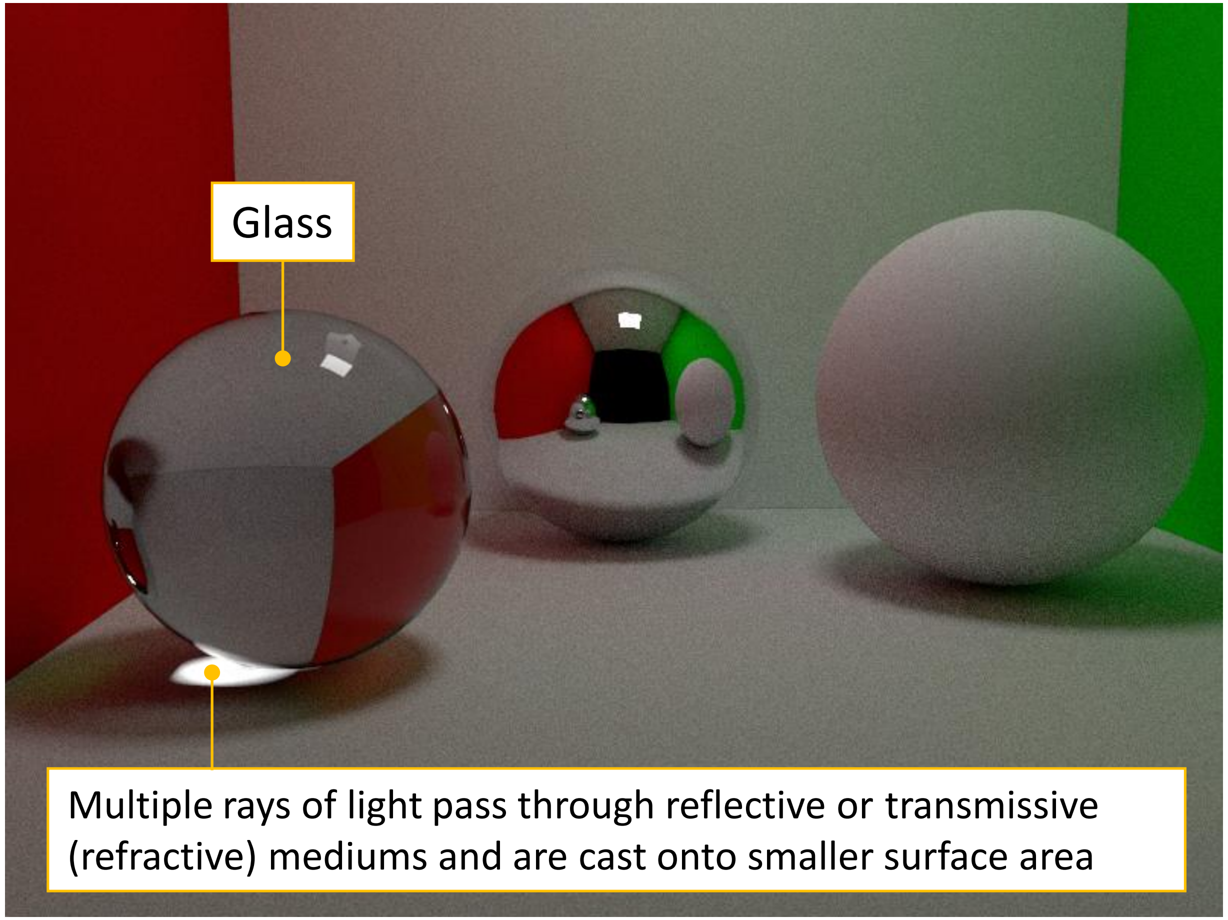
$$\boldsymbol{t}' = \boldsymbol{t} - (\boldsymbol{t} \cdot \hat{\boldsymbol{n}})\hat{\boldsymbol{n}}$$

$$\boldsymbol{b}' = \boldsymbol{b} - (\boldsymbol{b} \cdot \hat{\boldsymbol{n}})\hat{\boldsymbol{n}} - (\boldsymbol{b} \cdot \boldsymbol{t}')\boldsymbol{t}' / \cancel{\boldsymbol{t}'^2}$$

and the new TBN matrix takes the form

$$TBN_{TS \to WS} = \begin{pmatrix} \vdots & \vdots & \vdots \\ \widehat{\boldsymbol{t}'} & \widehat{\boldsymbol{b}'} & \hat{\boldsymbol{n}} \\ \vdots & \vdots & \vdots \end{pmatrix}$$

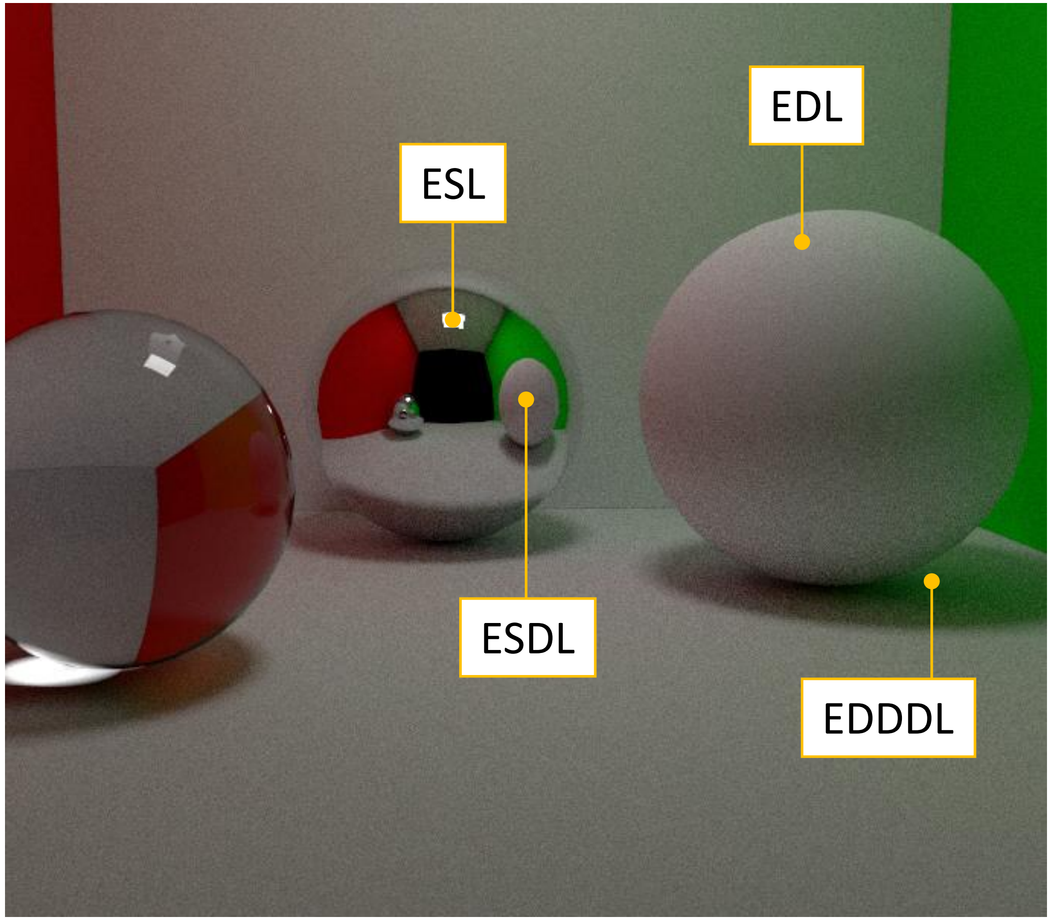
Path Tracing - Caustic



Path Classification

- Whitted-style ray tracer:
  E S* D L
- Path tracer:
  E (S|D)* L

  E: Eye
  S: both reflection and refraction
  D: Diffuse
  L: Light source



* $0 \ldots n$ hits

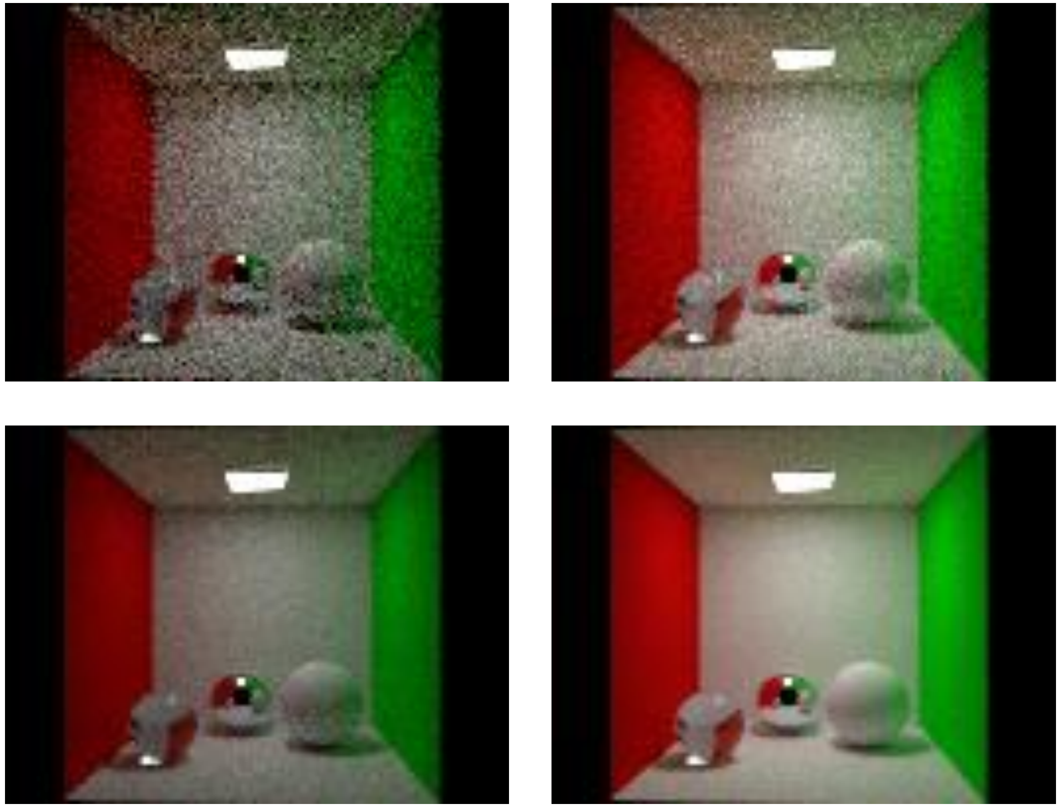
Sampling Strategies

- Uniform sampling (simple but terribly inefficient for specular highlights)
- Cosine distributed sampling (good for diffuse surface)

$$\int_{H(\boldsymbol{x})} L_i(\boldsymbol{x}, \omega_i)\, f_r(\boldsymbol{x}, \omega_i, \omega_0) \cos\theta_i \,\mathrm{d}\omega_i$$

- BRDF proportional sampling (for glossy surfaces)
- Proportional to the incident radiance (usually unknown, can be determined by other techniques, e.g. photon mapping)
- Combination

Area Light Direct Sampling



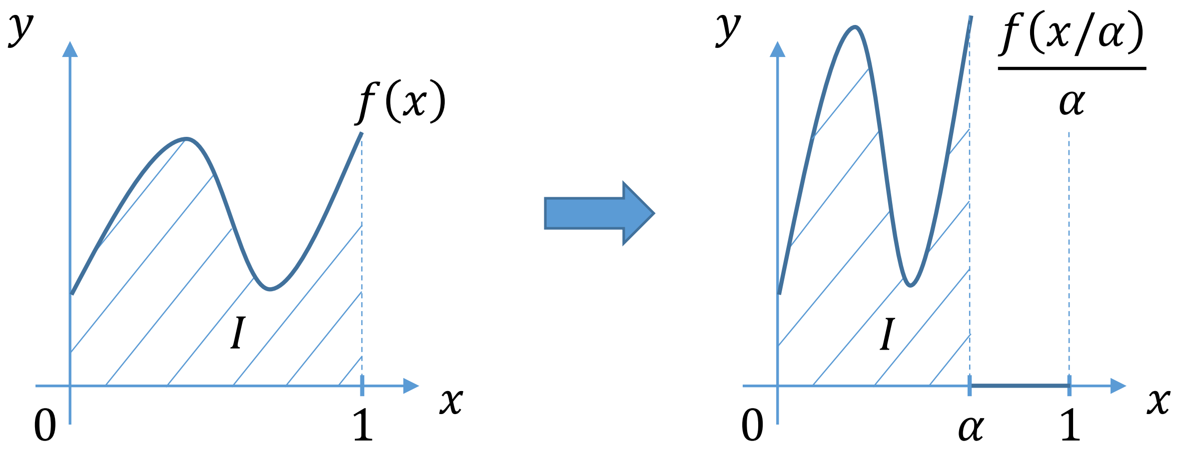
Russian Roulette

- Technique that can be used to terminate infinite recursive algorithms (light bounces around the scene infinitely)
- Reduces the effort spent evaluating unimportant samples that are expensive to evaluate and make a small contribution to the final result (e.g. long light path reaching the light source after many energy dissipating bounces)
- O1: Termination after fixed number of bounces is biased
- O2: With probabilistic termination, we avoid infinite paths without bias and the result will be unbiased (i.e. no error is introduced by this step)

Russian Roulette

- Pick any value $\alpha \in (0,1)$, $1-\alpha$ is absorption probability
- Keep the same PDF and sampling of „squeezed“ function f

$$I = \int_0^1 f(x)\mathrm{d}x = \int_0^\alpha \frac{f(x/\alpha)}{\alpha}\mathrm{d}x$$



$$I \approx \frac{1}{N}\sum_{i=0}^{N-1}\frac{f(x_i)}{p(x_i)} \approx \frac{1}{N}\sum_{i=0}^{N-1}\frac{f(x_i/\alpha)}{\alpha\, p(x_i)}$$

Russian Roulette - Example

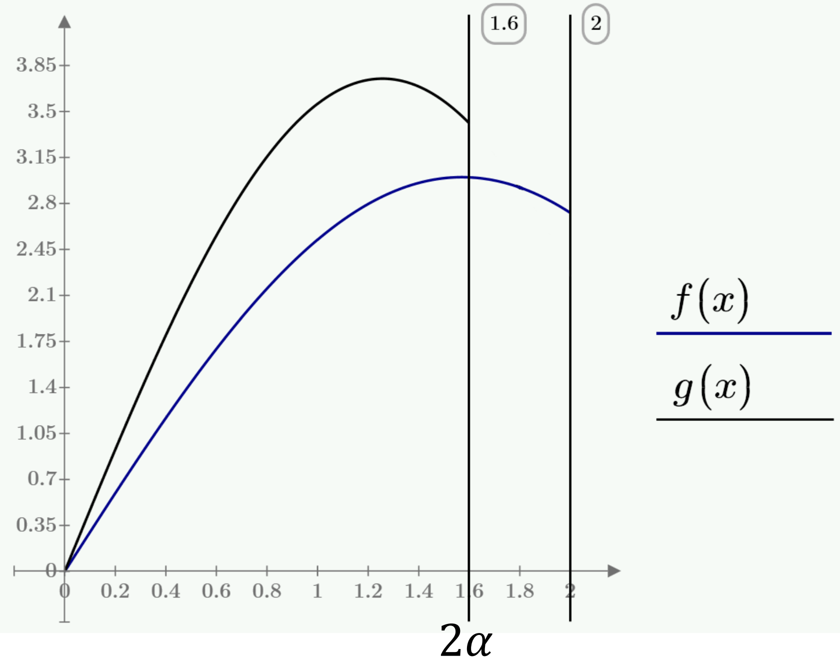
- $I_f = \int_0^2 f(x)\mathrm{d}x = [-3\cos(x)]_0^2 \approx 4.248$

- $g(x) = \dfrac{f\left(\frac{x}{\alpha}\right)}{\alpha} = \dfrac{3\sin\left(\frac{x}{\alpha}\right)}{\alpha}$
- $I_g = \int_0^{2\alpha} g(x)\mathrm{d}x = \left[-3\cos\left(\frac{x}{\alpha}\right)\right]_0^{2\alpha} \approx 4.248$

- $I_f = I_g$

$f(x) := 3\sin(x)$

$g(x) := \dfrac{f\left(\frac{x}{\alpha}\right)}{\alpha}$ where $\alpha = 0.8$



Russian Roulette - Example

```
float I = 0.0f;


for ( int i = 0; i < N; ++i ) {
  float x_i = random->next_float( tid, 0.0f, 2.0f );
  float pdf_x = 1.0f / 2.0f;


  float f_i = 3.0f * sinf( x_i );
  I += f_i / pdf_x;
}


I /= N;

I → 4.248170
```

MC without RR

```
float I = 0.0f;
float a = 0.8f;

for ( int i = 0; i < N; ++i ) {
  float x_i = random->next_float( tid, 0.0f, 2.0f );
  float pdf_x = 1.0f / 2.0f;

  if ( x_i < 2.0f * a ) {
    float f_i = 3.0f * sinf( x_i / a );
    I += f_i / ( a * pdf_x );
  } else {
    // I += 0.0f;
  }
}

I /= N;

I → 4.248834
```

MC with RR on full range $\langle 0,2)$

```
float I = 0.0f;
float a = 0.8f;

for ( int i = 0; i < N; ++i ) {
  float x_i = random->next_float( tid, 0.0f, 2 * a );
  float pdf_x = 1.0f / ( 2.0f * a );


  float f_i = 3.0f * sinf( x_i / a );
  I += f_i / ( a * pdf_x );
}


I /= N;

I → 4.248680
```

MC on restricted range $\langle 0,2\alpha)$

Russian Roulette

- Pick any value $\alpha \in (0,1)$ and uniform random number $\xi \in \langle 0,1)$
- In PT, α represents the prob. of non-terminated path ($1-\alpha$ is absorption probability) and can be deduced from diffuse or specular coefficient (or any other representant of path throughput)
- Replace the original function/estimator with the following one
- $\mathrm{g}(x) = \begin{cases} \frac{1}{\alpha} f(x), & \text{with prob. } \alpha \text{ (i.e. if } \alpha > \xi\text{)} \\ 0, & \text{otherwise (i.e. with prob. } 1-\alpha \text{ or if } \alpha \le \xi\text{)} \end{cases}$

non-terminated path

terminated path

- $\mathrm{E}[\mathrm{g}(x)] = \alpha\left(\frac{1}{\alpha} f(x)\right) + (1-\alpha)0 = f(x)$

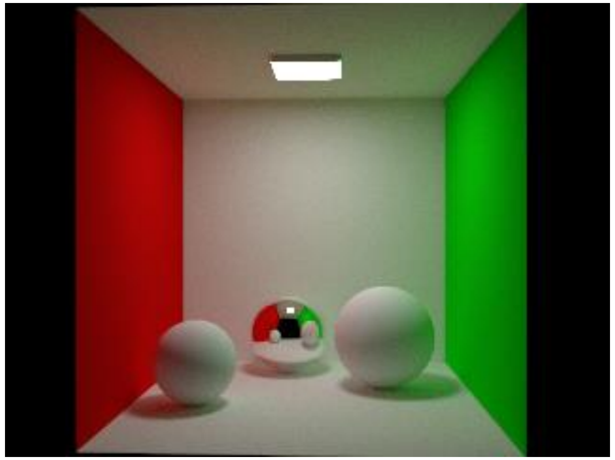
Does Russian Roulette really provide an unbiased result? Yes, it does!

Russian Roulette

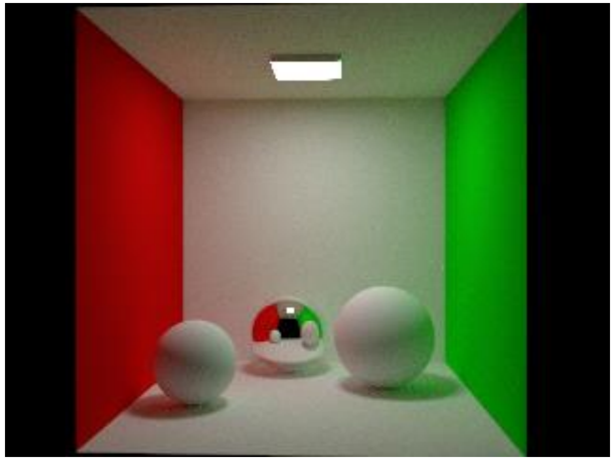
```
trace_ray(ray, depth):
    α = Albedo // or anything else representing the throughput of the entire light path
    if ( α <= random ) or ( depth > 100 ) return 0 // expl. depth is termination guarantee
    p := find_closest_intersection(ray, scene)
    if ( no hit ) return background
    L_e := get_emmision(p, omega_o)    // note that omega_o = -ray.dir
    if L_e ≠ 0: return L_e             // we hit a source and stopped our light path here
    omega_i, pdf := sample_hemisphere(normal)
    L_i := trace_ray(make_secondary_ray(p, omega_i), depth + 1)
    f_r := Albedo / π                  // e.g. Lambert BRDF
    L_r := L_i * f_r * (omega_i · normal) / ( pdf * α ) // sample is weighted by prob. α
    return L_r
```

How do we compensate for energy losses caused by terminating low throughput paths? We boost the energy of the non-terminated paths by their probability.

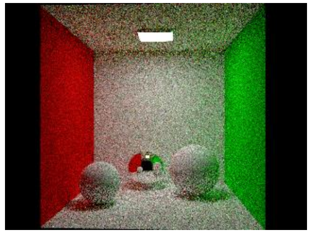
Russian Roulette



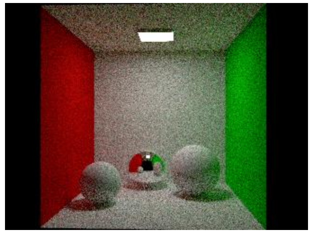
No RR and max depth = 50



With RR (ρ = min(Albedo.max(), 0.95)) and max depth = 50



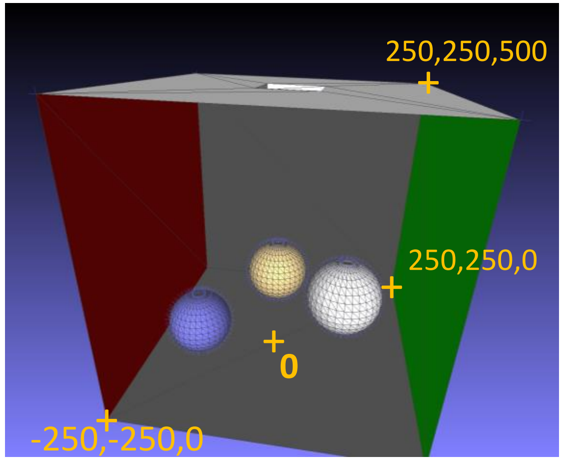
Samples weighted by ρ
(image has correct brightness)

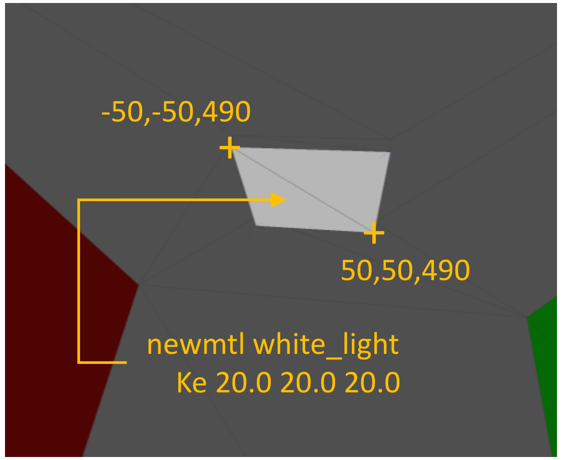


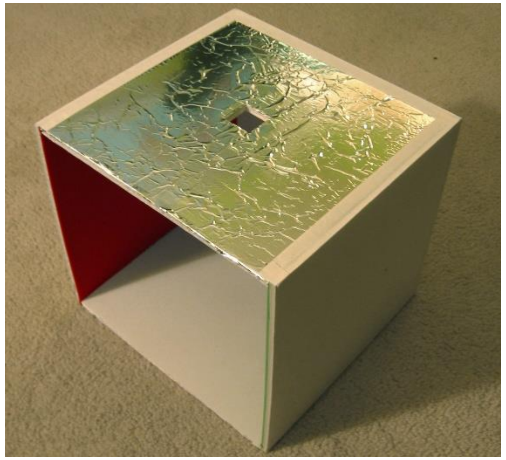
Weighting by ρ omited
(image is darker)

Cornell Box

- Traditional test scene used for the confirmation of the accuracy of light transport simulations
- http://www.graphics.cornell.edu/online/box/data.html





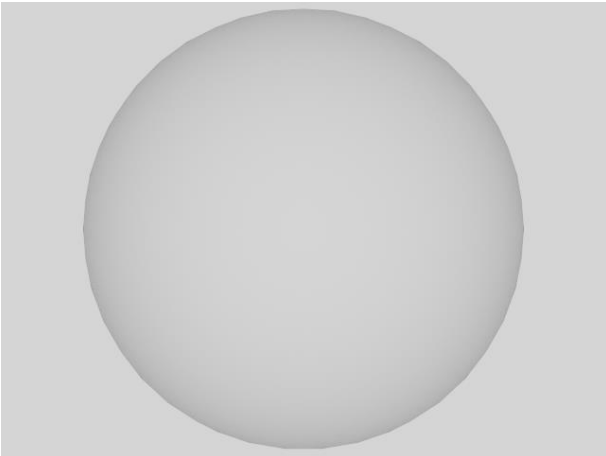


Source: http://hatchstudios.com/work/cornell-box-physical-model/

Furnace Test

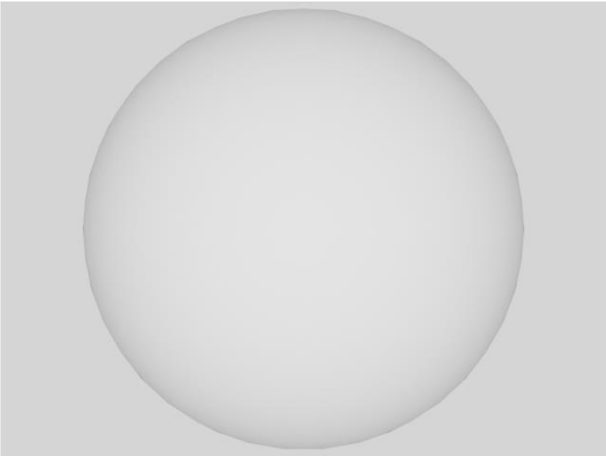
The name comes from the property of the furnace in thermal equilibrium. When the furnace reaches equilibrium (when the amount of energy received at a given point equals the amount of energy radiated), the interior of the furnace has a uniform appearance so that all geometric features disappear. Source: https://www.scratchapixel.com

- Test scene: white sphere (or any other surface) surrounded by white environment
- If the sphere is supposed to reflect 100 % of radiance (no matter how) coming from all directions we should see the same amount of radiance (i.e. 100 % of background radiance) reflected from each point of the sphere



wrong result
(surface is too dim)

correct result
(surface disappear)



wrong result
(surface is too bright)

Probability

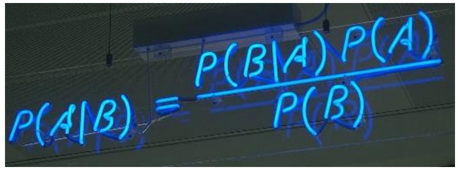
- Joint prob. $P(A, B) = P(A \wedge B) = P(A \text{ and } B) = \overbrace{P(A|B)P(B) = \underbrace{P(B|A)}_{\substack{\text{conditional prob.} \\ (| = \text{given})}}P(A)}^{\text{symmetrical}} =$ iff A and B are independent $= P(B)\underbrace{P(A)}_{\substack{\text{prob. of event } A \\ P(x = A)}} = P(A)P(B)$

$P(A|B) \neq P(B|A)$ (not symmetrical)

- $P(x)$ is prob. density of x (a random variable)
- Bayes' theorem $\boxed{P(A|B) = \frac{P(B|A)P(A)}{P(B)}} = \frac{P(A,B)}{P(B)} = \frac{P(A|B)\cancel{P(B)}}{\cancel{P(B)}}$

Note that $P(B)$ must be nonzero

- Marginal prob. (no special notation) $P(A) = \underbrace{\sum_i P(A, y = B_i)}_{\text{sum rule}}$



The marginal (simple) prob. is different from the conditional prob. because it considers the union of all events for the second variable rather than the probability of a single event.

$$P(A3, B2) \stackrel{\text{def}}{=} P(B2|A3)P(A3) = 0.581 \cdot 0.395 = 0.229$$

$$\|$$

$$P(A3, B2) \stackrel{\text{def}}{=} P(A3|B2)P(B2) = 0.375 \cdot 0.611 = 0.229$$

Probability

Two dependent random variables

Histogram	$A1$	$A2$	$A3$	
$B1$	**30**	**5**	**26**	61
$B2$	**20**	**40**	**36**	96
	50	45	62	157

1. Probability dist./mass function

Joint probs.

$P(Ai, Bj)$	$A1$	$A2$	$A3$	$P(Bj)$
$B1$	0.191	0.032	0.166	0.389
$B2$	0.127	0.255	0.229	0.611
$P(Ai)$	0.318	0.287	0.395	1

2. Marginal dist. of r. v. y

Marginal (simple) probs.

Marginal dist. of r. v. x

$P(Ai\|Bj)$	$A1$	$A2$	$A3$	
$B1$	0.492	0.082	0.426	1
$B2$	0.208	0.417	0.375	1

$P(Bi\|Aj)$	$A1$	$A2$	$A3$
$B1$	0.600	0.111	0.419
$B2$	0.400	0.889	0.581
	1	1	1

3. Conditional probs.

$$P(A1|B1) = \frac{P(A1, B1)}{P(B1)} = \frac{0.191}{0.389} = 0.492$$

$$P(B1|A1) = \frac{P(B1, A1)}{P(A1)} = \frac{0.191}{0.318} = 0.600$$

Bayes' theorem $P(A1|B1) = \frac{P(B1|A1)P(A1)}{P(B1)}$ $= \frac{0.600 \cdot 0.318}{0.389} = 0.492 = \frac{0.191}{0.389} = \frac{P(A1,B1)}{P(B1)}$

Note that $P(A1, B1) = 0.191 \neq 0.124 = 0.318 \cdot 0.389 = P(A1)P(B1)$ because r. v. x and r. v. y are not independent!

$P(A3,B2) \stackrel{\text{def}}{=} P(B2|A3)P(A3) = 0.611 \cdot 0.395 = 0.241$

$\shortparallel$

$P(A3,B2) \stackrel{\text{def}}{=} P(A3|B2)P(B2) = 0.395 \cdot 0.611 = 0.241$

Probability

Two independent random variables

Histogram	$A1$	$A2$	$A3$	
$B1$	**19**	**17**	**24**	61
$B2$	**31**	**28**	**38**	96
	50	45	62	157

1. Probability dist./mass function

Joint probs.

$P(Ai,Bj)$	$A1$	$A2$	$A3$	$P(Bj)$
$B1$	0.124	0.111	0.153	0.389
$B2$	0.195	0.175	0.241	0.611
$P(Ai)$	0.318	0.287	0.395	1

2. Marginal dist. of r. v. y

Marginal (simple) probs.

Marginal dist. of r. v. x

Intuitively, two r. v. are independent if knowing the value of one of them does not change the probabilities for the other one

$P(Ai\|Bj)$	$A1$	$A2$	$A3$	
$B1$	0.318	0.287	0.395	1
$B2$	0.318	0.287	0.395	1

$P(Bi\|Aj)$	$A1$	$A2$	$A3$
$B1$	0.389	0.389	0.389
$B2$	0.611	0.611	0.611
	1	1	1

3. Conditional probs.

$$P(A1|B1) = \frac{P(A1,B1)}{P(B1)} = \frac{0.124}{0.389} = 0.318$$

$$P(B1|A1) = \frac{P(B1,A1)}{P(A1)} = \frac{0.124}{0.318} = 0.389$$

Bayes' theorem $P(A1|B1) = \frac{P(B1|A1)P(A1)}{P(B1)} = \frac{0.389 \cdot 0.318}{0.389} = 0.318 = \frac{0.124}{0.389} = \frac{P(A1,B1)}{P(B1)}$

Note that $P(A1,B1) = 0.124 = 0.124 = 0.318 \cdot 0.389 = P(A1)P(B1)$ because r. v. x and r. v. y are independent!

Probability

- A natural question that arises here is what makes two variables dependent or independent. The answer is quite straightforward – its all about the contingency between those two variables

Histogram	$A1$	$A2$	$A3$	
$B1$	$\mathbf{30}$	$\mathbf{5}$	$\mathbf{26}$	61
$B2$	$\mathbf{20}$	$\mathbf{40}$	$\mathbf{36}$	96
	50	45	62	157

Histogram	$A1$	$A2$	$A3$	
$B1$	$\mathbf{19}$	$\mathbf{17}$	$\mathbf{24}$	61
$B2$	$\mathbf{31}$	$\mathbf{28}$	$\mathbf{38}$	96
	50	45	62	157

- If the proportions of random variables in the different columns vary significantly between rows (or vice versa), we say that there is a contingency between the two variables and the variables are dependent. If there is no contingency, we say that the variables are independent.

Probability

- A natural question that arises here is what makes two variables dependent or independent. The answer is quite straightforward – its all about the contingency between those two variables

Two dependent random variables

$30/50 \neq 5/45 \neq 26/62$
$20/50 \neq 40/45 \neq 36/62$

$30/61 \neq 20/96$
$5/61 \neq 40/96$
$26/61 \neq 36/96$

Histogram	$A1$	$A2$	$A3$	
$B1$	$\mathbf{30}$	$\mathbf{5}$	$\mathbf{26}$	61
$B2$	$\mathbf{20}$	$\mathbf{40}$	$\mathbf{36}$	96
	50	45	62	157

Two independent random variables

Histogram	$A1$	$A2$	$A3$	
$B1$	$\mathbf{19}$	$\mathbf{17}$	$\mathbf{24}$	61
$B2$	$\mathbf{31}$	$\mathbf{28}$	$\mathbf{38}$	96
	50	45	62	157

$19/50 \approx 17/45 \approx 24/62$
$31/50 \approx 28/45 \approx 38/62$

$19/61 \approx 31/96$
$17/61 \approx 28/96$
$24/61 \approx 38/96$

- If the proportions of random variables in the different columns vary significantly between rows (or vice versa), we say that there is a contingency between the two variables and the variables are dependent. If there is no contingency, we say that the variables are independent.

Probability

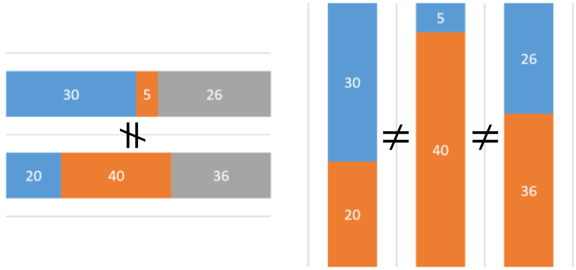
- The distribution of the random variable x is (or is not) affected by the values of the random variable y (and vice versa)

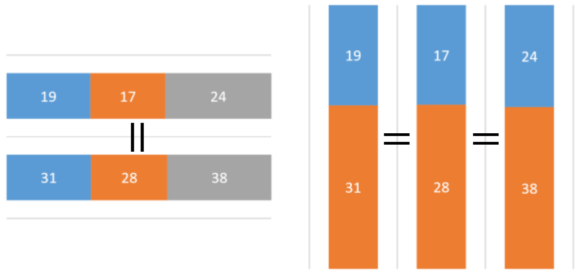
Two dependent random variables

Histogram	$A1$	$A2$	$A3$	
$B1$	**30**	**5**	**26**	61
$B2$	**20**	**40**	**36**	96
	50	45	62	157

Two independent random variables

Histogram	$A1$	$A2$	$A3$	
$B1$	**19**	**17**	**24**	61
$B2$	**31**	**28**	**38**	96
	50	45	62	157





Environment Map Importance Sampling

- In general, sampling an environment map is equivalent to sampling piecewise-constant 2D functions which has foundation in sampling piecewise-constant 1D functions
- Transformation from image to spherical coordinates

$$T(i,j) = \begin{pmatrix} \varphi = \frac{2\pi i}{w} \\ \theta = \frac{\pi j}{h} \end{pmatrix} \text{where } J = \begin{pmatrix} \frac{\partial\varphi}{\partial i} & \frac{\partial\varphi}{\partial j} \\ \frac{\partial\theta}{\partial i} & \frac{\partial\theta}{\partial j} \end{pmatrix} = \begin{pmatrix} \frac{2\pi}{w} & 0 \\ 0 & \frac{\pi}{h} \end{pmatrix} \Rightarrow \det J = \frac{2\pi^2}{w\,h}$$

- Transformation from spherical to cartesian coordinates

$$T(\varphi,\theta) = \begin{pmatrix} x = r\cos(\varphi)\sin(\theta) \\ y = r\sin(\varphi)\sin(\theta) \\ z = r\cos(\theta) \end{pmatrix} \text{where } J = \begin{pmatrix} \frac{\partial x}{\partial\varphi} & \frac{\partial x}{\partial\theta} & \frac{\partial x}{\partial r} \\ \frac{\partial y}{\partial\varphi} & \frac{\partial y}{\partial\theta} & \frac{\partial y}{\partial r} \\ \frac{\partial z}{\partial\varphi} & \frac{\partial z}{\partial\theta} & \frac{\partial z}{\partial r} \end{pmatrix} =$$

$$\begin{pmatrix} -r\sin(\varphi)\sin(\theta) & r\cos(\varphi)\cos(\theta) & \cos(\varphi)\sin(\theta) \\ r\cos(\varphi)\sin(\theta) & r\sin(\varphi)\cos(\theta) & \sin(\varphi)\sin(\theta) \\ 0 & -r\sin(\theta) & \cos(\theta) \end{pmatrix} \Rightarrow \det J = r^2\sin(\theta)$$

Environment Map Importance Sampling

- $\int_{\Omega} pdf(\omega)\mathrm{d}\omega = \int_{\varphi} \int_{\theta} pdf(\varphi, \theta)\sin(\theta)\mathrm{d}\theta\mathrm{d}\varphi = 1$
- $\int_{S^2} \mathrm{d}\omega = 4\pi = \int_{\varphi} \int_{\theta} \sin(\theta)\mathrm{d}\theta\mathrm{d}\varphi = (\cos(\theta_0) - \cos(\theta_1))(\varphi_1 - \varphi_0) = \mathrm{d}A$