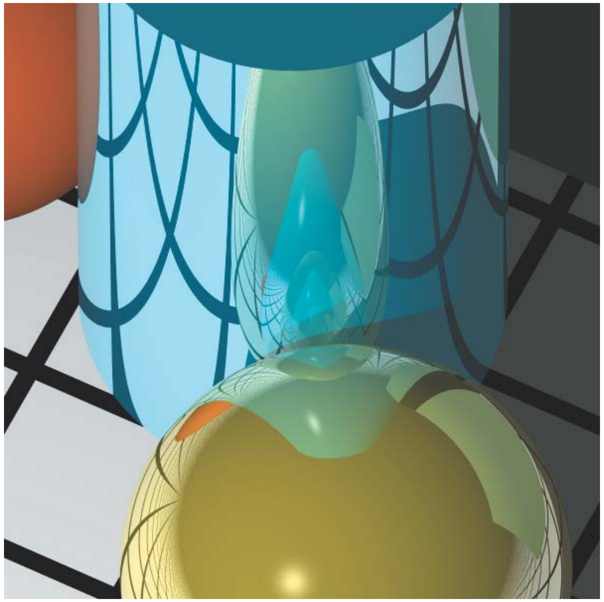
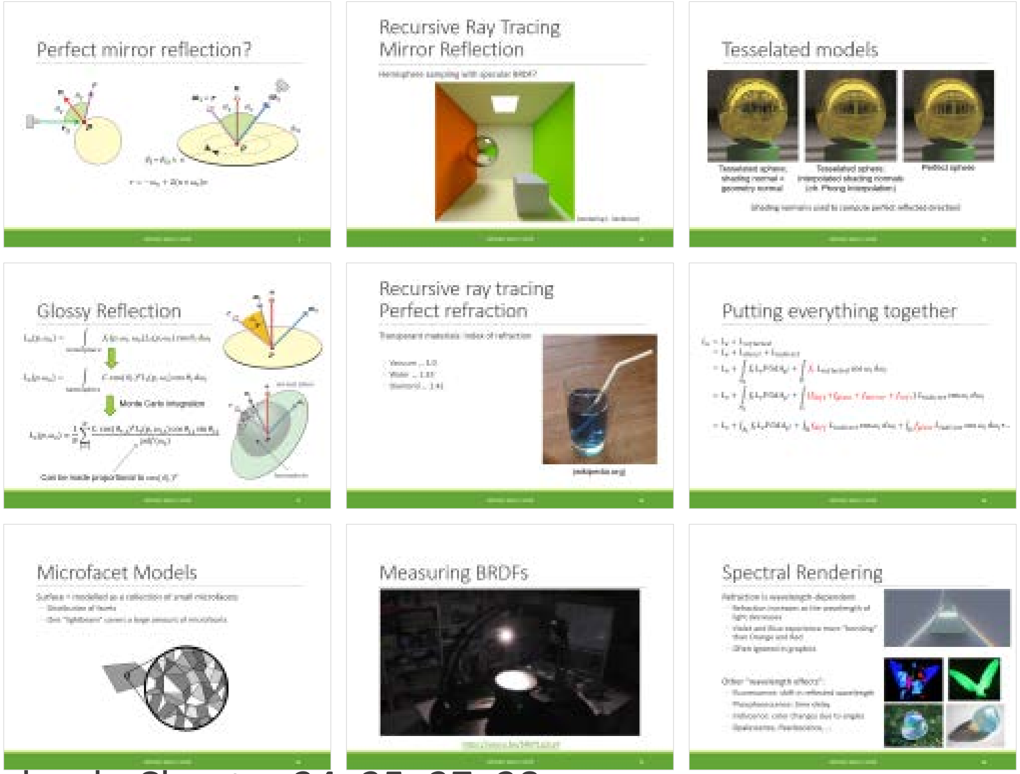
Advanced Visual Effects



FUNDAMENTALS OF COMPUTER GRAPHICS
PHILIP DUTRÉ
DEPARTMENT OF COMPUTER SCIENCE

Overview Lecture

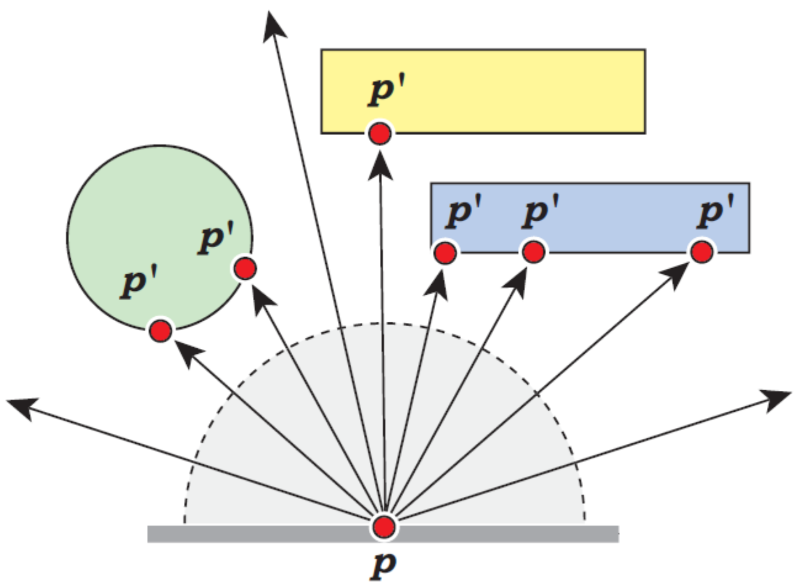


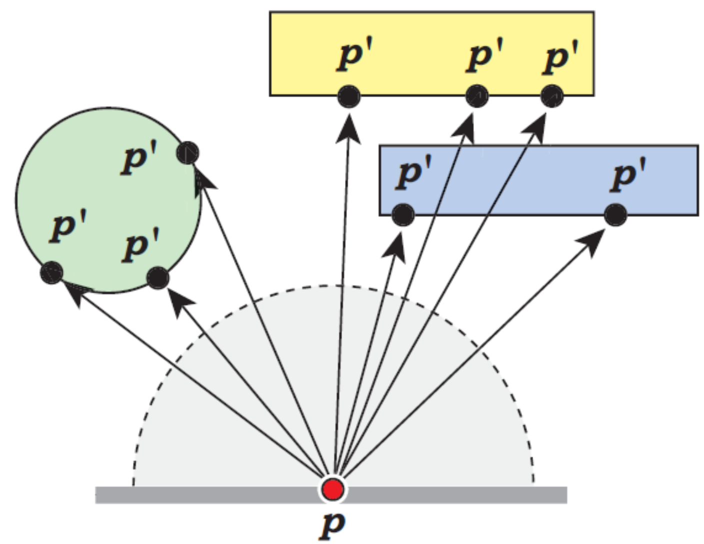
Relevant sections in book: Chapter 24, 25, 27, 28
(Illustrations from *Ray Tracing From The Ground Up*, *Physically-Based Rendering*, *Fundamentals of Computer Graphics*)

Rendering Equation

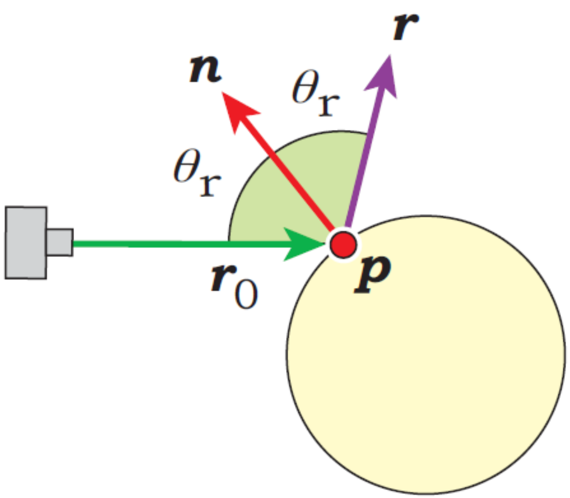
$$L_o(p,\omega_o) = L_e(p,\omega_o) + \int_{hemisphere} f_r(p,\omega_i,\omega_o)L_i(p,\omega_i)\cos\theta_i\, d\omega_i$$

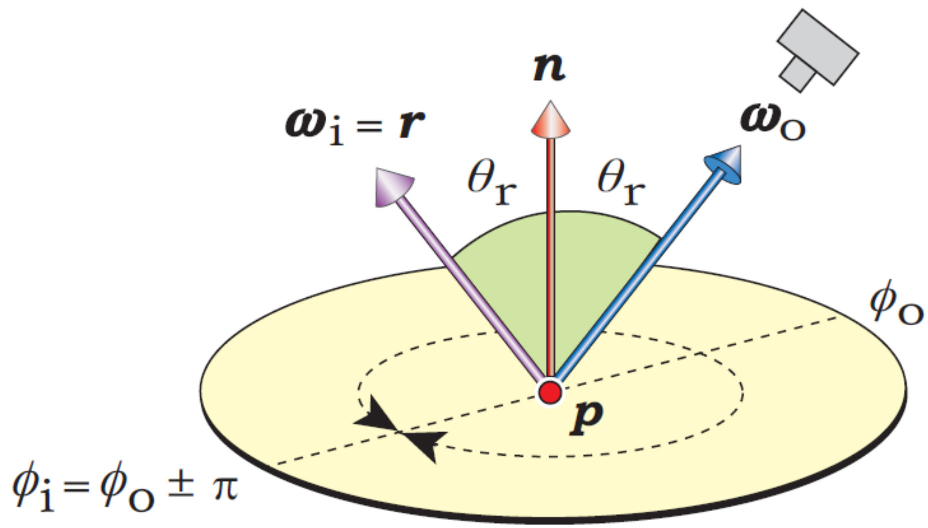
$$L_o(p,\omega_o) = L_e(p,\omega_o) + \int_A f_r(p,\omega_i,\omega_o)L_o(p',-\omega_i)V(p,p')G(p,p')dA_{p'}$$





Perfect mirror reflection?



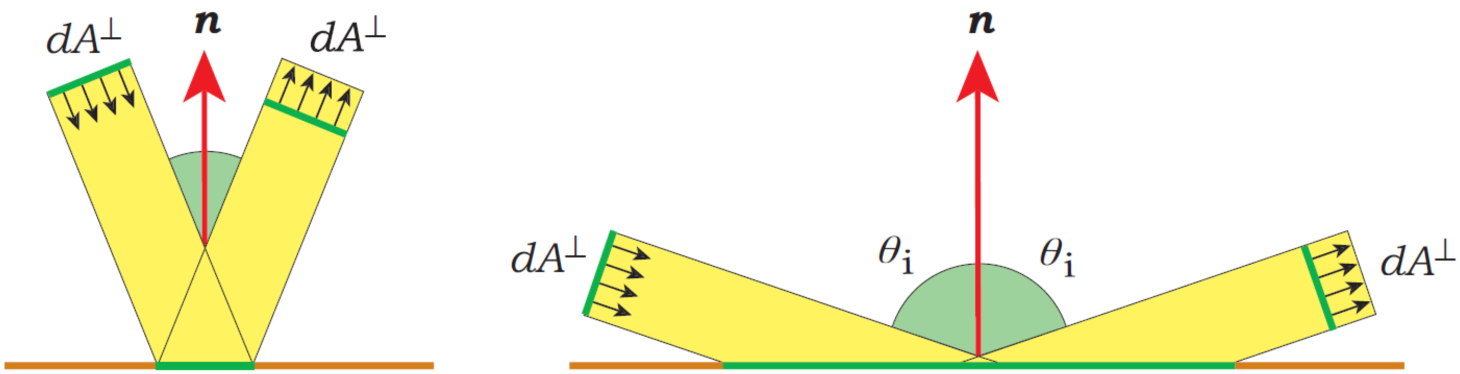


$$r = -\omega_o + 2(n \bullet \omega_o)n$$

Perfect mirror reflection?

Incoming radiance is reflected into outgoing radiance, only at the perfect mirror direction

$$L_o(p,\omega_o) = \int\limits_{hemisphere} f_r(p,\omega_i,\omega_o) L_i(p,\omega_i)\cos\theta_i \, d\omega_i$$
$$= f_{specular}(p,\omega_i,\omega_o) L_i(p,\omega_i)$$



How much light does a material reflect?

- Constant factor (approximation)
- Fresnel reflectance

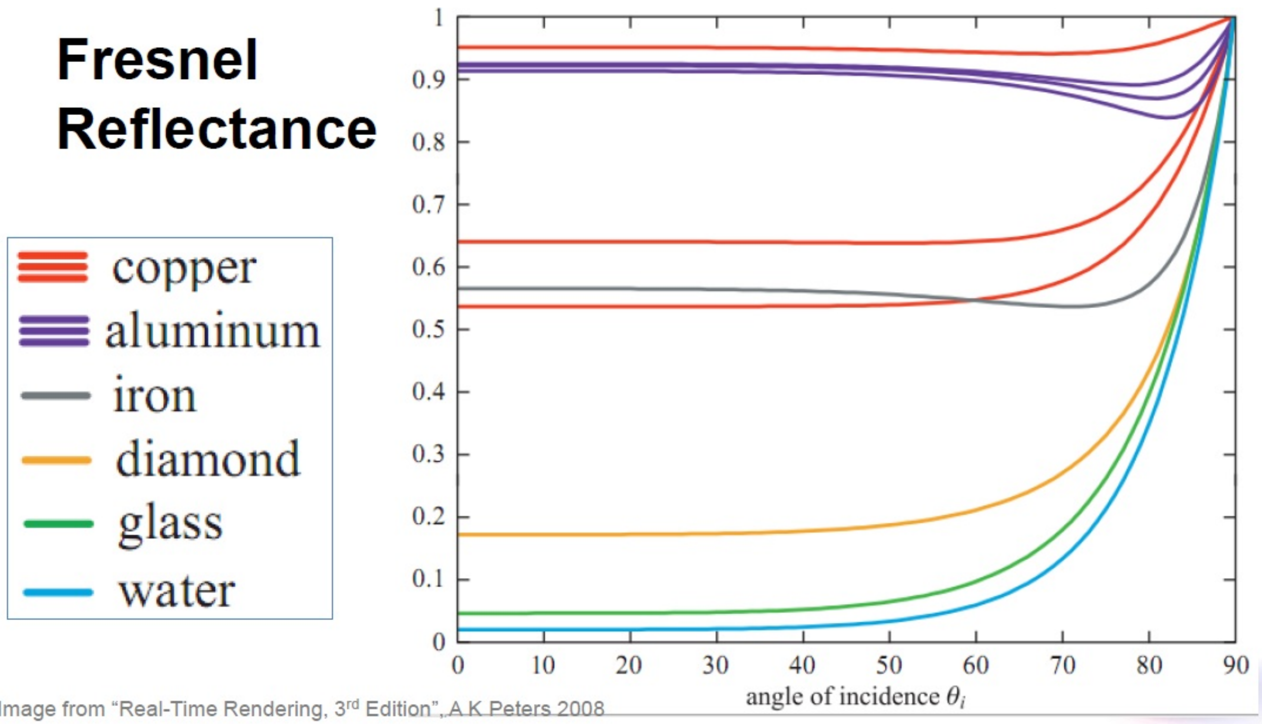


Image from “Real-Time Rendering, 3rd Edition”, A K Peters 2008

How much light does a material reflect?

Fresnel Reflectance

$$r_{\parallel} = \frac{\eta_t \cos\theta_i - \eta_i \cos\theta_t}{\eta_t \cos\theta_i + \eta_i \cos\theta_t}$$

$$r_{\perp} = \frac{\eta_i \cos\theta_i - \eta_t \cos\theta_t}{\eta_i \cos\theta_i + \eta_t \cos\theta_t},$$

$$F_r = \frac{1}{2}(r_{\parallel}^2 + r_{\perp}^2).$$

Table 8.1: Indices of refraction for a variety of objects, giving the ratio of the speed of light in a vacuum to the speed of light in the medium. These are generally wavelength-dependent quantities; these values are averages over the visible wavelengths.

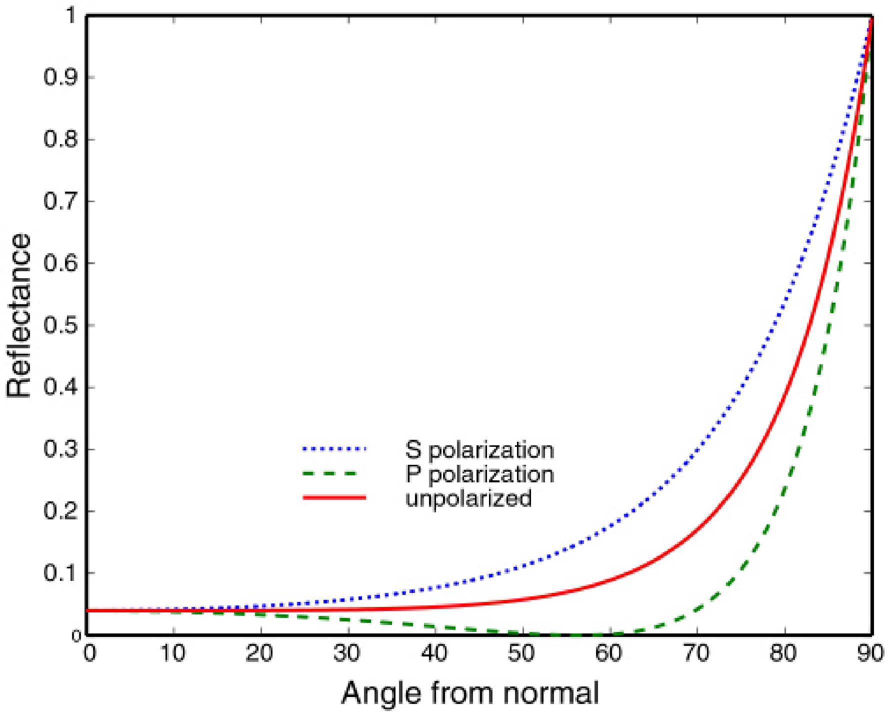
Medium	**Index of refraction** η
Vacuum	1.0
Air at sea level	1.00029
Ice	1.31
Water (20°C)	1.333
Fused quartz	1.46
Glass	1.5–1.6
Sapphire	1.77
Diamond	2.42

How much light does a material reflect?

Approximation for Fresnel Reflectance:

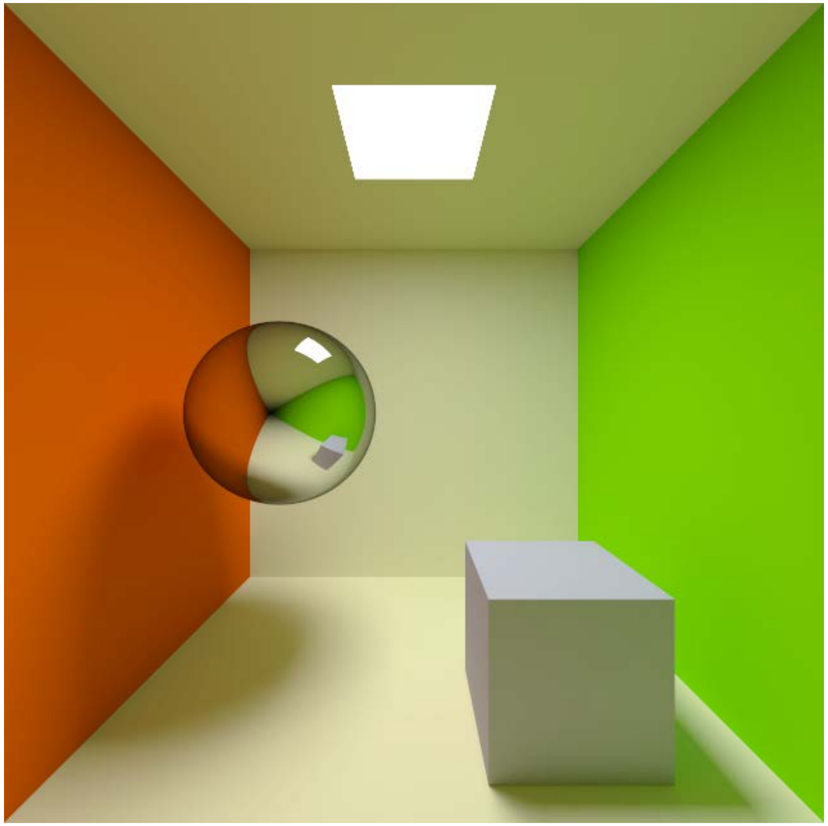
$$R(\theta) = R_0 + (1 - R_0)(1 - \cos\theta)^5$$

$$R_0 = \left(\frac{n_t - 1}{n_t + 1}\right)^2$$



Recursive Ray Tracing Mirror Reflection

Hemisphere sampling with specular BRDF?

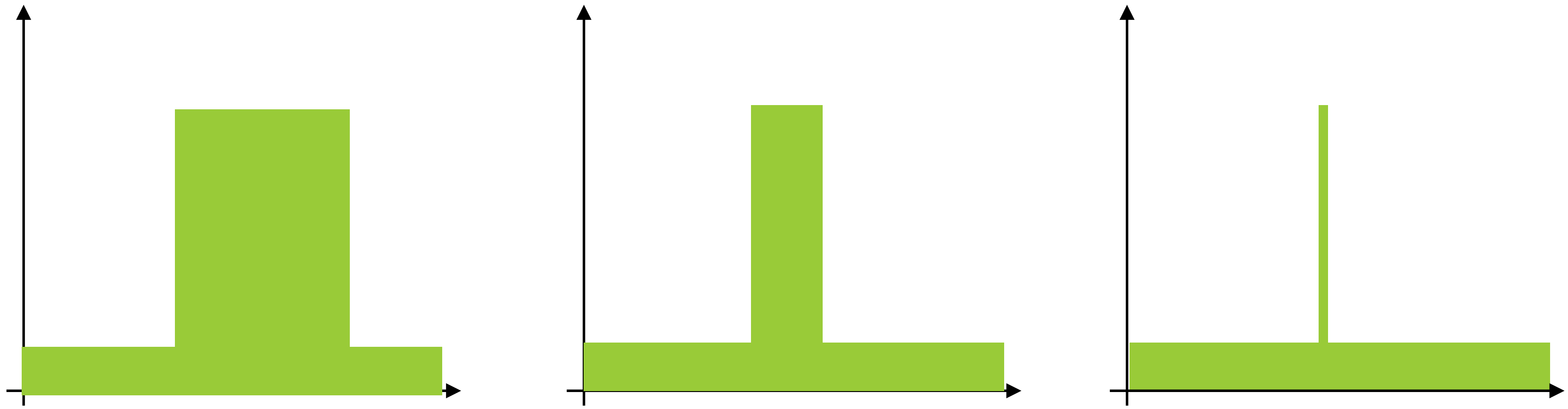


(rendering L. Vanbesien)

Recursive Ray Tracing Mirror Reflection

Can we compute perfect mirror reflection using Monte Carlo ray tracing?

- … generate random rays over the hemisphere?
- … how to compute an integral (using Monte Carlo integration) of a function with a very sharp peak at a (known) location?



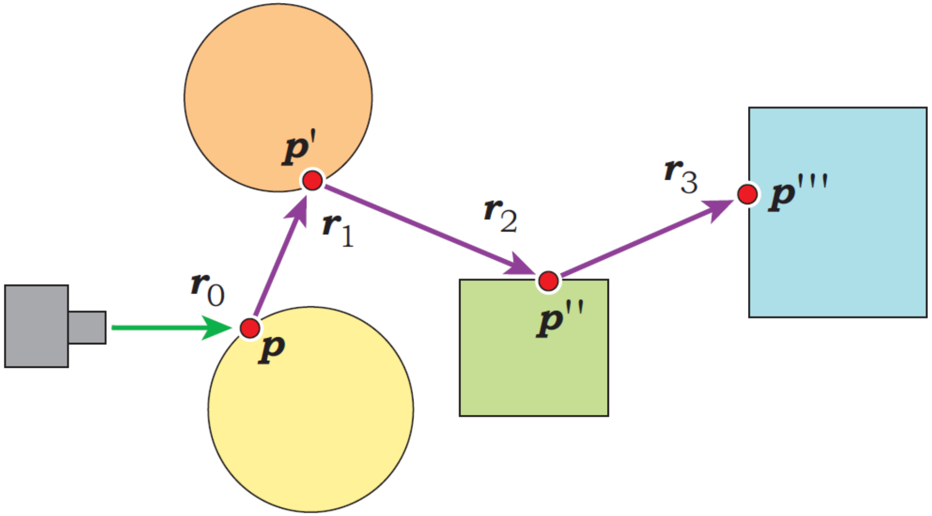
Recursive Ray Tracing Mirror Reflection

Special case of indirect illumination

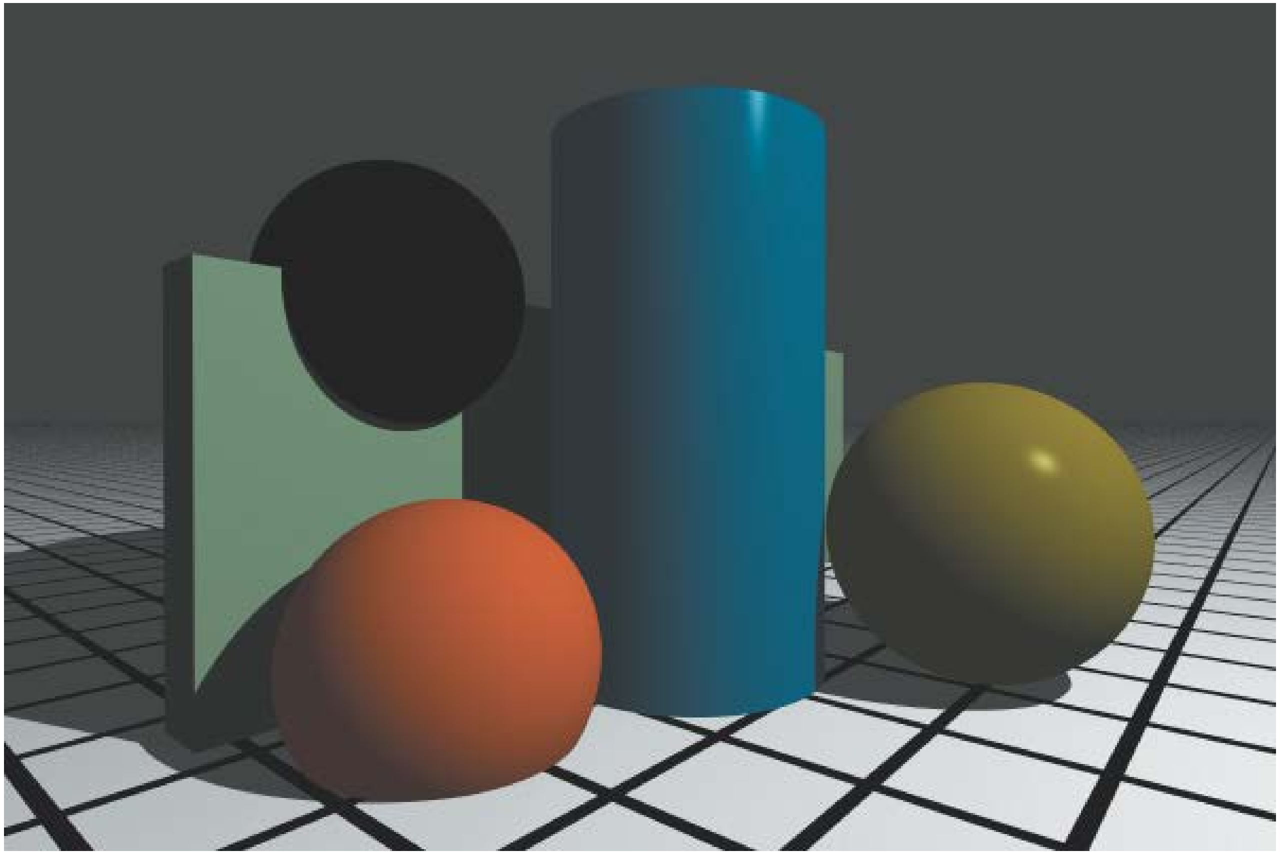
- Trace (only) the perfect mirrored reflection

How to control recursion?

- Maximum depth
- Product of successive BRDFs along path < threshold
- Russian Rou

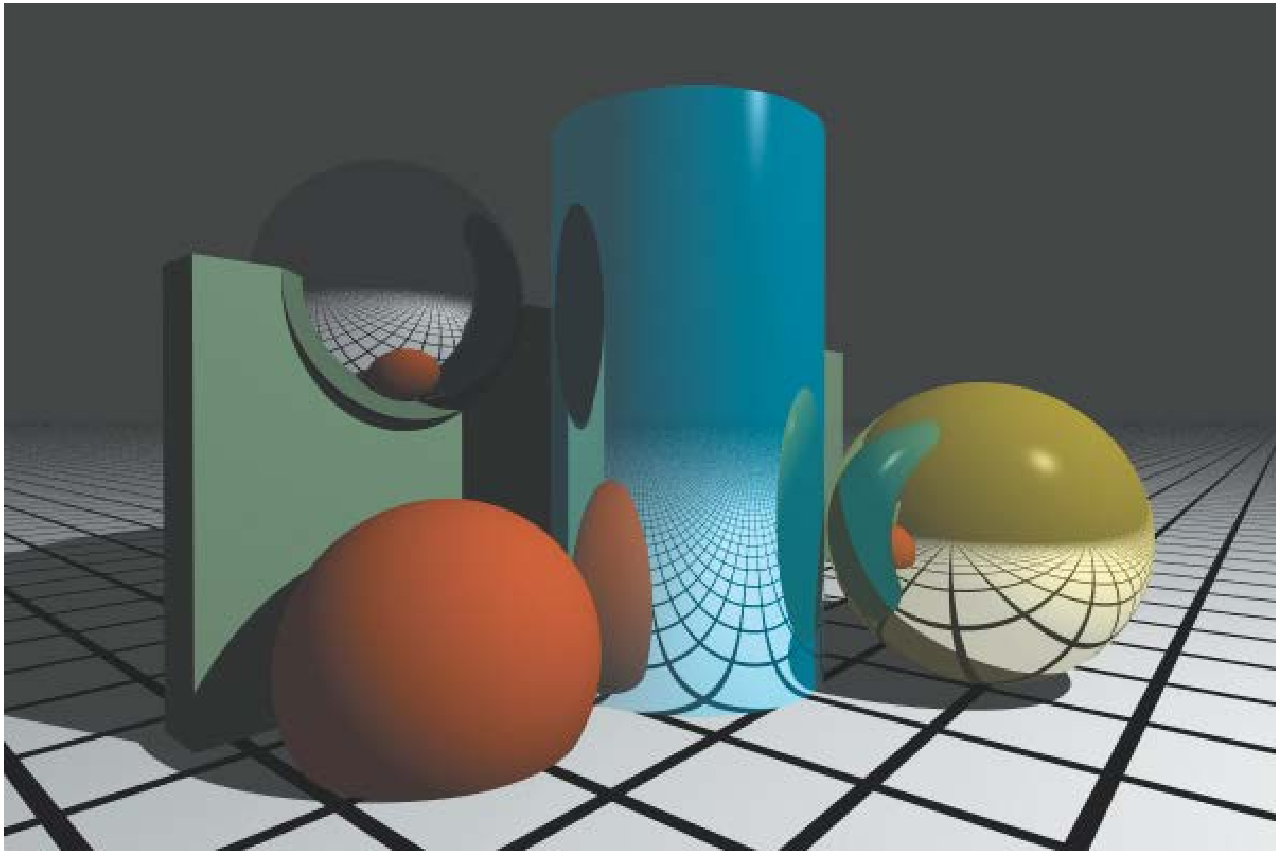


Recursive Ray Tracing
Mirror Reflection



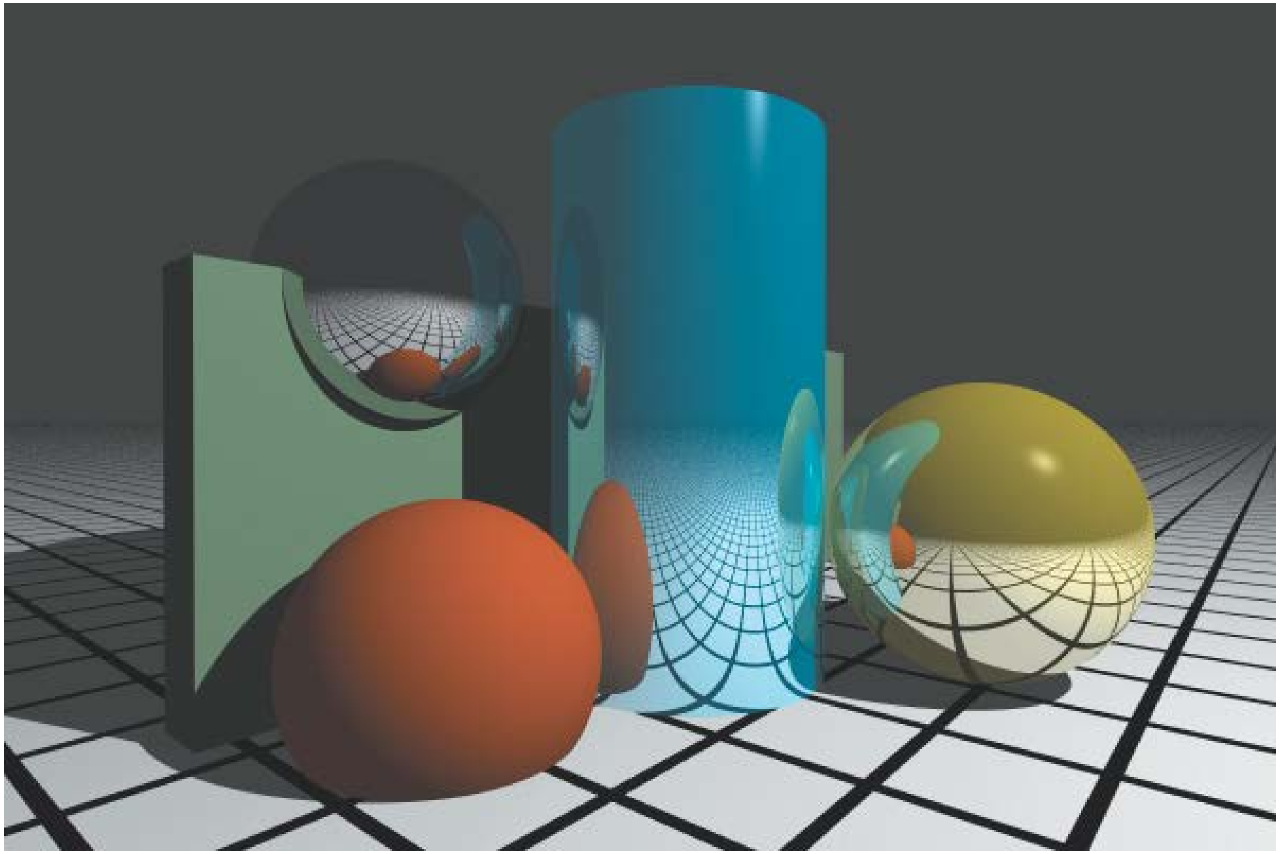
Max depth = 0

Recursive Ray Tracing
Mirror Reflection



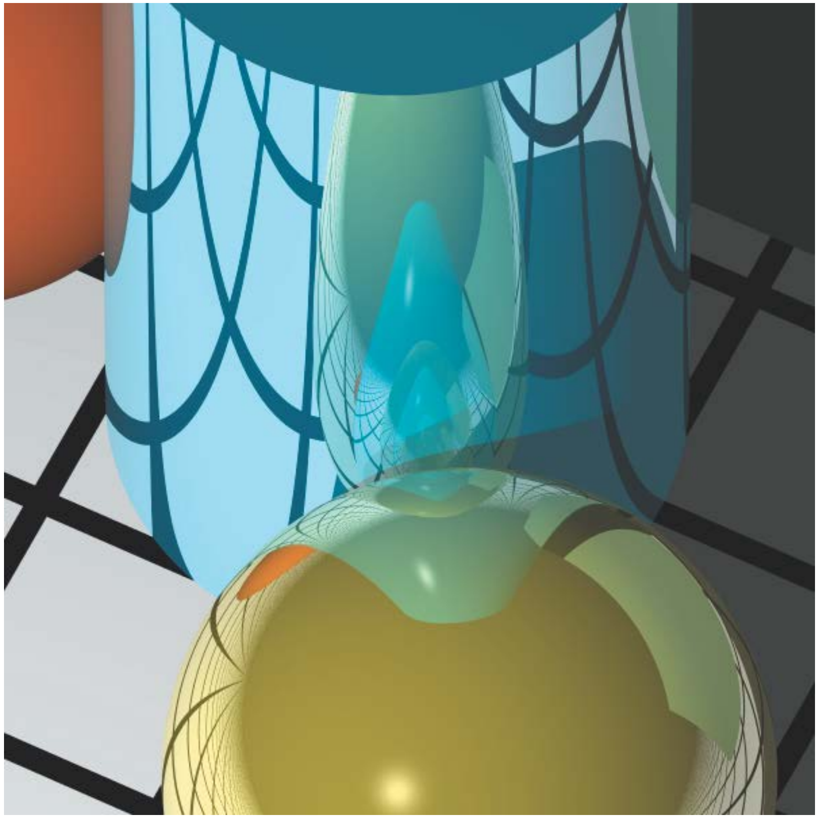
Max depth = 1

Recursive Ray Tracing
Mirror Reflection

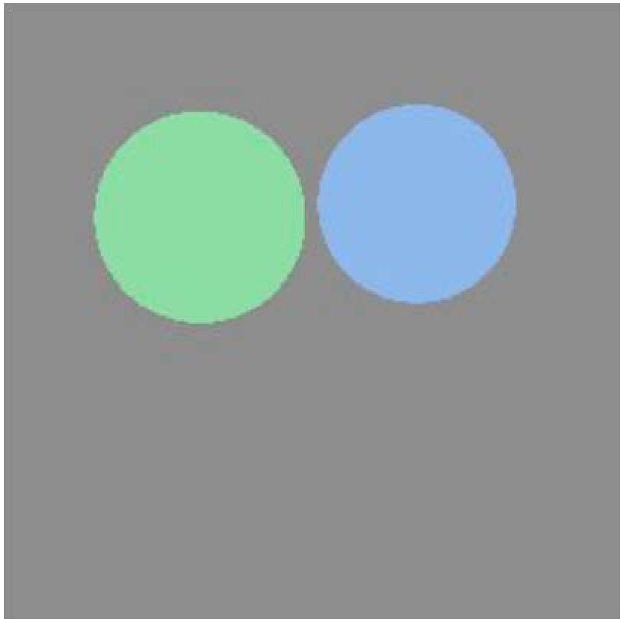
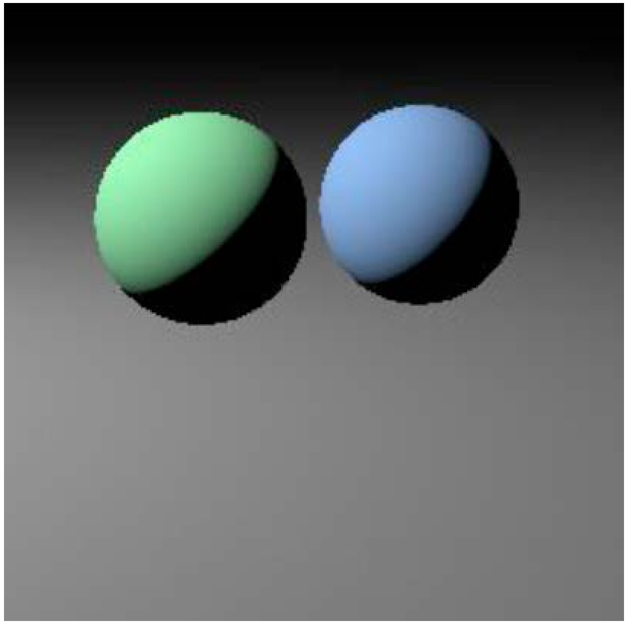
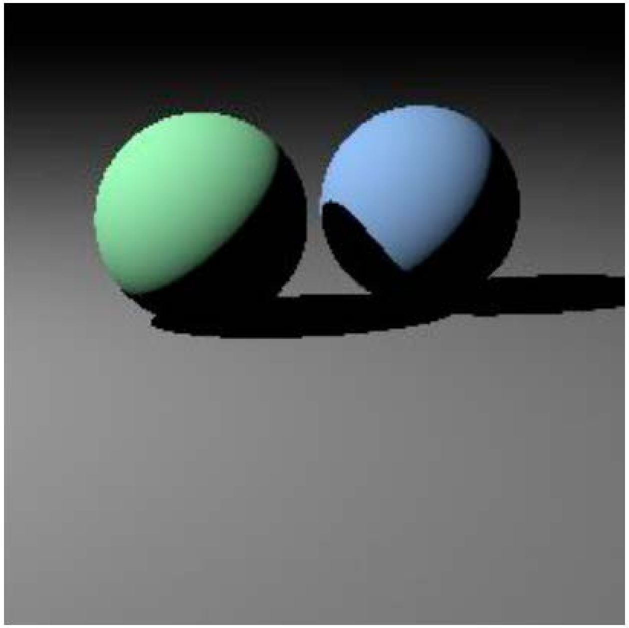
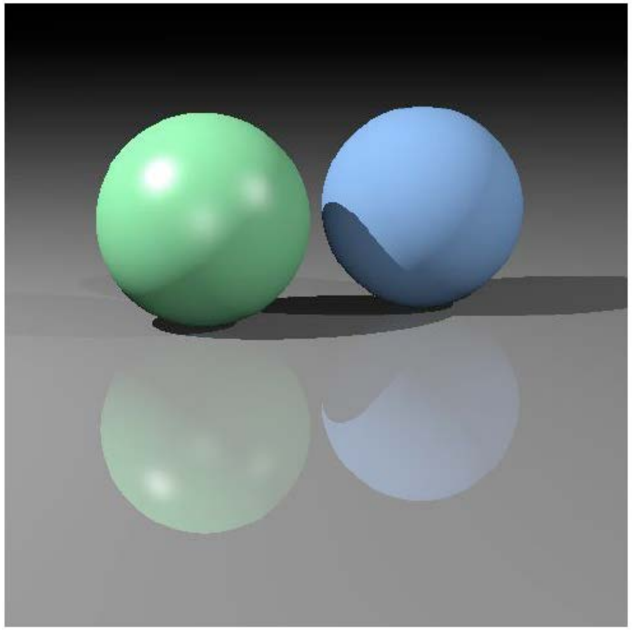


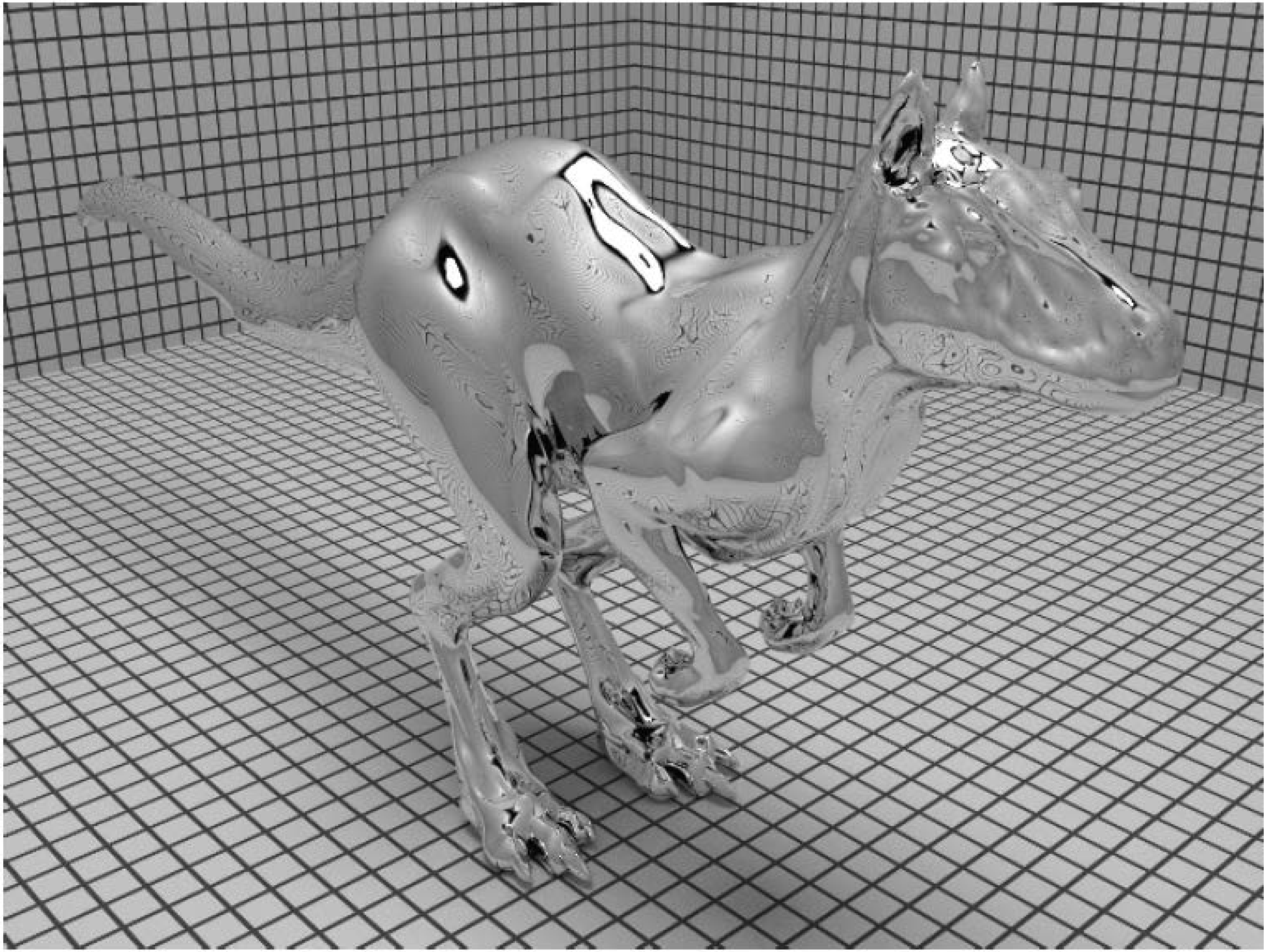
Max depth = 10

Recursive Ray Tracing
Mirror Reflection



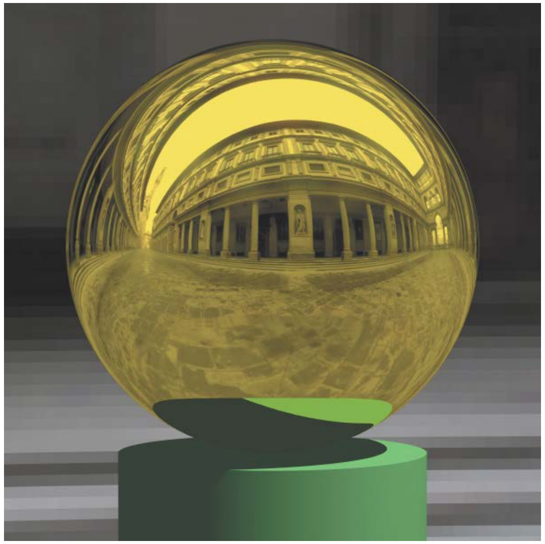
Max depth = 10

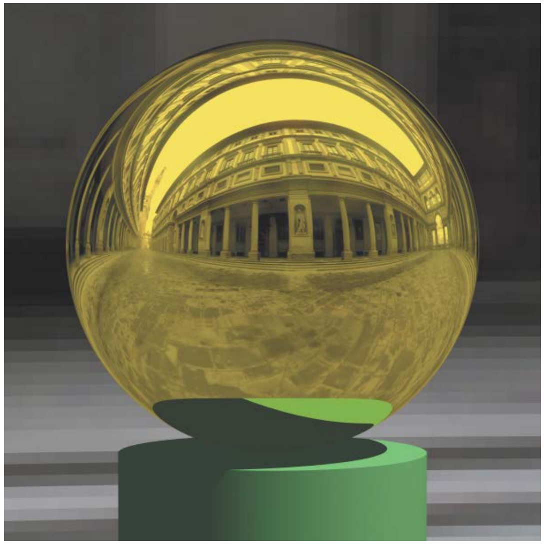


Tesselated models

Tesselated sphere; shading normal = geometry normal



Tesselated sphere; interpolated shading normals (cfr. Phong Interpolation)



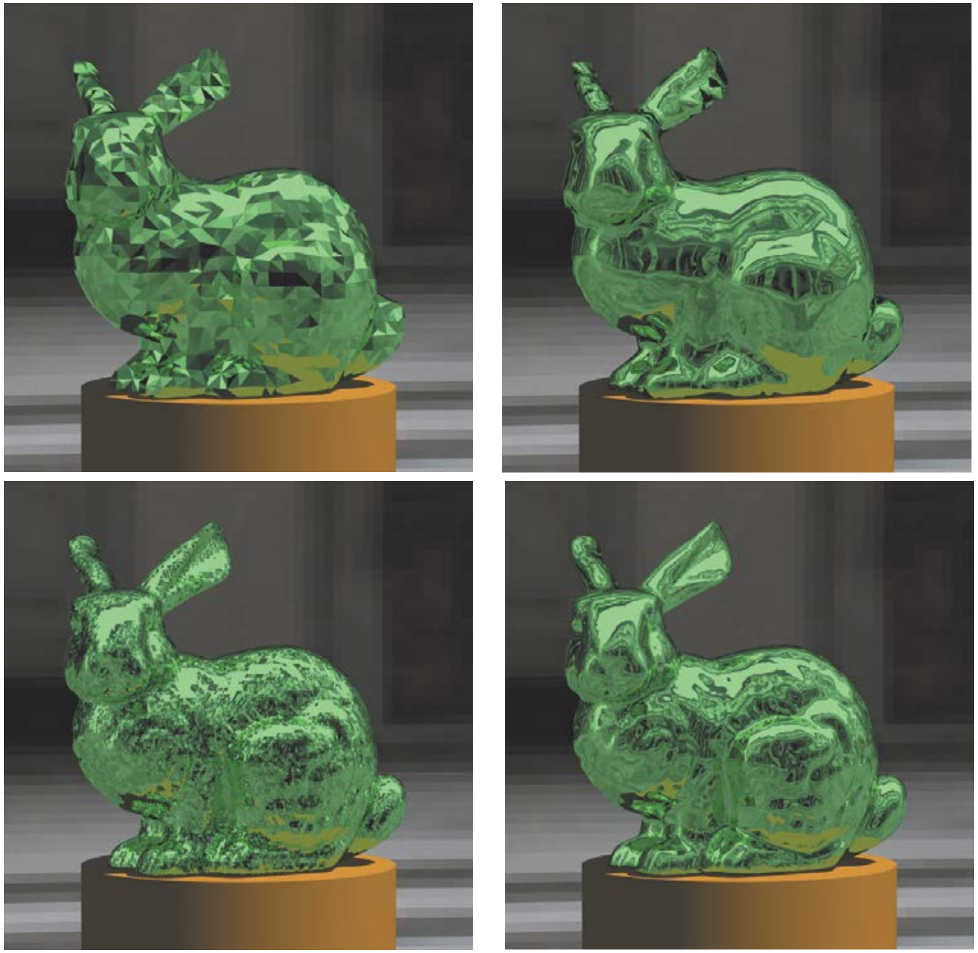
Perfect sphere

(shading normal is used to compute perfect reflected direction)

Tesselated models

3K triangles

3K triangles
interpolated normals



69K triangles

69K triangles
interpolated normals

Glossy Reflection

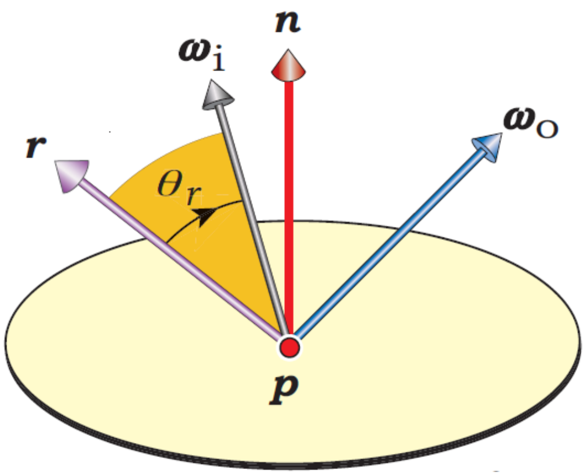
$$L_o(p,\omega_o) = \int_{hemisphere} f_r(p,\omega_i,\omega_o) L_i(p,\omega_i) \cos\theta_i \, d\omega_i$$

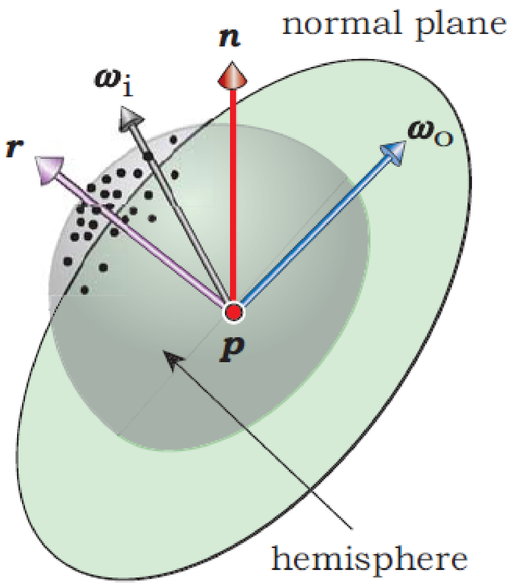
$$L_o(p,\omega_o) = \int_{hemisphere} C.\cos(\theta_r)^e L_i(p,\omega_i) \cos\theta_i \, d\omega_i$$

Monte Carlo integration

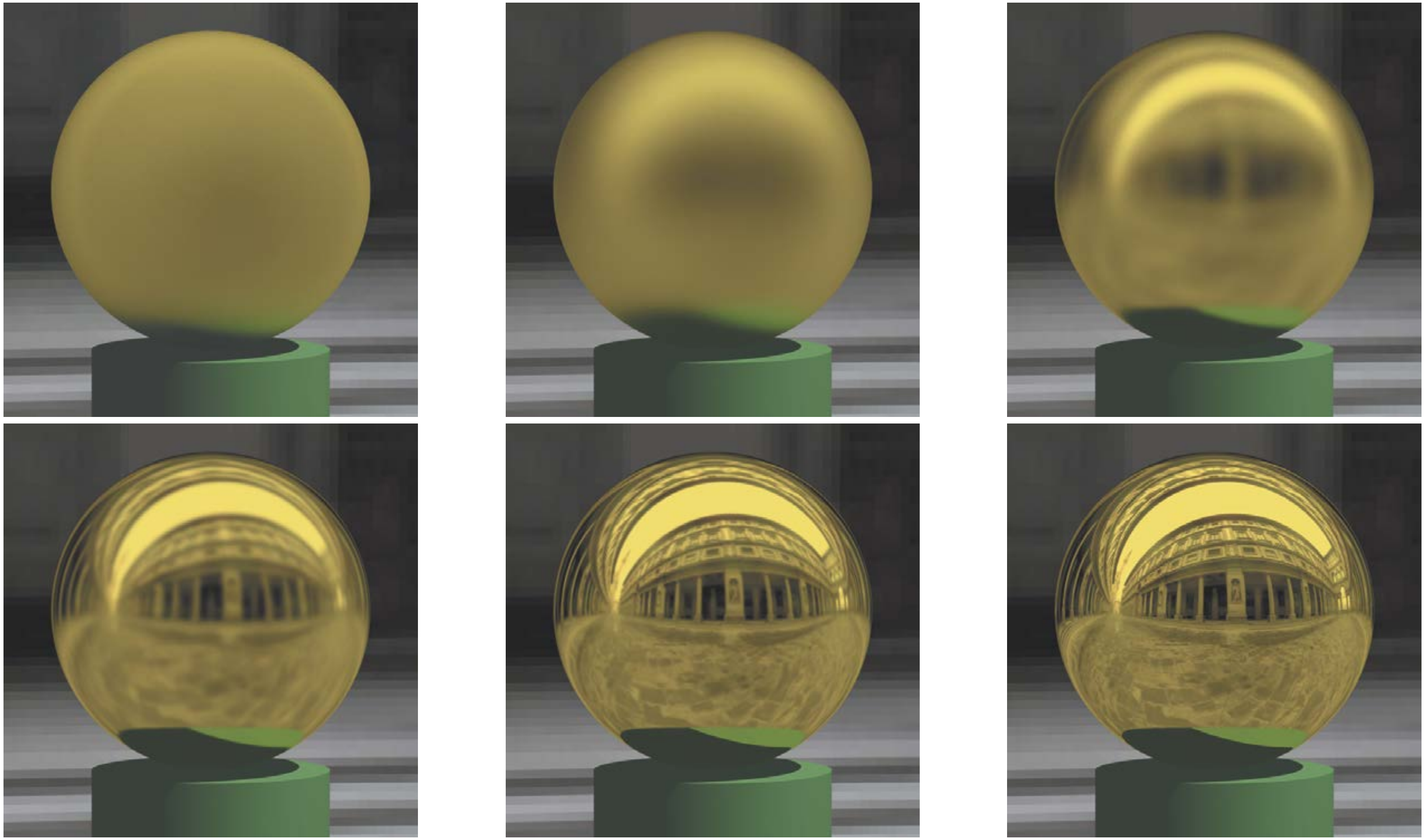
$$L_o(p,\omega_o) \approx \frac{1}{N}\sum_{j=1}^{N} \frac{C.\cos(\theta_{r,j})^e L_i(p,\omega_{i,j}) \cos\theta_{i,j} \sin\theta_{i,j}}{pdf(\omega_j)}$$

Can be made proportional to $\cos(\theta_r)^e$

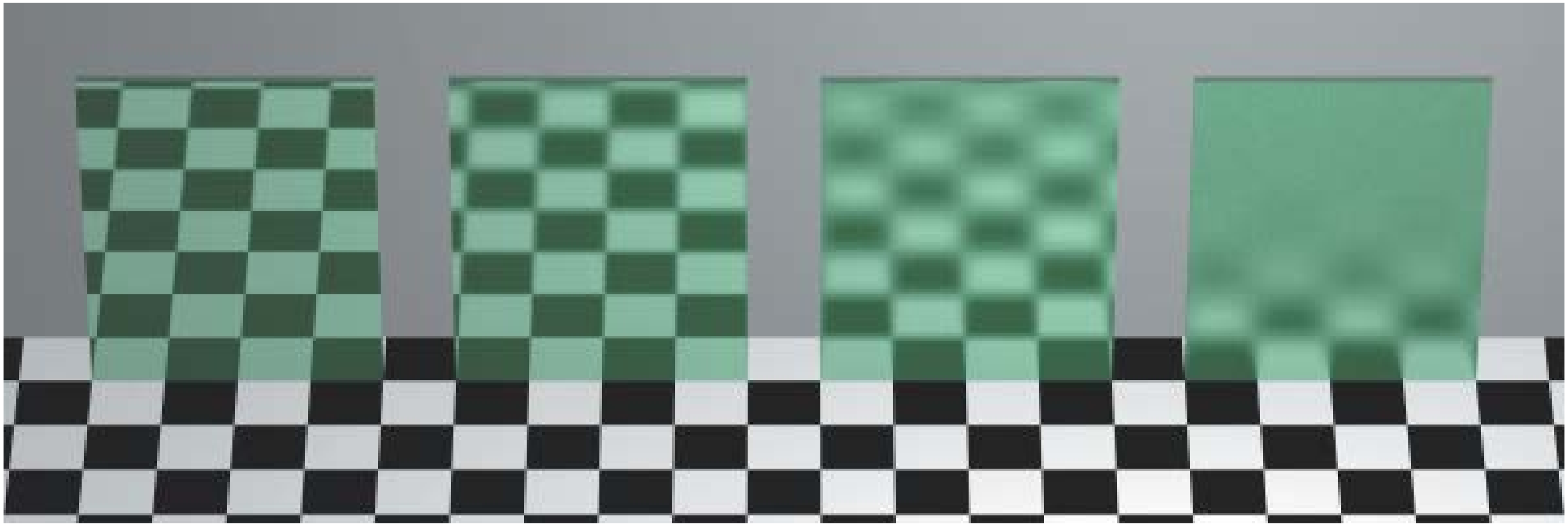




Glossy Reflection



Glossy Reflection



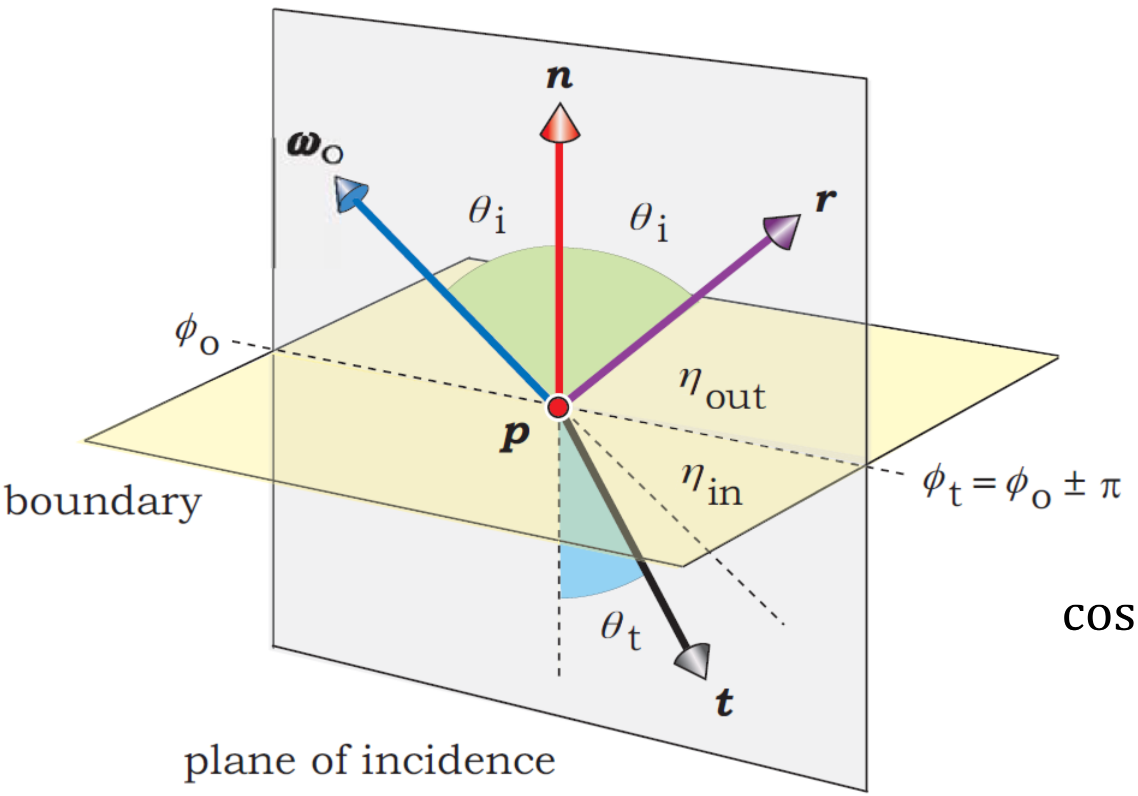
Recursive ray tracing
Perfect refraction

Transparant materials: index of refraction

- Vacuum … 1.0
- Water … 1.33
- Diamond … 2.42

(wikipedia.org)

Recursive ray tracing
Perfect refraction

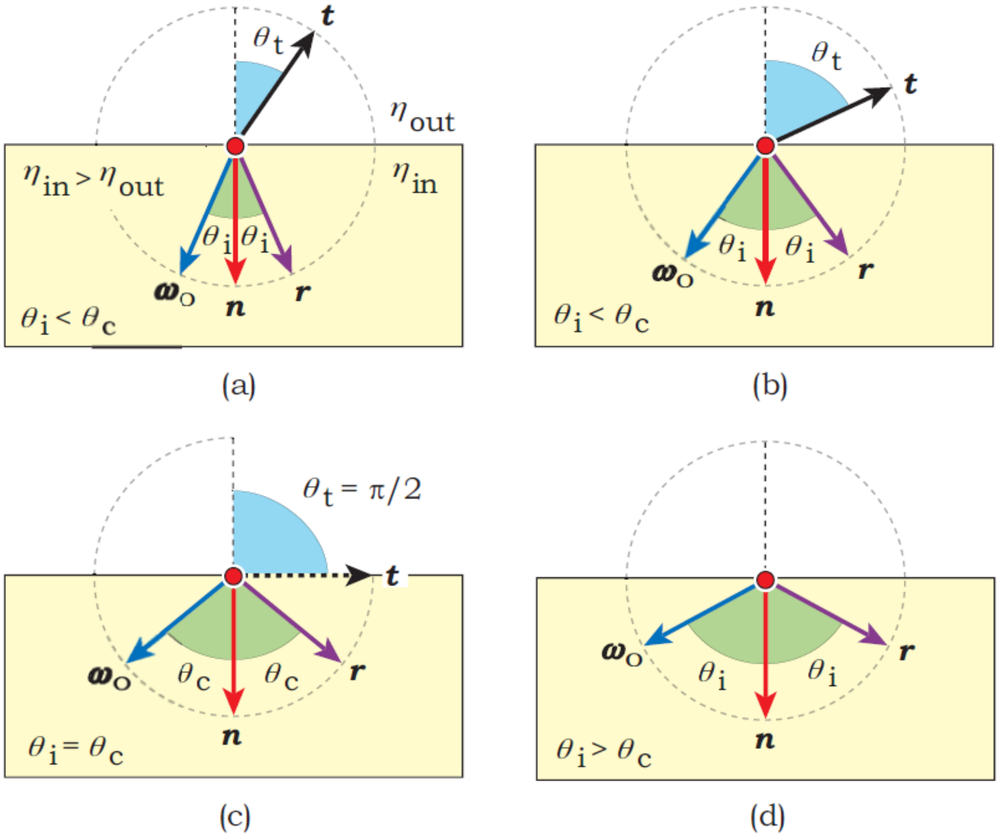


$$\frac{\sin(\theta_i)}{\sin(\theta_t)} = \frac{\eta_{in}}{\eta_{out}} = \eta$$

$$\cos(\theta_t) = \sqrt{(1 - \frac{1}{\eta^2}(1 - \cos^2(\theta_i))}$$

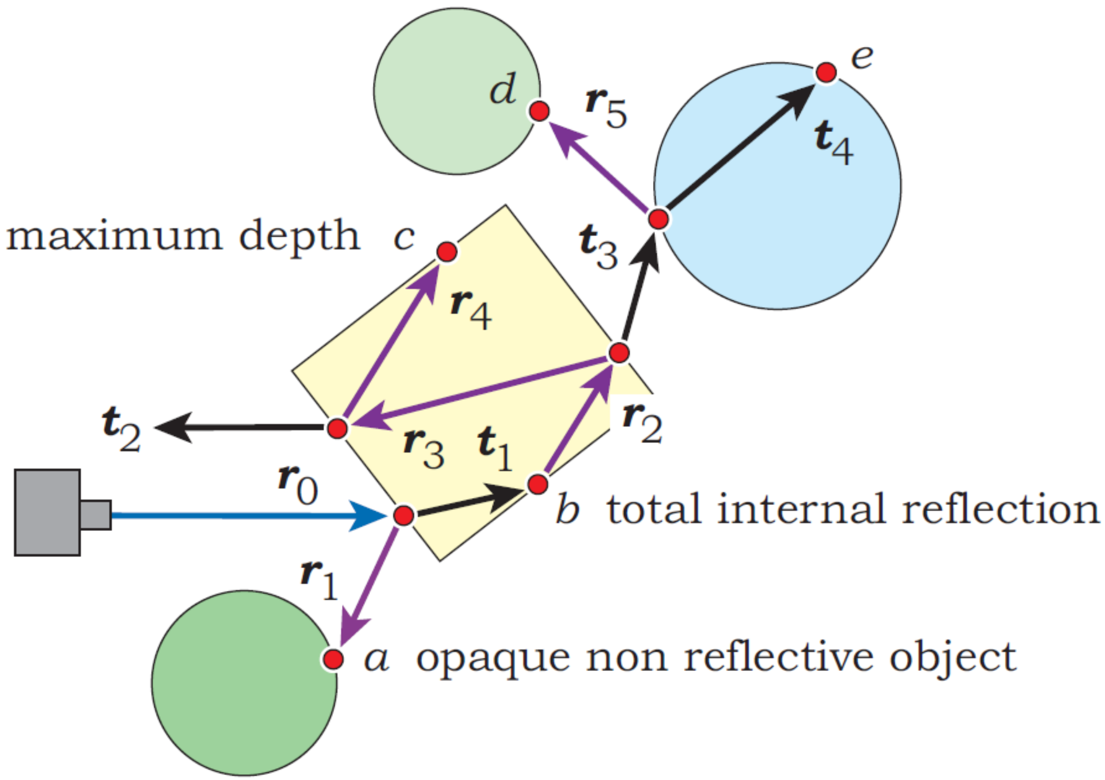
Recursive ray tracing
Perfect refraction

Total internal refraction

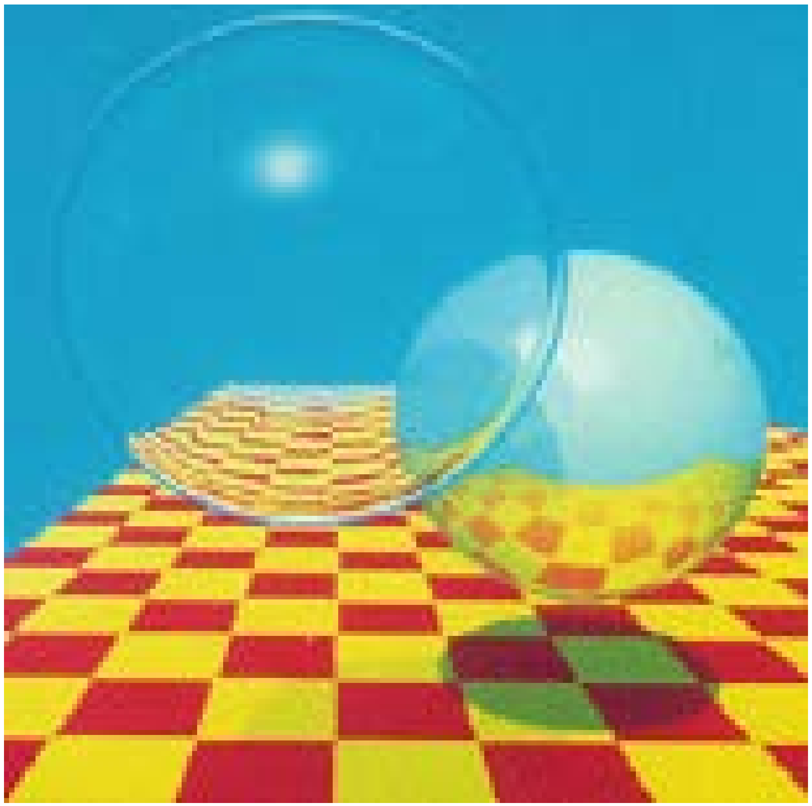


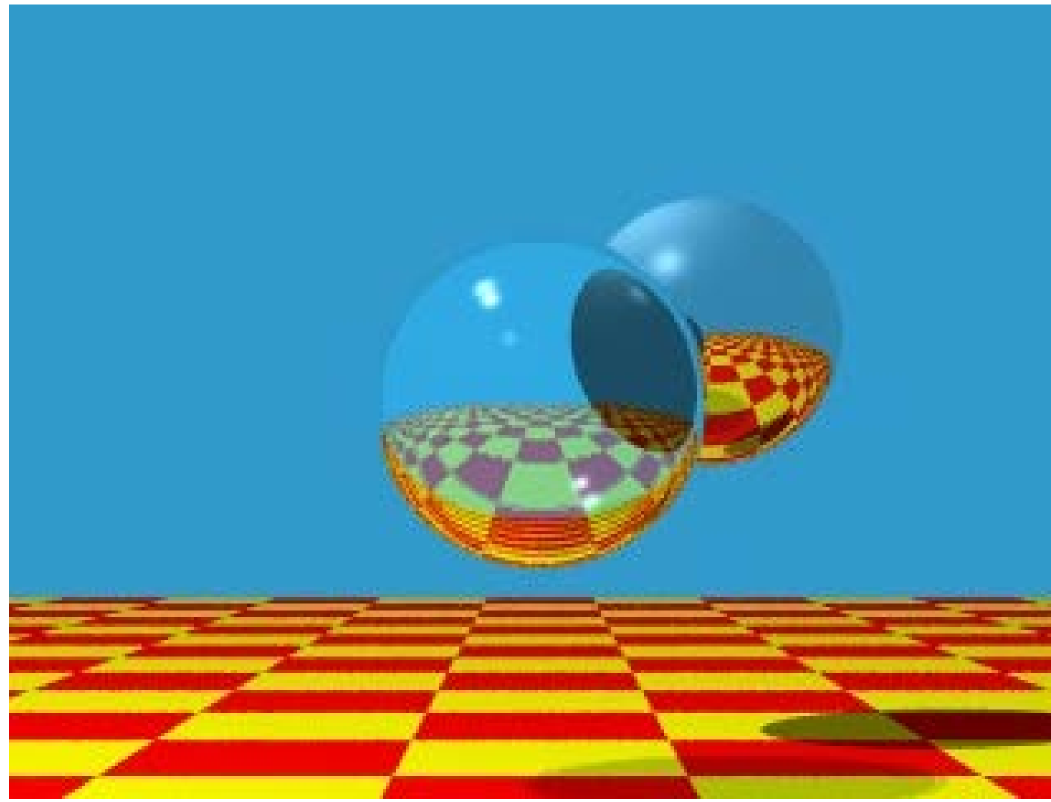
Recursive ray tracing
Perfect refraction

Trace transparant ray recursively (cfr. reflective ray)



Very first raytraced pictures

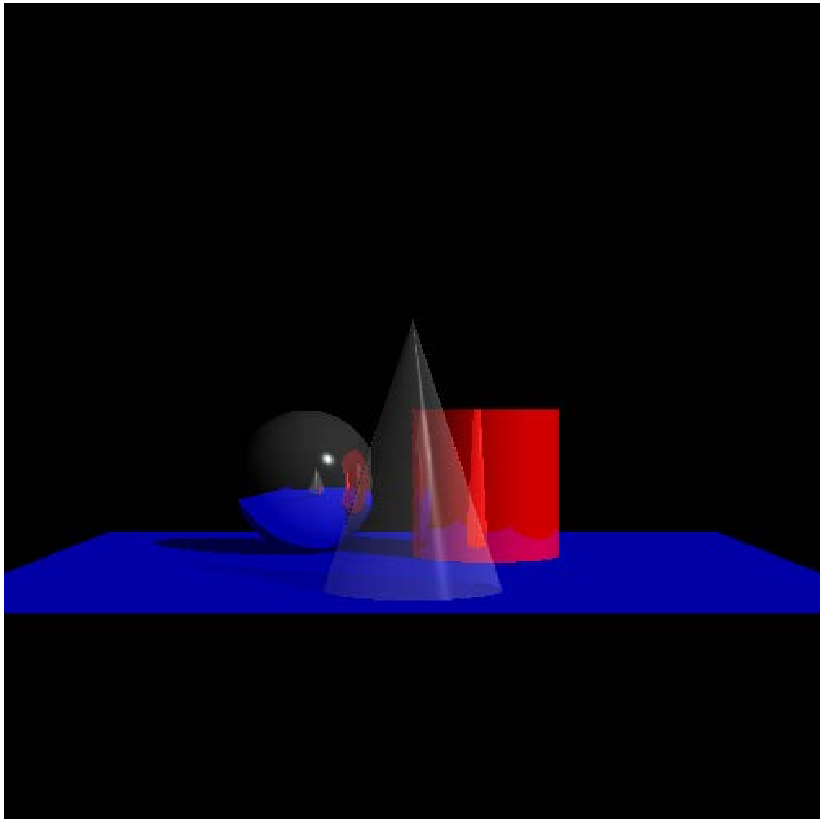




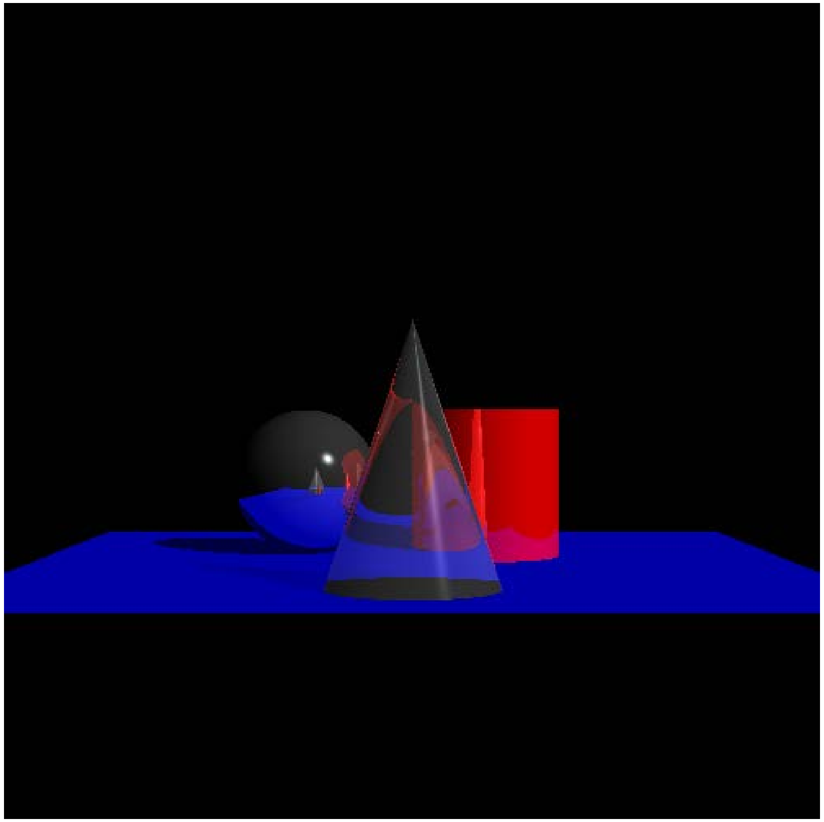
(Whitted 1980)

https://youtu.be/WV4qXzM641o

Recursive ray tracing
Perfect refraction

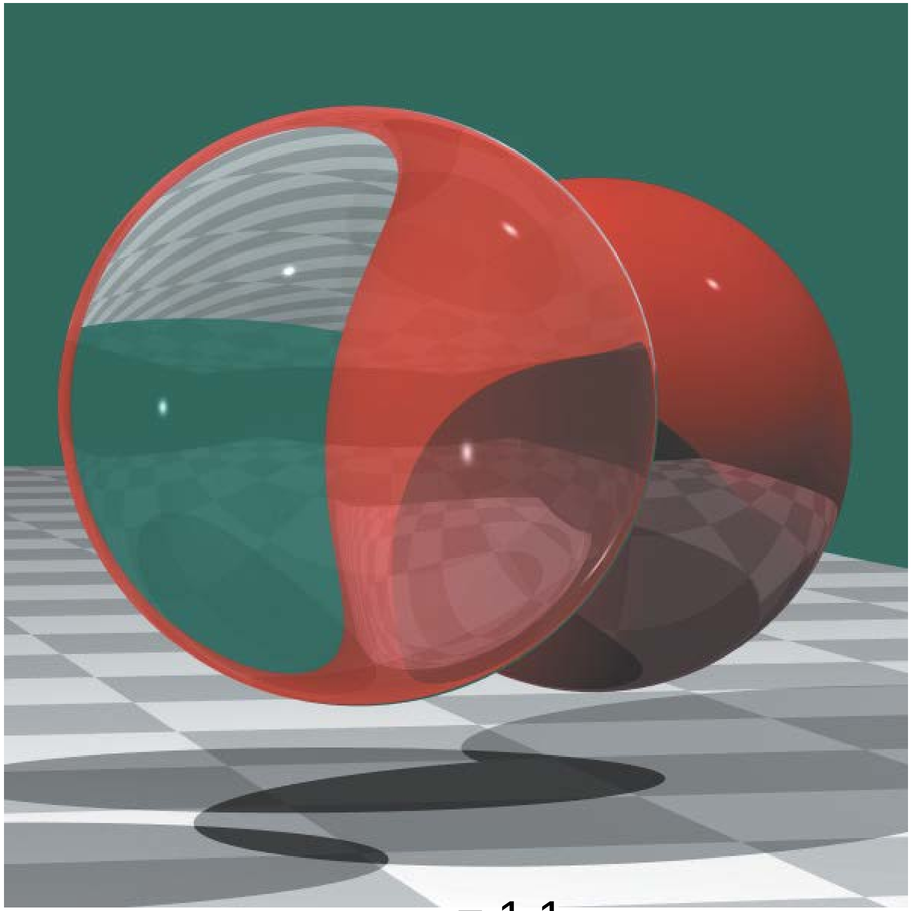


No refraction

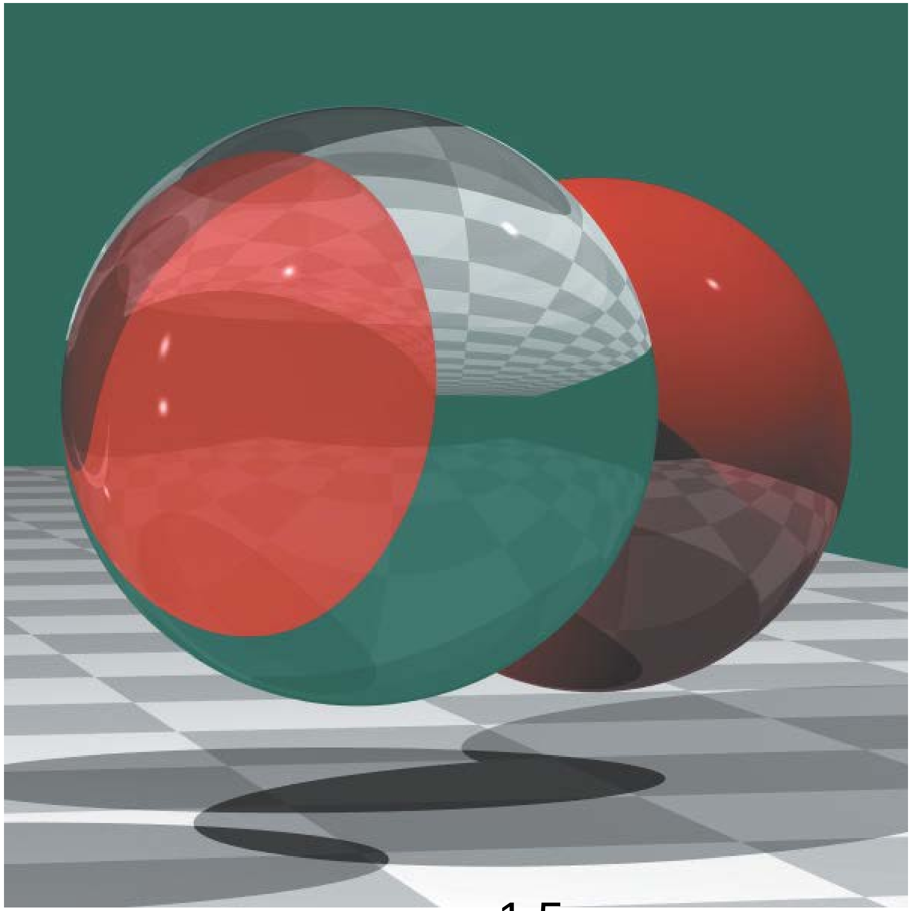


Refraction

Recursive ray tracing
Perfect refraction

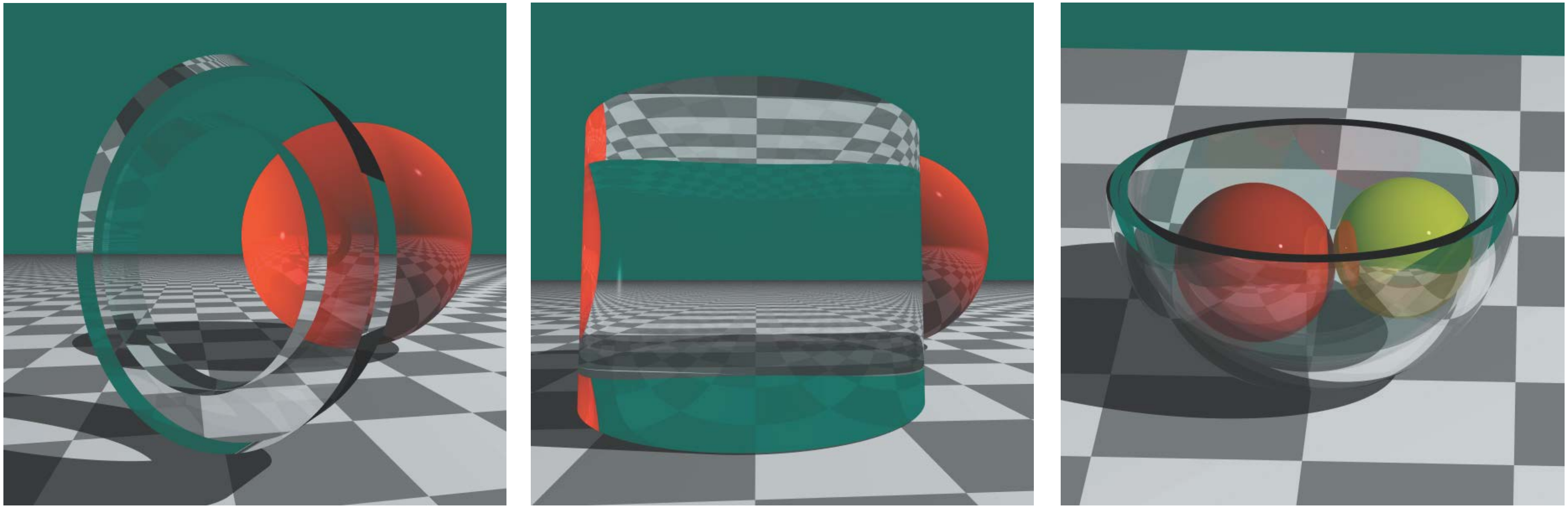


$\eta = 1.1$



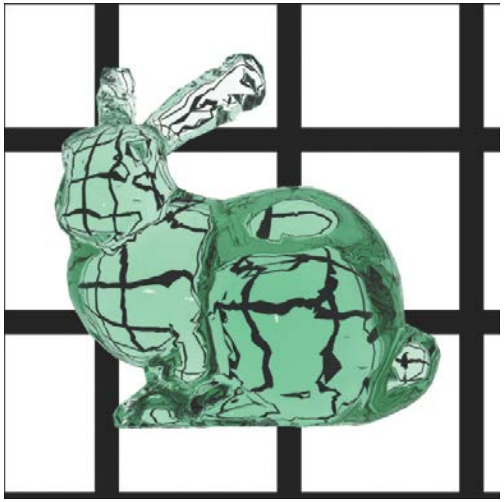
$\eta = 1.5$

Recursive ray tracing
Perfect refraction

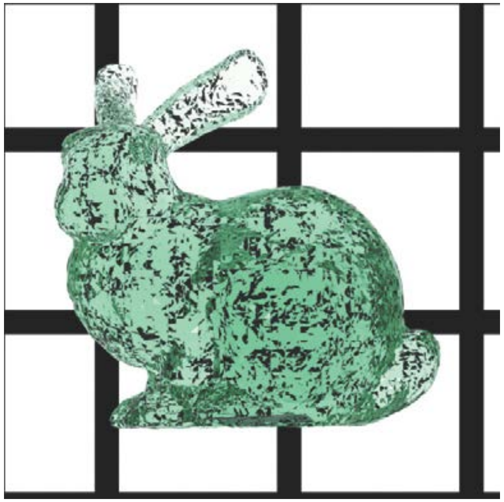


Tesselated models

Cfr. tesselated models for mirror reflection



3K / interp. normals

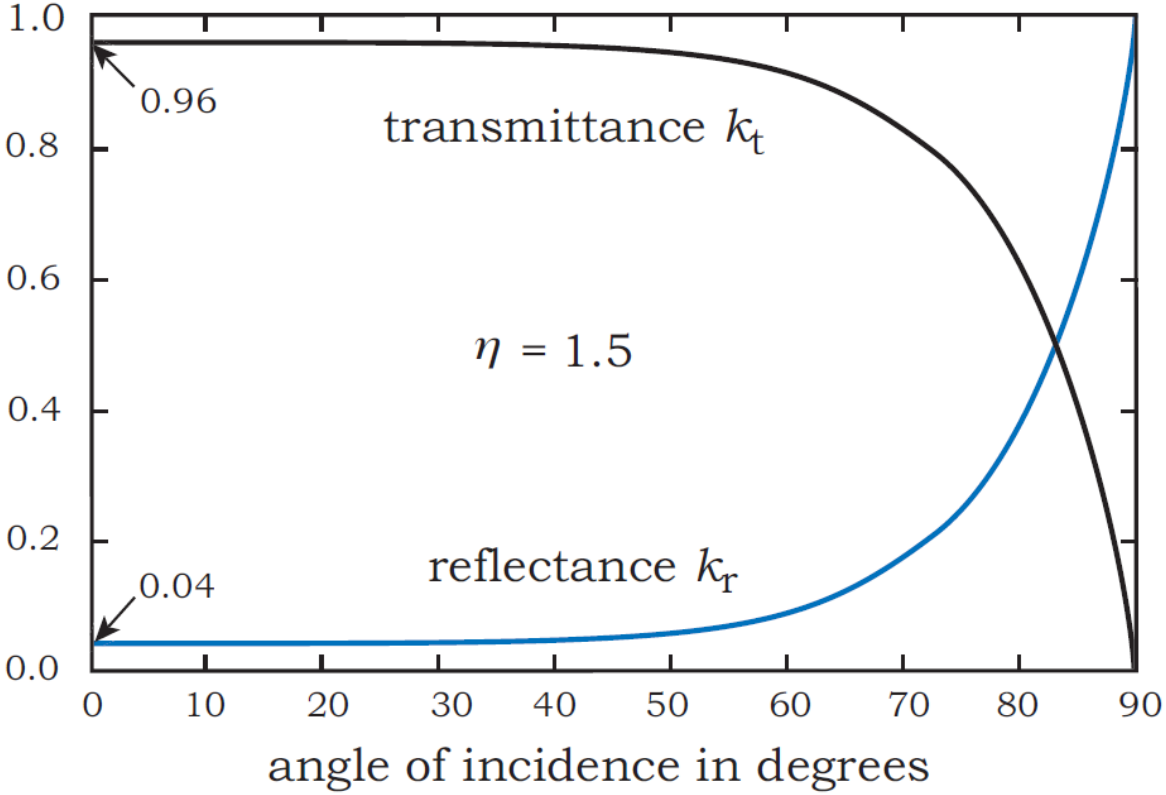


16K, no interpolation

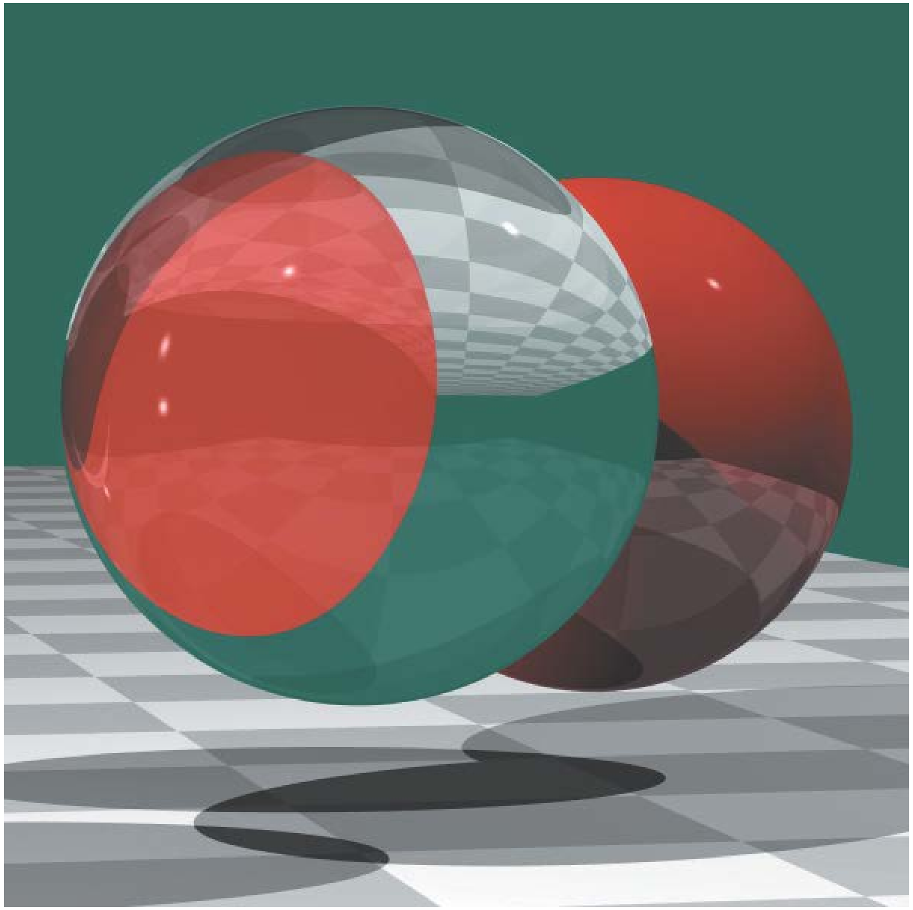
69K / interp. normals

Realistic Transparancy

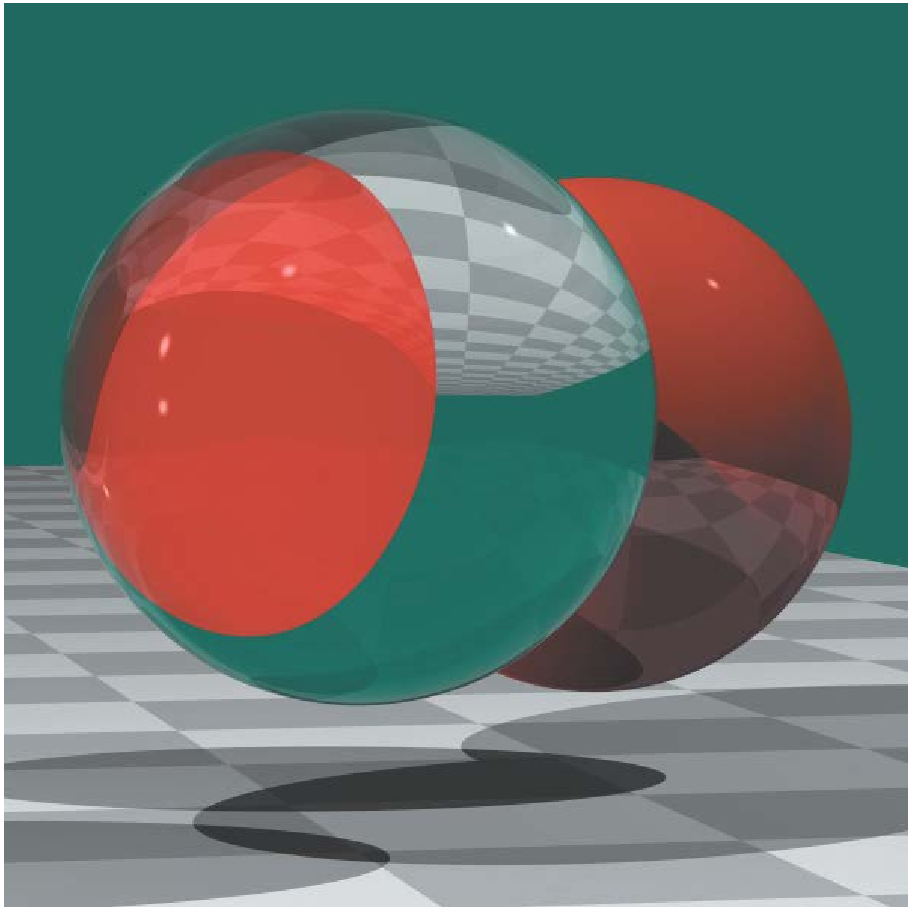
Fresnel reflectance and transmittance



Realistic Transparancy



No Fresnel



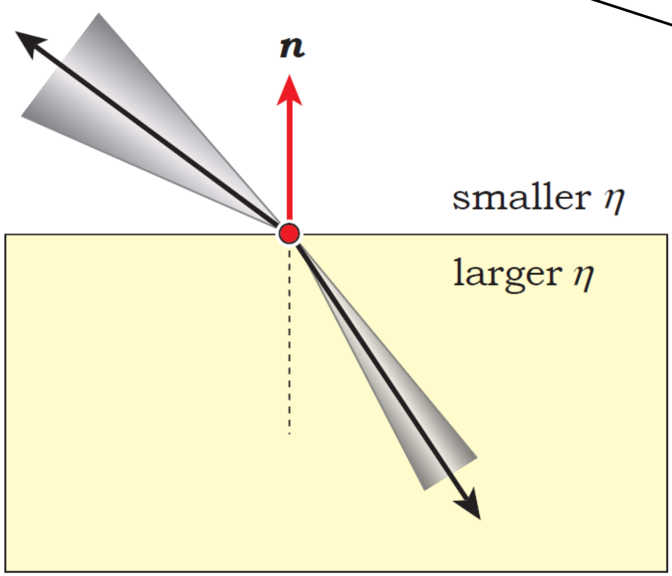
Fresnel

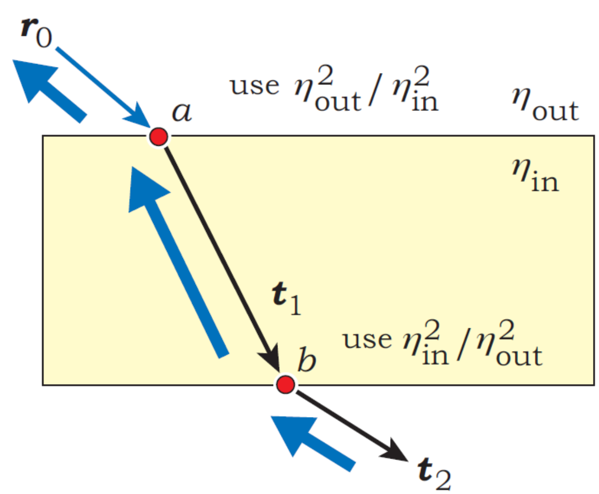
Refraction and rendering equation

$$L_{indirect}(p,\omega_o) = \int_{hemisphere} f_r(p,\omega_i,\omega_o) L_i(p,\omega_i)\cos\theta_i \, d\omega_i$$

$$= \frac{\eta_t^2}{\eta_i^2} . \tau . L_i(p,\omega_i)$$

τ = 1 - Fresnel reflectance





Transparant Attenuation

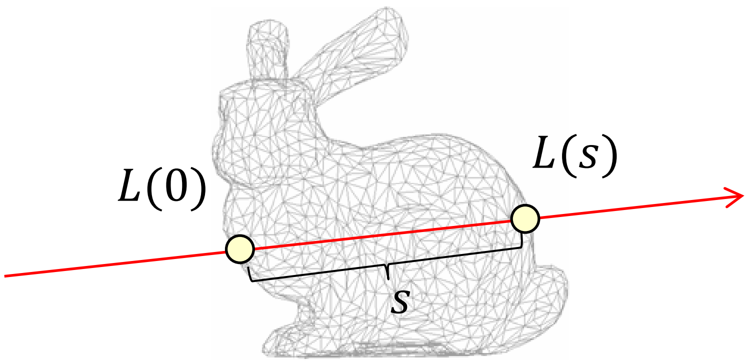
Beer's law:

$$L(s) = L(0)e^{s.\ln(a)}$$

s = distance in material

a = intensity change after one unit of distance

$$L(1) = L(0)e^{\ln(a)} = aL(0)$$



(store distance traveled for each ray since last intersection point)

Putting everything together

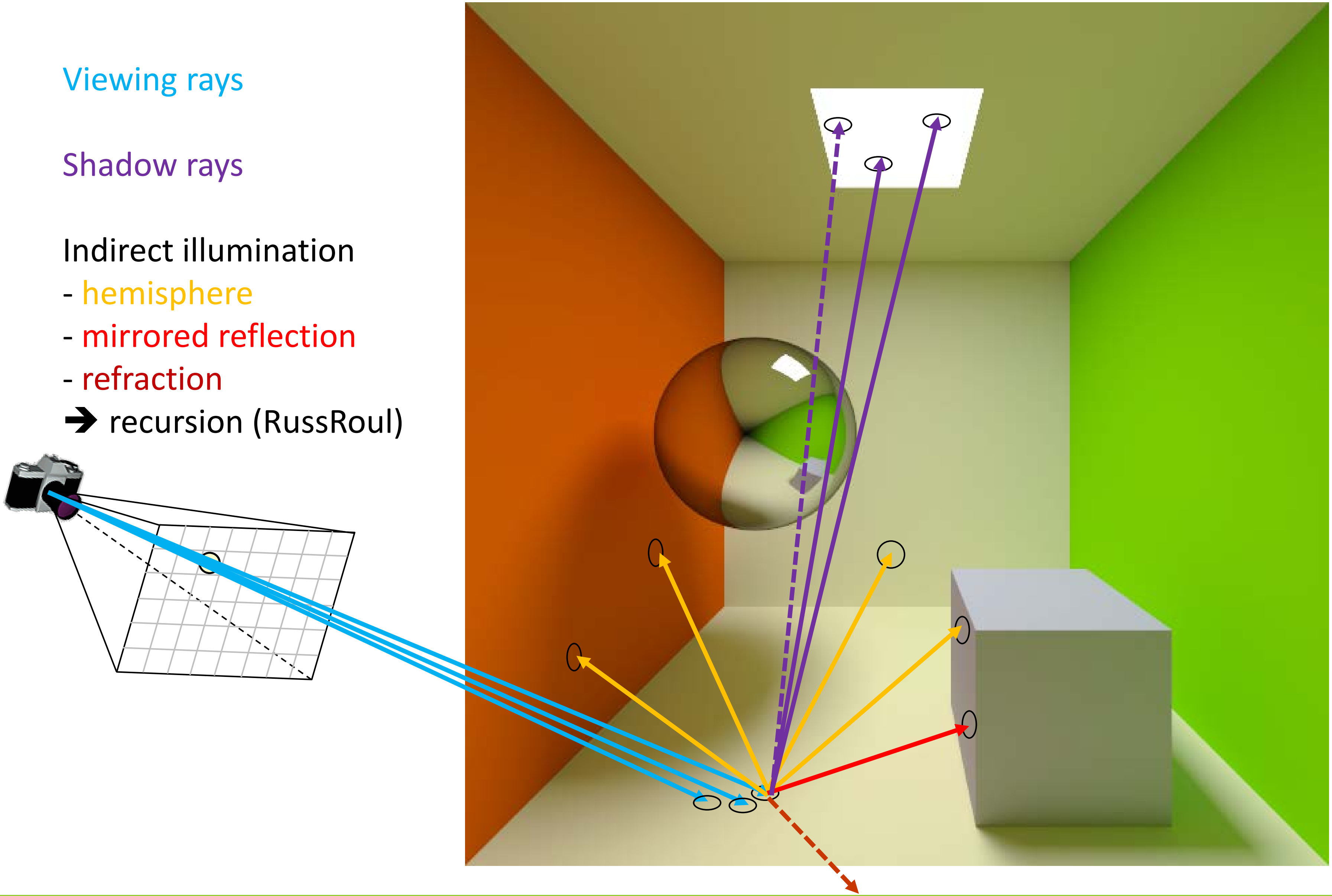
$$
\begin{aligned}
L_o &= L_e + L_{reflected} \\
&= L_e + L_{direct} + L_{indirect} \\
&= L_e + \int_{A_L} f_r L_e V G dA_{p'} + \int_{\Omega} f_r \, L_{reflected} \cos\omega_i \, d\omega_i \\
&= L_e + \int_{A_L} f_r L_e V G dA_{p'} + \int_{\Omega} (f_{diff} + f_{gloss} + f_{mirror} + f_{refr}) \, L_{indirect} \cos\omega_i \, d\omega_i \\
&= L_e + \int_{A_L} f_r L_e V G dA_{p'} + \int_{\Omega} f_{diff} \, L_{indirect} \cos\omega_i \, d\omega_i + \int_{\Omega} f_{gloss} \, L_{indirect} \cos\omega_i \, d\omega_i + \ldots
\end{aligned}
$$

Viewing rays

Shadow rays

Indirect illumination
- hemisphere
- mirrored reflection
- refraction

➔ recursion (RussRoul)

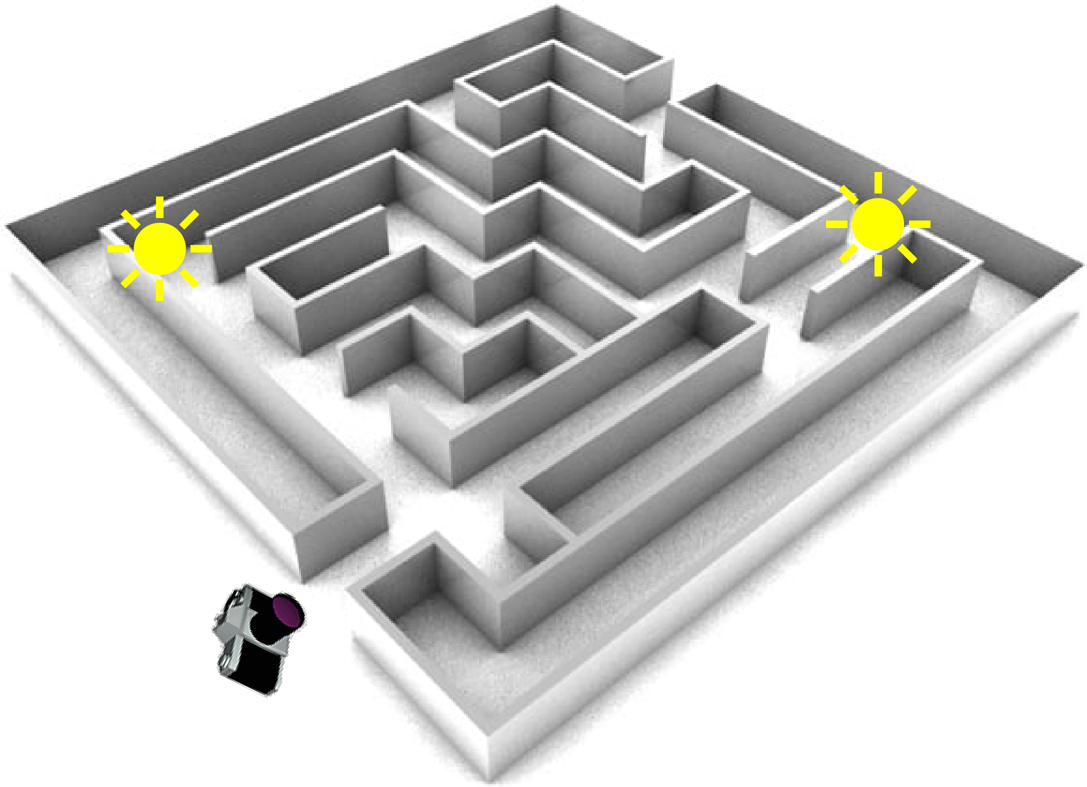


Putting everything together

Can we now render all possible light effects?

Putting everything together

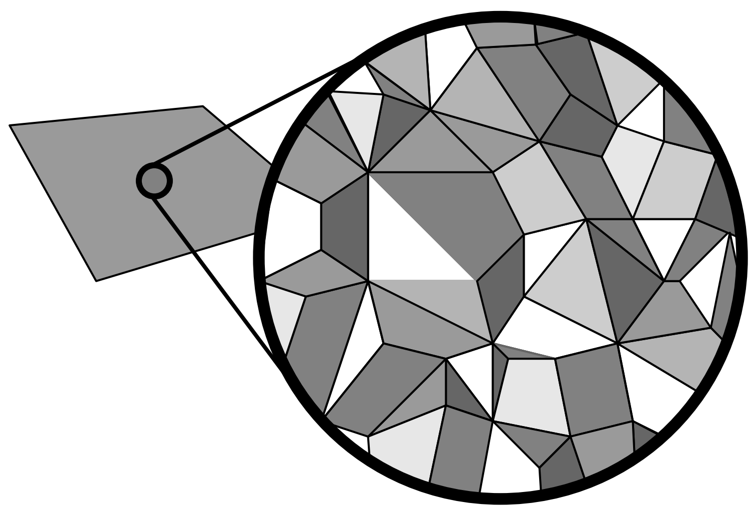
Can we now render all possible light effects?



Microfacet Models

Surface = modelled as a collection of small microfacets

- Distribution of facets
- One "lightbeam" covers a large amount of microfacets



Microfacet Models

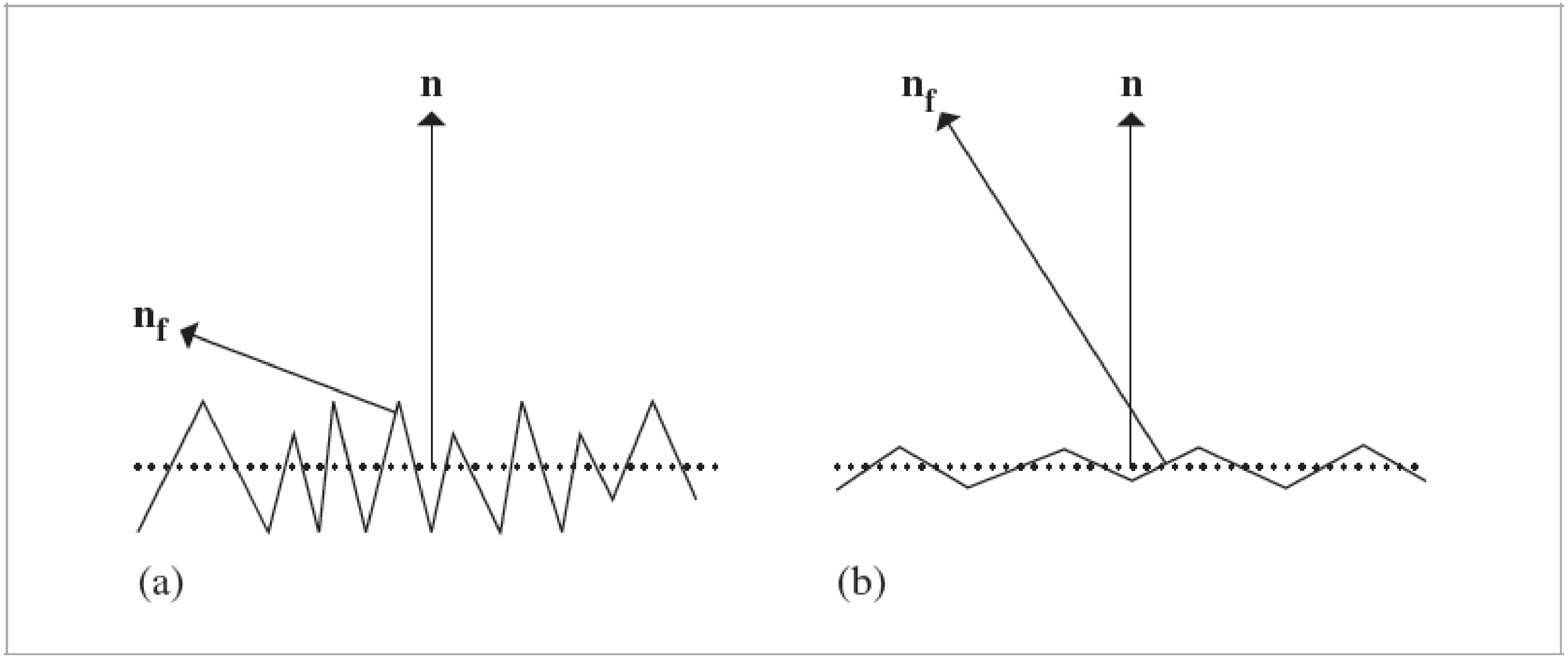
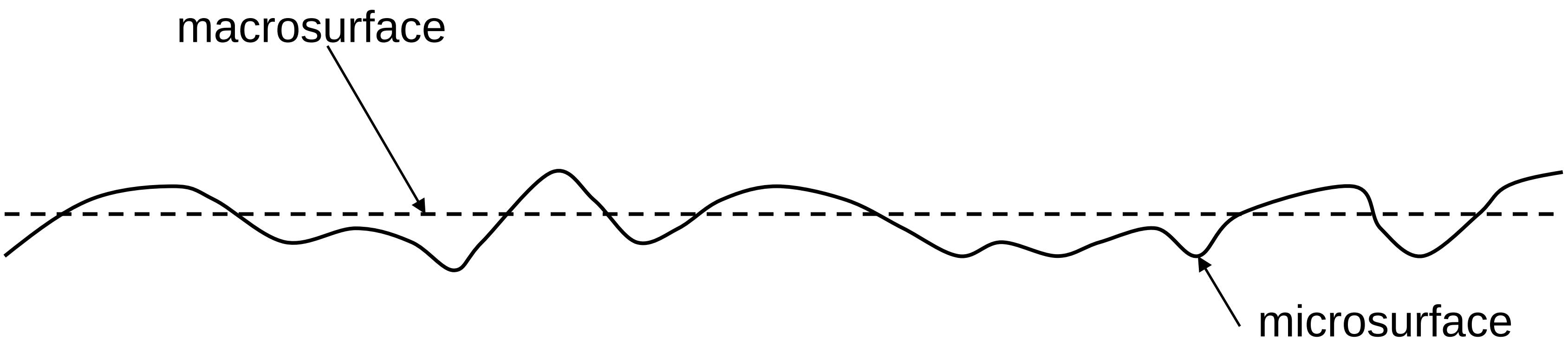


Figure 8.12: Microfacet surface models are often described by a function that gives the distribution of microfacet normals $\mathbf{n_f}$ with respect to the surface normal $\mathbf{n}$. (a) The greater the variation of microfacet normals, the rougher the surface is. (b) Smooth surfaces have relatively little variation of microfacet normals.

Microfacet Models

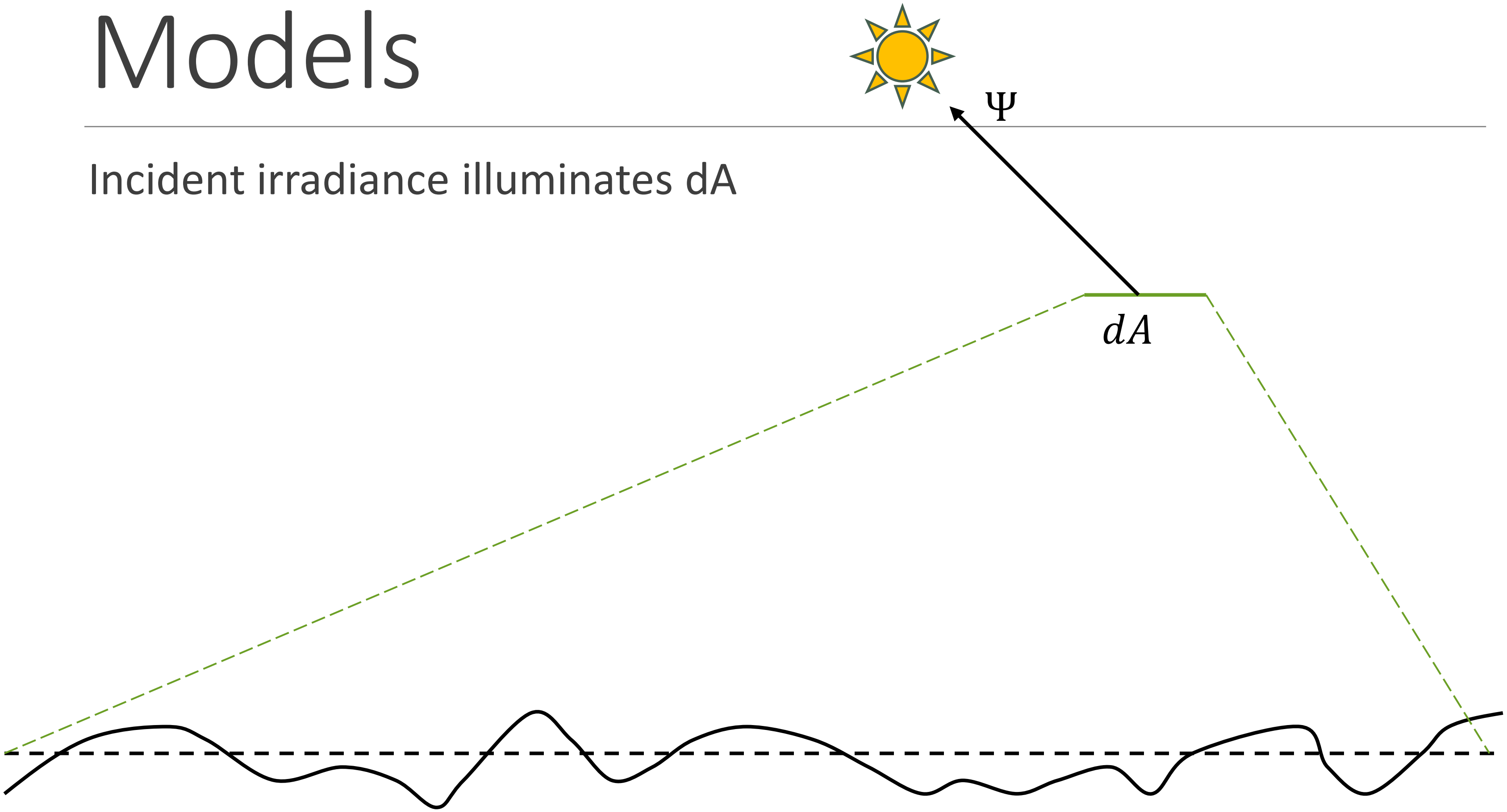
Surface

- rough at microscale
- flat at macroscale



Microfacet Models

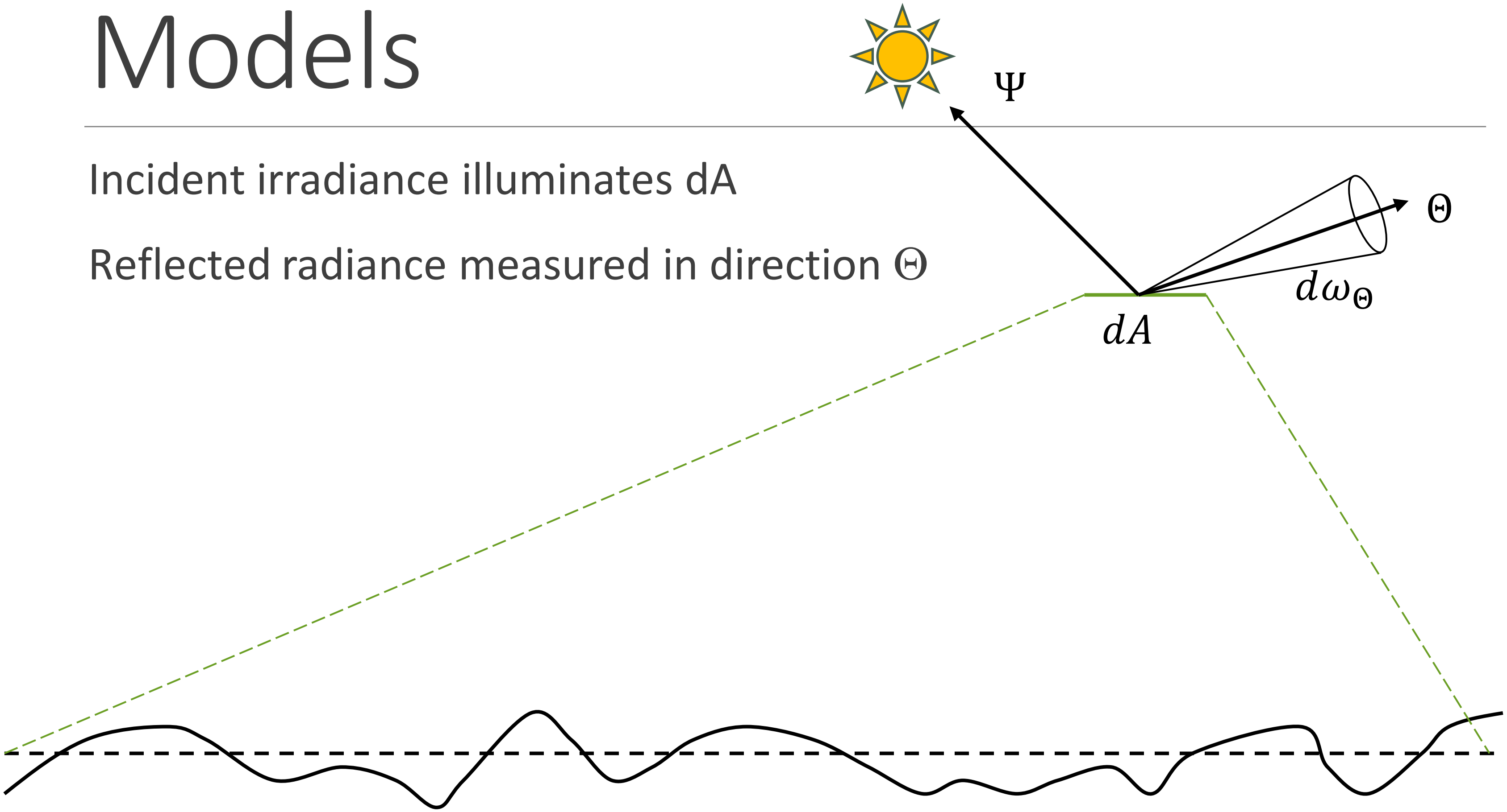
Incident irradiance illuminates dA



Microfacet Models

Incident irradiance illuminates dA

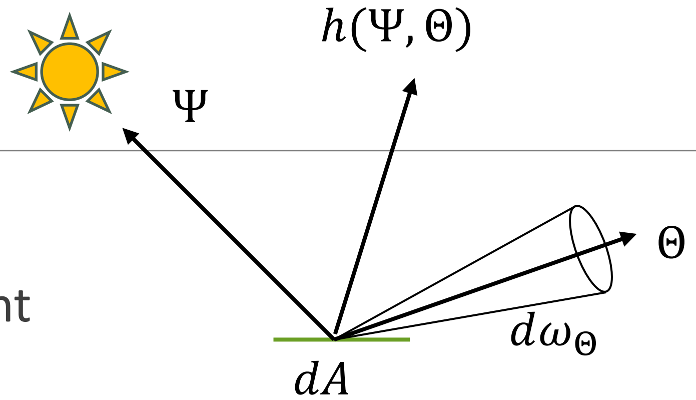
Reflected radiance measured in direction Θ

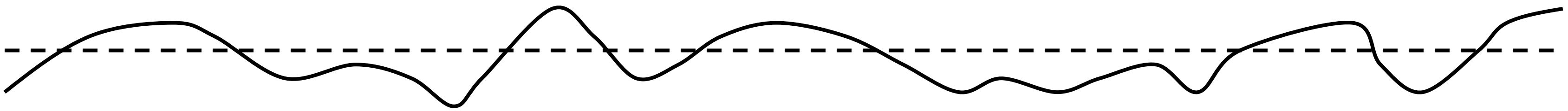


Microfacet Models

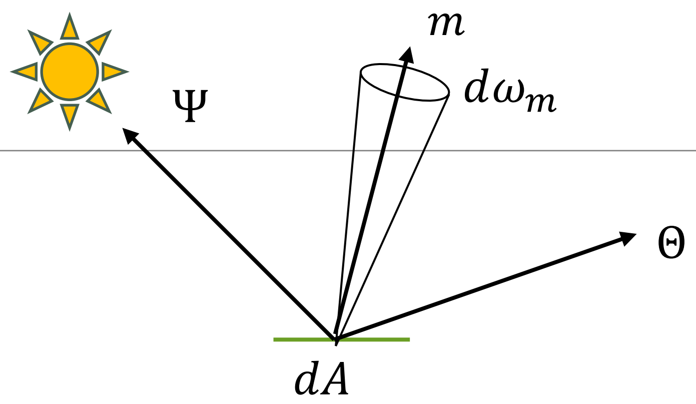
Halfvector h(Ψ, Θ):

microfacet orientation that scatters light from Ψ to Θ



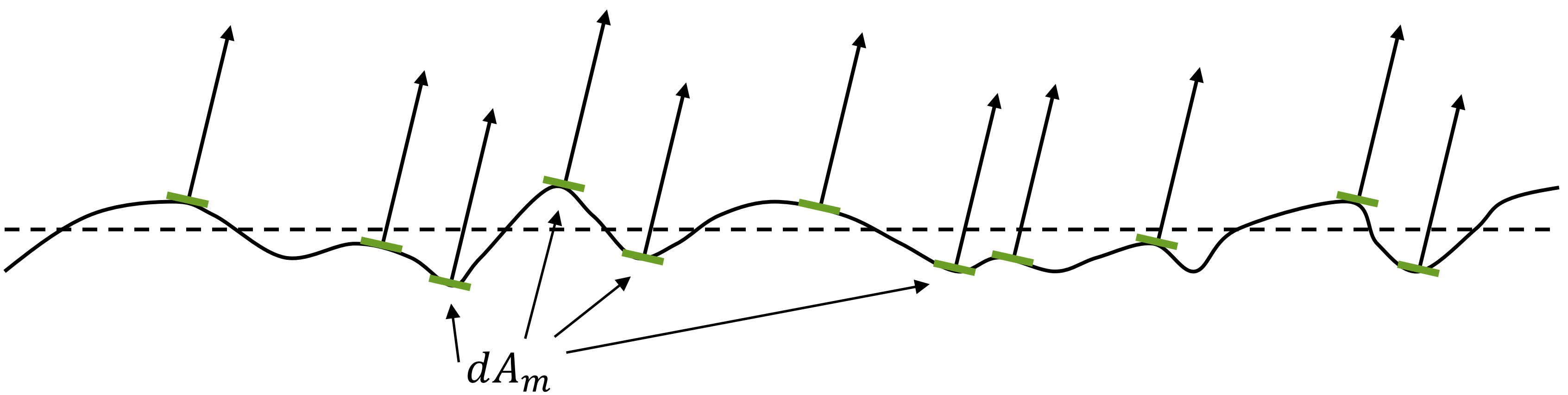


Microfacet Models



Normal distribution $D(m)$

Measures density of microsurface area w.r.t. normal m



Masking effects

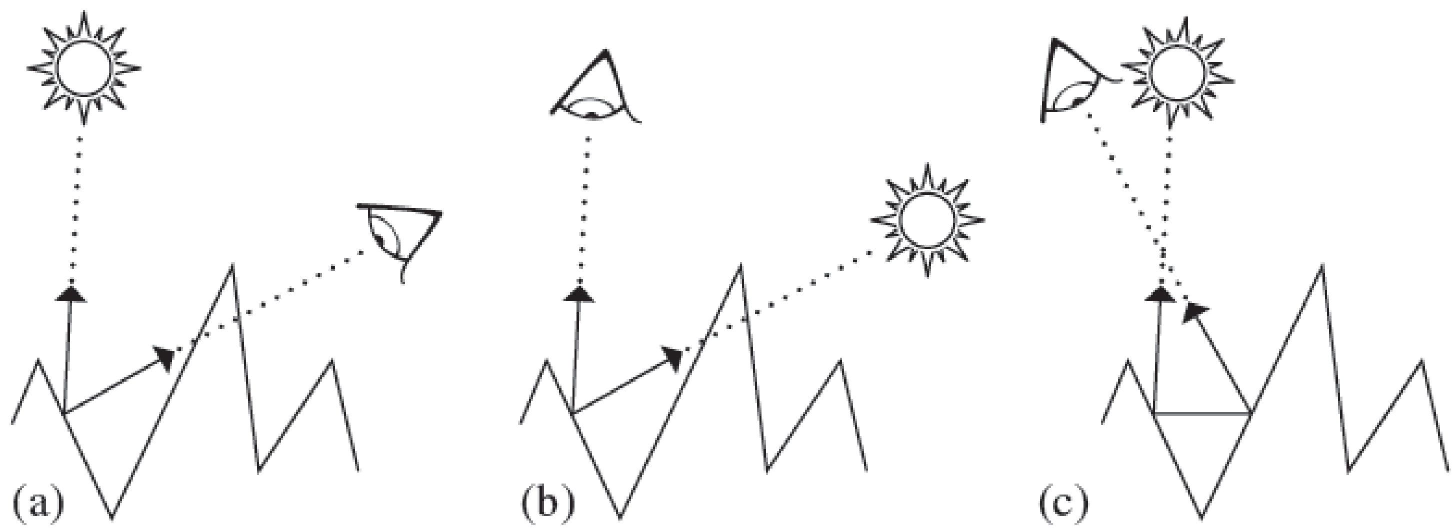
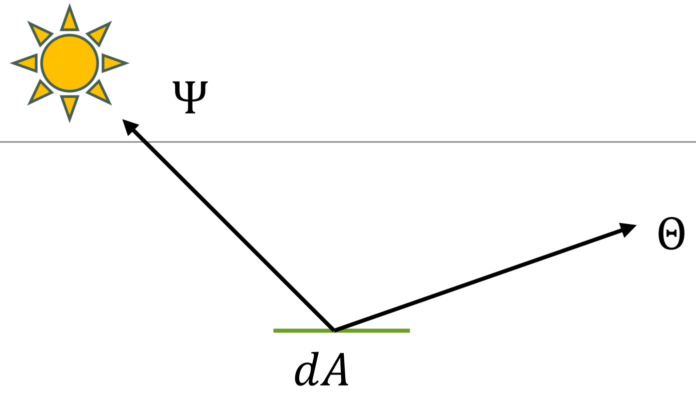
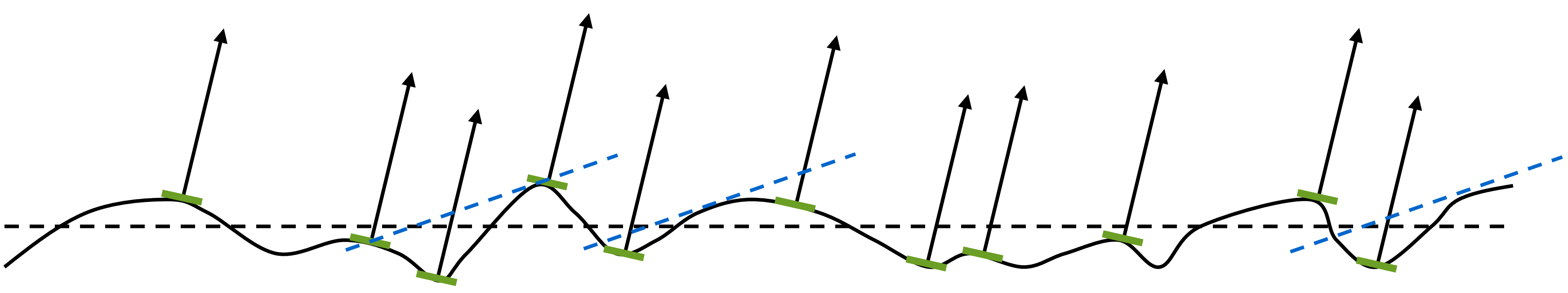


Figure 8.13: Three Important Geometric Effects to Consider with Microfacet Reflection Models. (a) *Masking*: the microfacet of interest isn't visible to the viewer due to occlusion by another microfacet. (b) *Shadowing*: analogously, light doesn't reach the microfacet. (c) *Interreflection*: light bounces among the microfacets before reaching the viewer.

Masking effects

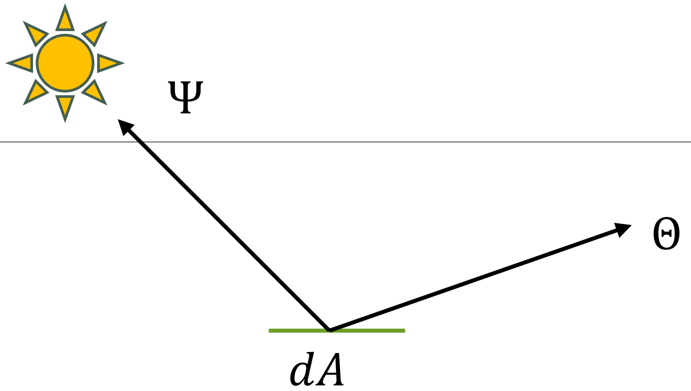
Shadow masking $G(\Psi, \Theta,\ m)$

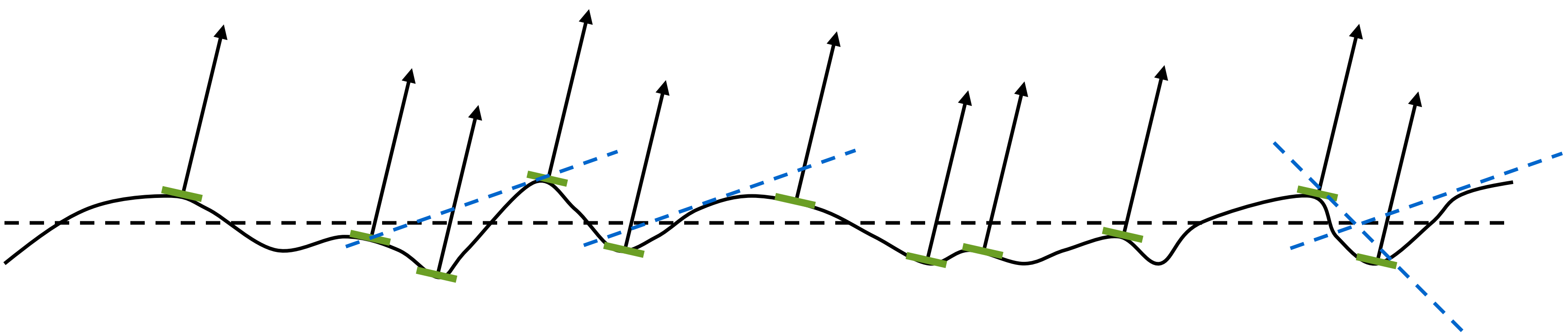




Masking effects

Shadow masking $G(\Psi, \Theta,\ m)$





Microfacet Models

Many different models have been proposed …

Torrance-Sparrow model:

$$f_{\mathrm{r}}(\mathrm{p}, \omega_{\mathrm{o}}, \omega_{\mathrm{i}}) = \frac{D(\omega_{\mathrm{h}})\ G(\omega_{\mathrm{o}}, \omega_{\mathrm{i}})\ F_{\mathrm{r}}(\omega_{\mathrm{o}})}{4\ \cos\theta_{\mathrm{o}}\ \cos\theta_{\mathrm{i}}}$$

$$D(\omega_{\mathrm{h}}) \propto (\omega_{\mathrm{h}} \cdot \mathbf{n})^{e}$$

$$G(\omega_{\mathrm{o}}, \omega_{\mathrm{i}}) = \min\left(1, \min\left(\frac{2(\mathbf{n}\cdot\omega_{\mathrm{h}})(\mathbf{n}\cdot\omega_{\mathrm{o}})}{\omega_{\mathrm{o}}\cdot\omega_{\mathrm{h}}}, \frac{2(\mathbf{n}\cdot\omega_{\mathrm{h}})(\mathbf{n}\cdot\omega_{\mathrm{i}})}{\omega_{\mathrm{o}}\cdot\omega_{\mathrm{h}}}\right)\right)$$

Microfacet Models

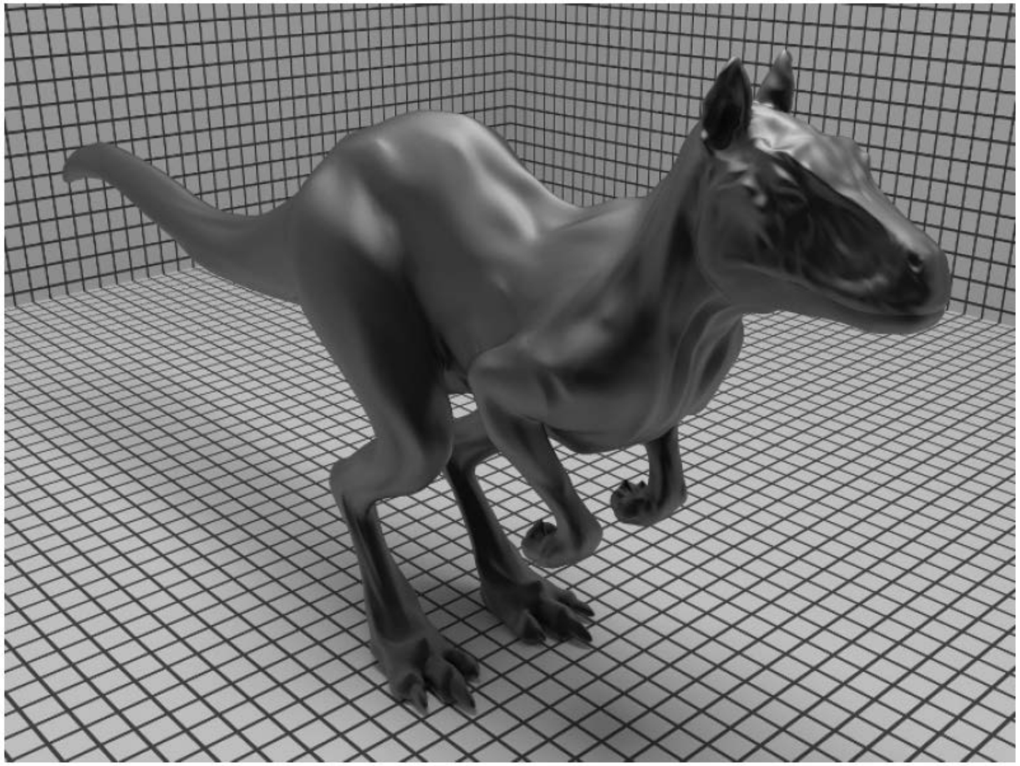


Figure 8.17: Killeroo model rendered with the Torrance–Sparrow microfacet model and Blinn microfacet distribution function. *(Model courtesy of headus/Rezard.)*

Measuring BRDFs

https://youtu.be/bRDf1Jj2cyY

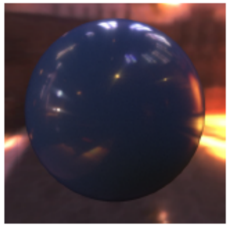
Measuring BRDFs

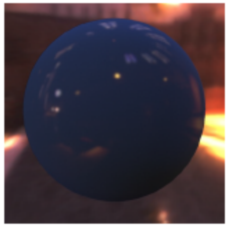
Material Name: acrylic-blue

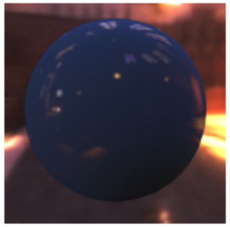
Fitted Parameters/Error

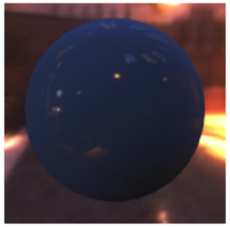
Model	d_r	d_g	d_b	s_r	s_g	s_b	p_0	p_1	p_2	Error
Ward	0.0138	0.0326	0.0636	0.00774	0.00547	0.00339	0.0162			0.0194
Ward-Duer	0.0137	0.0326	0.0637	0.00534	0.00364	0.00224	0.0162			0.0165
Blinn-Phong	0.0147	0.0332	0.064	0.0016	0.00115	0.000709	1.37e+004			0.0222
Lafortune et al.	0.0145	0.0332	0.064	0.0238	0.0164	0.01	-0.577	0.577	4.06e+003	0.0167
Cook-Torrance	0.0143	0.033	0.0639	0.0291	0.0193	0.0118	0.117	0.0137		0.0155
He et al.	0.0147	0.0334	0.0641	2.93	1.92	1.17	28.7	0.063	1.08	0.0141
Ashikhmin-Shirley	0.0143	0.0331	0.0639	0.0366	0.0241	0.0147	0.0949	1.16e+004		0.0153

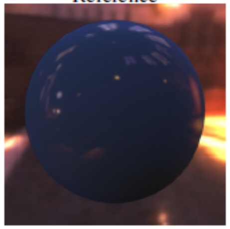
Rendered Images

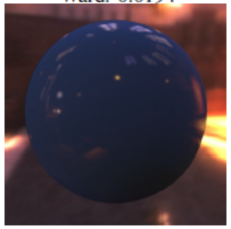

Reference

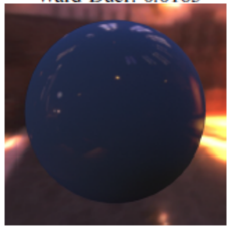

Ward: 0.0194

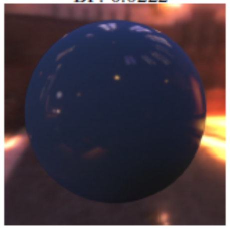

Ward-Duer: 0.0165


BP: 0.0222


Lafortune: 0.0167


CT: 0.0155


He: 0.0141


Ash: 0.0153

http://people.csail.mit.edu/addy/research/brdf/

Measuring BRDFs

An Adaptive Parameterization for Efficient Material Acquisition and Rendering

Jonathan Dupuy — Unity Technologies
Wenzel Jakob — EPFL

In *Transactions on Graphics (Proceedings of SIGGRAPH Asia 2018)*

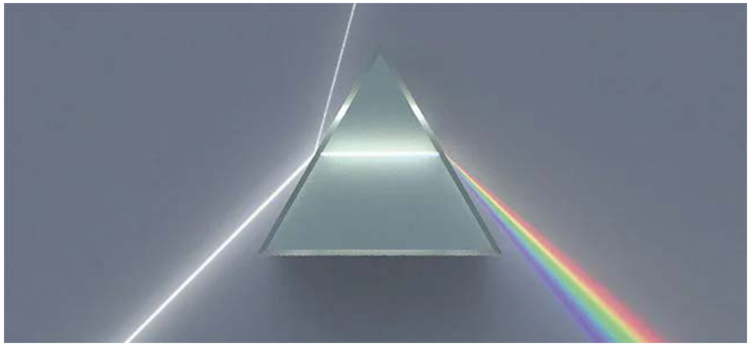
Spectral rendering of isotropic and anisotropic materials acquired from real-world samples using our method; insets show corresponding reflectance spectra. We measured these BRDFs using a motorized gonio-photometer, leveraging our novel adaptive parameterization to simultaneously handle BRDF acquisition, storage, and efficient Monte Carlo sample generation during rendering. Our representation requires 16 KiB of storage per spectral sample for isotropic materials and 544 KiB per spectral sample for anisotropic specimens.

http://rgl.epfl.ch/publications/Dupuy2018Adaptive

Spectral Rendering

Refraction is wavelength-dependent

- Refraction increases as the wavelength of light decreases
- Violet and Blue experience more “bending” than Orange and Red
- Often ignored in graphics



Other “wavelength effects”:

- Fluorescence: shift in reflected wavelength
- Phosphorescence: time-delay
- Iridiscence: color changes due to angles
- Opalescence, Pearlescence, …

Spectral Rendering

Rendering equation?

$$L_o(x,\omega_o,\lambda_o,t_o) = L_e(x,\omega_o,\lambda_o,t_o) + \int_0^{t_o}\int_\lambda\int_\Omega f(x,\omega_i,\omega_o,\lambda_i,\lambda_o,t_i,t_o)L_i(x,\omega_i,\lambda_i,t_i)\cos\theta\, d\omega_i\, d\lambda_i\, dt_i$$

Store multiple wavelengths per ray:
- As discrete samples (cfr rgb)
- As basis functions + weights defined over wavelength domain
- One ray is refracted into may rays, energy per wavelength might change
- Importance sampling techniques in wavelength domain

Light sources and BRDFs also need to be multi-spectral

At some point (file format, display, …) conversion torgb necessary

Spectral Rendering

Some examples from “Efficient Spectral Rendering on the GPU for Predictive Rendering”, 2021, https://hal.inria.fr/hal-03331619/document

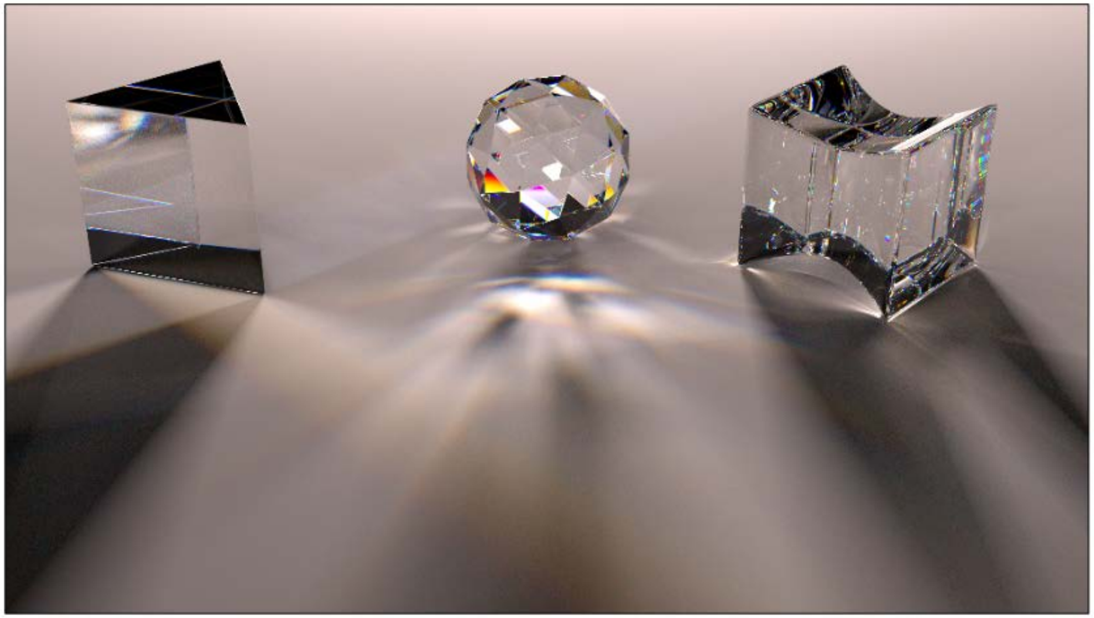
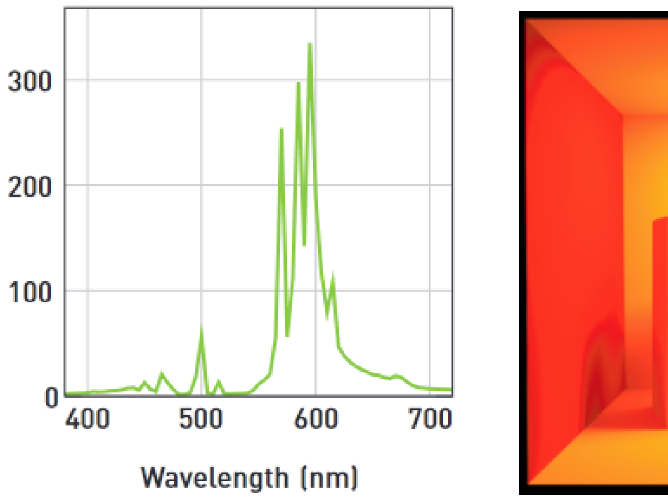


Figure 42-1. *Only a spectral renderer can simulate accurately the richness of color from wavelength-dependent effects, such as light dispersion.*

Spectral Rendering

Some examples from “Efficient Spectral Rendering on the GPU for Predictive Rendering”, 2021, https://hal.inria.fr/hal-03331619/document

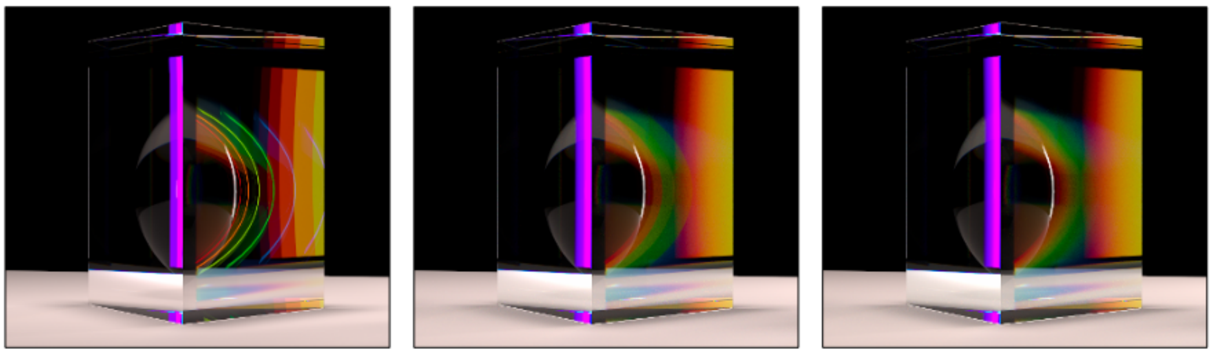


(a) *Illuminant* **(b)** *Tristimulus rendering* **(c)** *Spectral rendering*

Figure 42-3. *Discrepancies induced by a tristimulus renderer are even more prevalent in a global illumination context. This leads to important color and intensity differences, especially with narrow spectra like the one used to light this Cornell box scene (HP1, high-pressure vapor lamp).*

Spectral Rendering

Some examples from “Efficient Spectral Rendering on the GPU for Predictive Rendering”, 2021, https://hal.inria.fr/hal-03331619/document



(a) *Rendering with a discrete set of wavelengths*

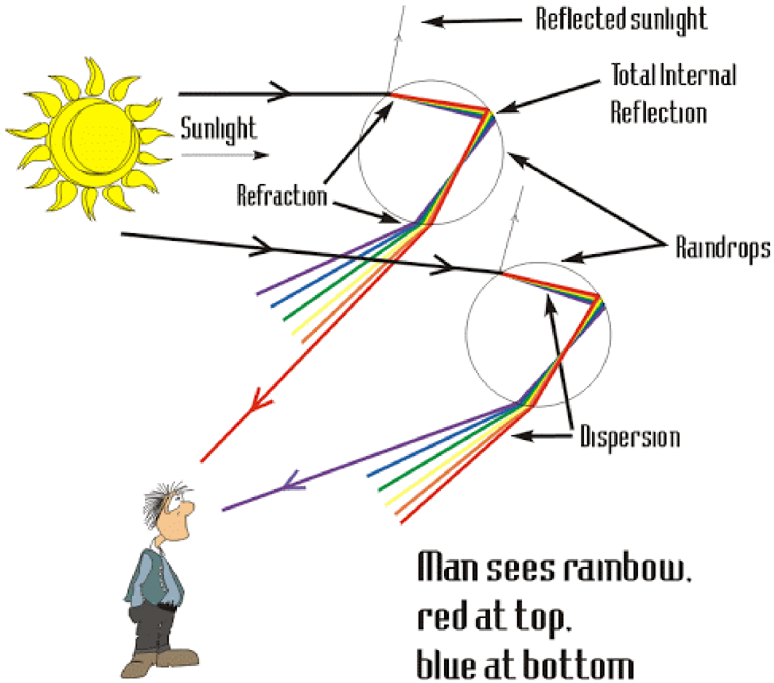
(b) *Sampling wavelength at each new sample; accumulation in nearest bins*

(c) *Sampling wavelength at each new sample; accumulation distributed in neighboring bins*

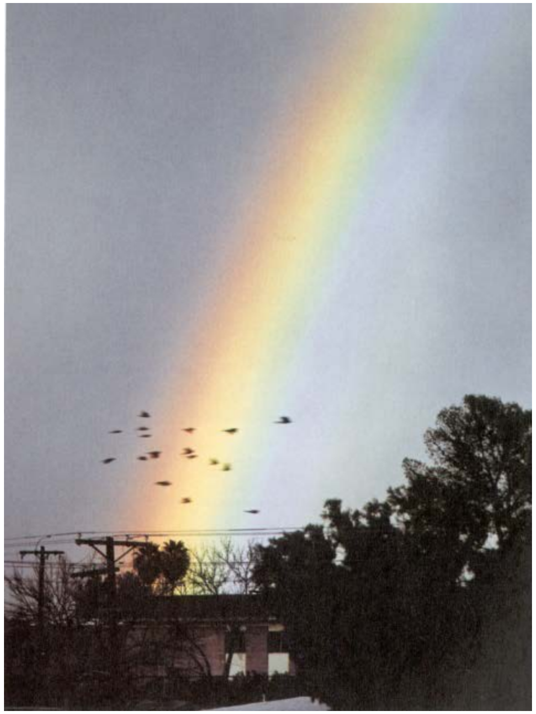
Figure 42-4. *This scene shows a prism made of a dispersive glass: it has a wavelength-dependent index of refraction (later detailed in Section 42.4). (a) A discrete set of wavelengths are used. This produces significant spectral banding artifacts because only a subset of paths is explored. (b) We jitter the wavelength within each bin for each sample and then accumulate the resulting radiance in the nearest bins. There is still some banding because the transitions between sampled bins are visible in the sphere reflection. (c) We use a different kernel to accumulate results in neighboring bins, thus further decreasing the hard transitions between bins.*

Spectral Rendering

Rainbow is caused by
refraction + internal reflection + refraction



http://www.rebeccapaton.net/rainbows/formatn.htm



From “Color and Light in Nature”
Lynch and Livingstone

Spectral Rendering

Physically-based Simulation of Rainbows

Iman Sadeghi[1] Adolfo Munoz[2] Philip Laven[3] Wojciech Jarosz[4,1] Francisco Seron[2] Diego Gutierrez[2] Henrik Wann Jensen[1]

[1]UC San Diego [2]Universidad de Zaragoza [3]Horley, UK [4]Disney Research Zürich

In *ACM Transactions on Graphics (Presented at SIGGRAPH)*, 2012

(a) (b) (c) (d) (e)

Our rendering results for different types of rainbows: (a) Rainbow derived from Lorenz-Mie theory. (b) Single primary rainbow with considering the angular view of the sun. (c) Double rainbow with a flipped secondary rainbow. (d) Multiple supernumerary rainbows caused by small water drops with uniform sizes. (e) Twinned rainbow resulted from mixture of non-spherical water drops and spherical ones.

https://cs.dartmouth.edu/wjarosz/publications/sadeghi11physically.html