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 12. 12. 2024

BRDF (sr^{-1})

Also note that we can rewritte the equation of BRDF

$$\mathrm{d}L_r(\boldsymbol{x},\omega_o) = f_r(\omega_i,\omega_o)L_i(\boldsymbol{x},\omega_i)\cos\theta_i\,\mathrm{d}\omega_i$$

which can be integrated to obtain reflected radiance

$$L_r(\boldsymbol{x},\omega_o) = \int_H f_r(\omega_i,\omega_o)L_i(\boldsymbol{x},\omega_i)\cos\theta_i\,\mathrm{d}\omega_i$$

- (Physically plausible) Bidirectional Reflectance Distribution Function

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

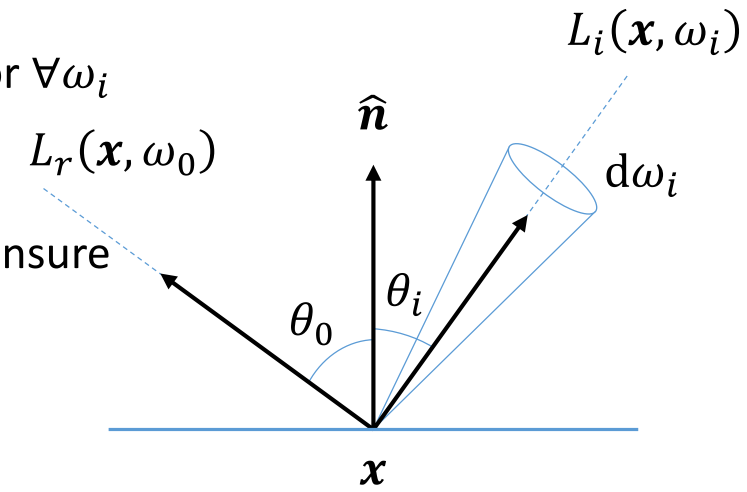
Weakening of irradiance due to incident angle (the Lambert's cosine law)

II. Energy conservation

$\rho(\omega_i) = \int_H f_r(\omega_i,\omega_o)\cos\theta_o\,\mathrm{d}\omega_o \leq 1$ for $\forall\omega_i$

The fraction of light coming in from any direction that is reflected in the entire hemisphere should be smaller than 1 to ensure conservation of energy.

Here, the ρ is the total hemispherical reflectivity.



III. Positivity

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

Basic BRDFs

- Perfect mirror

$$f_r^{Mirror}(\omega_i, \omega_o) = \begin{cases} \infty \text{ if } \theta_i = \theta_0 \\ 0 \text{ otherwise} \end{cases}$$

- Perfect diffusor (Lambertian surface)

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

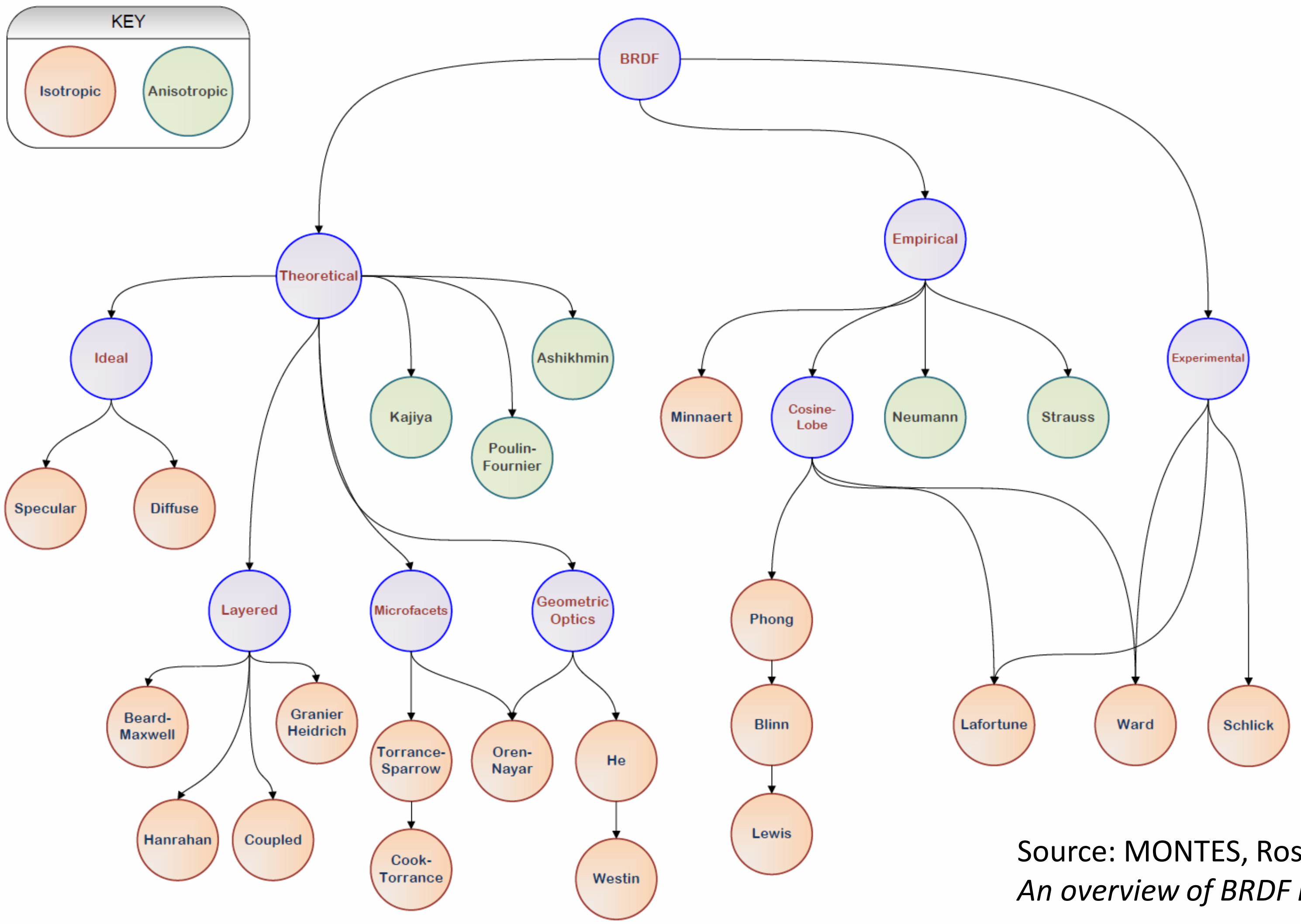
Albedo is the ration of outgoing and incoming flux

- Modified Phong (physically correct but still empirical model)

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

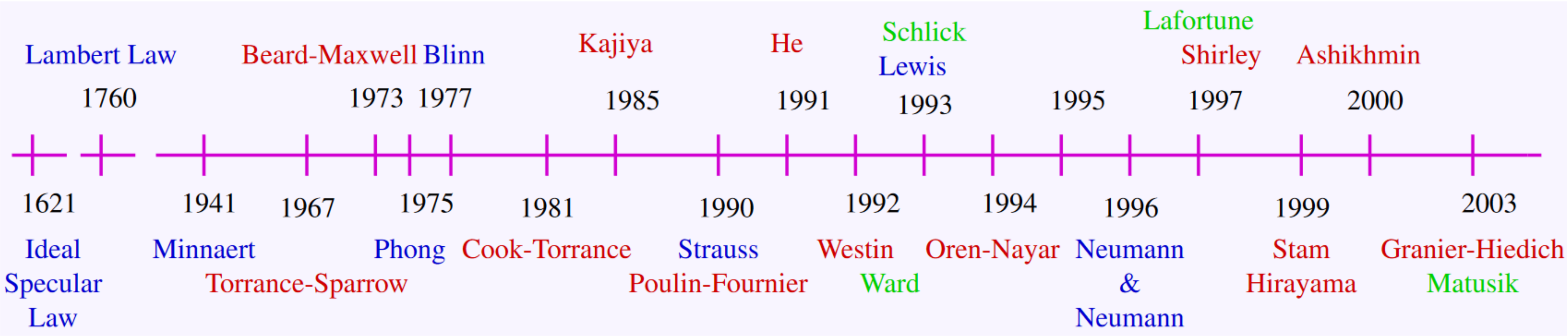
φ is angle between ω_o and perfect specular reflection of ω_i

BRDFs



Source: MONTES, Rosana; UREÑA, Carlos. *An overview of BRDF models*. 2012.

BRDFs Time Line



Source: MONTES, Rosana; UREÑA, Carlos. *An overview of BRDF models*. 2012.

General classification of BRDFs

- **Empirical**

Their main aim is to provide a simple formulation specifically designed to mimic a kind of reflection. Consequently, we get a fast computational model adjustable by parameters, but without considering the physics behind it.

- **Theoretical**

These models try to accurately simulate light scattering by using physics laws. They usually lead to complex expression and high computational effort, thus they are not normally employed in rendering systems.

- **Experimental**

The BRDF can be acquired using a gonioreflectometer which mechanically varies light source and sensor positions. This proces could take hours and usually data is limited by some angular resolution. Other techniques use digital cameras to acquire many BRDF samples with a single photograph. No much densely acquired data is readily available.

Experimental BRDFs - Gonioreflectometer

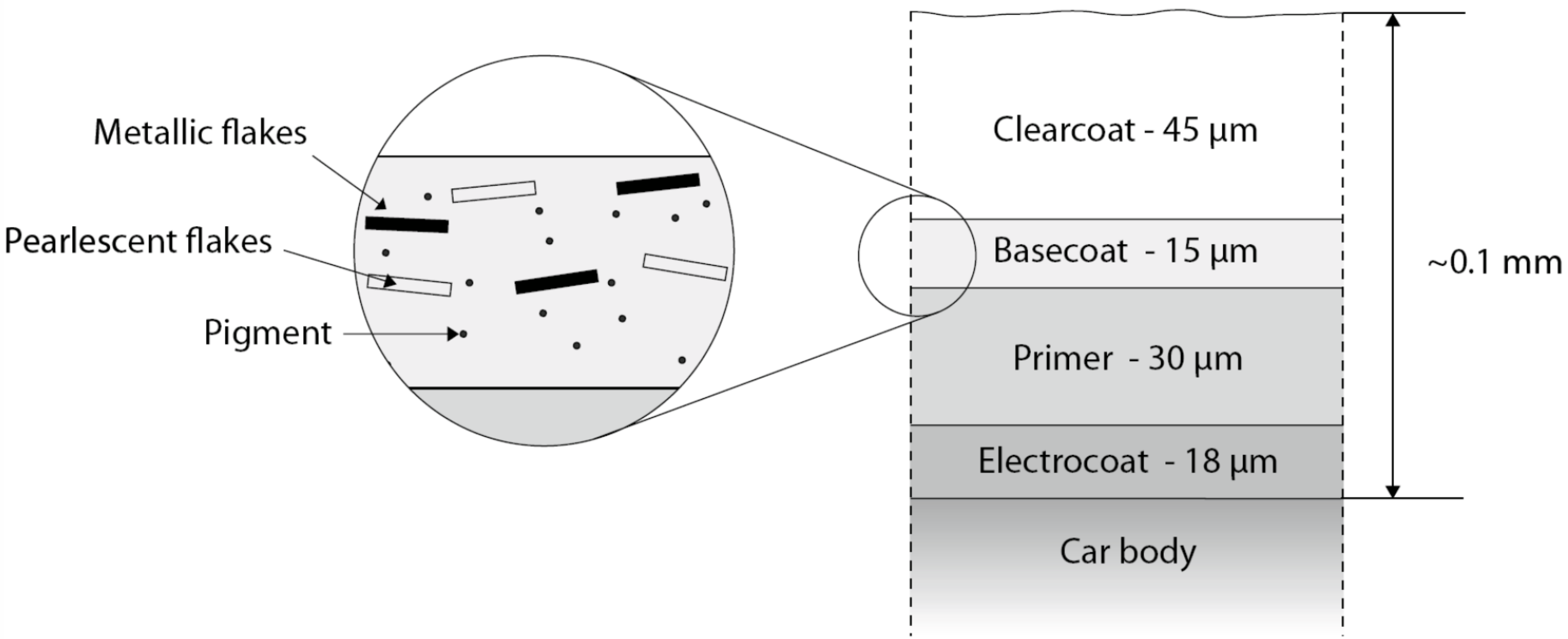
Source: Biliouris, D. et al. A Compact Laboratory Spectro-Goniometer to Assess the BRDF of Materials. *Sensors, 7*(9), pp. 1846-1870, 2007.

Models	Physical	Plausible	Fresnel Eq.	Anisototropic	Sampling	Rel.Cost (cycles)	Material Type
Ideal Specular	★	★	▼	▼	★	x	perfect specular
Ideal Diffuse	★	★	▼	▼	★	x	perfect diffuse
Minnaert	▼	· · ·	▼	▼	▼	$5.35x$	Moon surf.
Torrance-Sparrow	★	▼	★	★	▼	· · ·	rough surf.
Beard-Maxwell	★	▼	★	▼	▼	$397x$	painted surf.
Blinn-Phong	▼	▼	▼	▼	★	$9.18x$	rough surf.
Cook-Torrance	★	★	★	▼	▼	$16.9x$	metal,plastic
Kajiya	★	▼	★	★	▼	· · ·	metal,plastic
Poulin-Fournier	★	▼	▼	★	▼	$67x$	clothes
Strauss	▼	· · ·	★	▼	▼	$14.88x$	metal,plastic
He et al.	★	★	★	▼	▼	$120x$	metal
Ward	▼	▼	▼	★	★	$7.9x$	wood
Westin	★	· · ·	★	★	▼	· · ·	metal
Lewis	▼	★	▼	▼	★	$10.73x$	mats
Schlick	▼	★	★	★	▼	$26.95x$	heterogeneous
Hanrahan	★	· · ·	★	▼	▼	· · ·	human skin
Oren-Nayar	★	★	▼	▼	★	$10.98x$	matte, dirty.
Neumann	▼	★	▼	★	★	· · ·	metal,plastic
Lafortune	▼	★	▼	★	★	$5.43x$	rough surf.
Coupled	★	★	★	▼	★	$17.65x$	polished surf.
Ashikhmin-Shirley	★	▼	★	★	★	$79.6x$	polished surf.
Granier-Heidrich	★	· · ·	★	▼	▼	· · ·	old-dirty metal

Table 1: *Brief summary of the properties exhibited by the reviewed BRDFs. Legend: (★) if the BRDF has this property; (▼) if the BRDF does not; (· · ·) unknown value.*

Source: MONTES, Rosana; UREÑA, Carlos. *An overview of BRDF models*. 2012.

Car Paint



Scheme illustrating the layers inside a metallic paint with multiple types of flakes

Source: SMETANA, Róbert. *Overview of reflectance models focused on car paint simulation*. 2008.

Modified Phong BRDF

Note that the normalization factors are omitted here for brevity. See later slides.

- Original formulation of Phong reflection model (only specular part) is

$$L_o(\boldsymbol{x}, \omega_o) = \cdots + \rho_s (\cos\theta_r)^\gamma L_i(\boldsymbol{x}, \omega_i) \ ,$$

where $\omega_r = \mathrm{reflect}(\omega_o, \hat{\boldsymbol{n}})$, $\cos\theta_r = \omega_i \cdot \omega_r$ and γ is Phong exponent, thus the **original** Phong BRDF must read as $f_r^P(\omega_i, \omega_o) = (\cos\theta_r)^\gamma / \cos\theta_i$

- When we plug this into the rendering equation, we get the following

$$L_o(\boldsymbol{x}, \omega_o) = \cdots + \int \boxed{\frac{\rho_s (\cos\theta_r)^\gamma}{\cancel{\cos\theta_i}}} L_i(\boldsymbol{x}, \omega_i) \cancel{\cos\theta_i} \mathrm{d}\omega_i \ .$$

- The problem is that such a BRDF does not take into account the amount of light arriving at the infinitesimal patch around point $\boldsymbol{x}$ due to the missing term $\cos\theta_i$.
- Thus the modified Phong illumination formula must read as

$$L_o(\boldsymbol{x}, \omega_o) = \cdots + \rho_s (\cos\theta_r)^\gamma L_i(\boldsymbol{x}, \omega_i) \cos\theta_i \ .$$

- Subsequently, we get the **modified Phong BRDF** as $f_r^M(\omega_i, \omega_o) = \boldsymbol{\rho_s (\cos\theta_r)^\gamma}$, not $\frac{\rho_s (\cos\theta_r)^\gamma}{\cos\theta_i}$.

Modified Phong BRDF

- Lafortune and Willems (1994)
- $f_r^M(\omega_i, \omega_o) = k_d + k_s^M = \frac{\rho_d}{\pi} + \frac{\rho_s(\gamma+2)}{2\pi}(\cos\theta_r)^\gamma$

where $\omega_r = \mathrm{reflect}(\omega_o, \hat{\boldsymbol{n}})$, $\cos\theta_r = \omega_i \cdot \omega_r$ and γ is Phong exponent

- Also remember that $\rho_d + \rho_s \leq 1$ should hold for all channels/frequencies. We may also use the Fresnel reflectance instead of ρ_s (see later slides)

- BRDF-proportional importance sampling:
- 1. $\xi_0 \in \langle 0, \max(\rho_d) + \max(\rho_s))$
- 2. if $\xi_0 < \max(\rho_d)$ sample from diffuse (use cos-weighted hemisphere sampling) otherwise sample from specular component…

Specular/Cosine-lobe Sampling

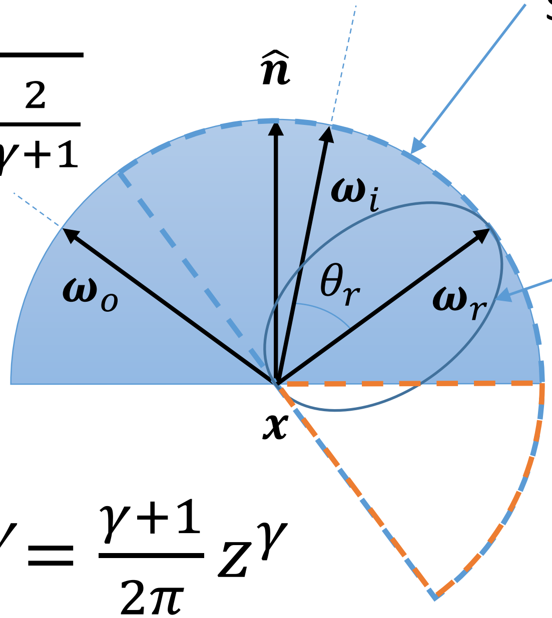
- Random point on a unit sphere (i.e. direction) $\omega_i = (x, y, z)$ where

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

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

$$z = {\xi_2}^{\frac{1}{\gamma+1}}$$

$$\mathrm{pdf}(\omega_i) = \frac{\gamma+1}{2\pi}(\cos\theta_r)^{\gamma} = \frac{\gamma+1}{2\pi} z^{\gamma}$$

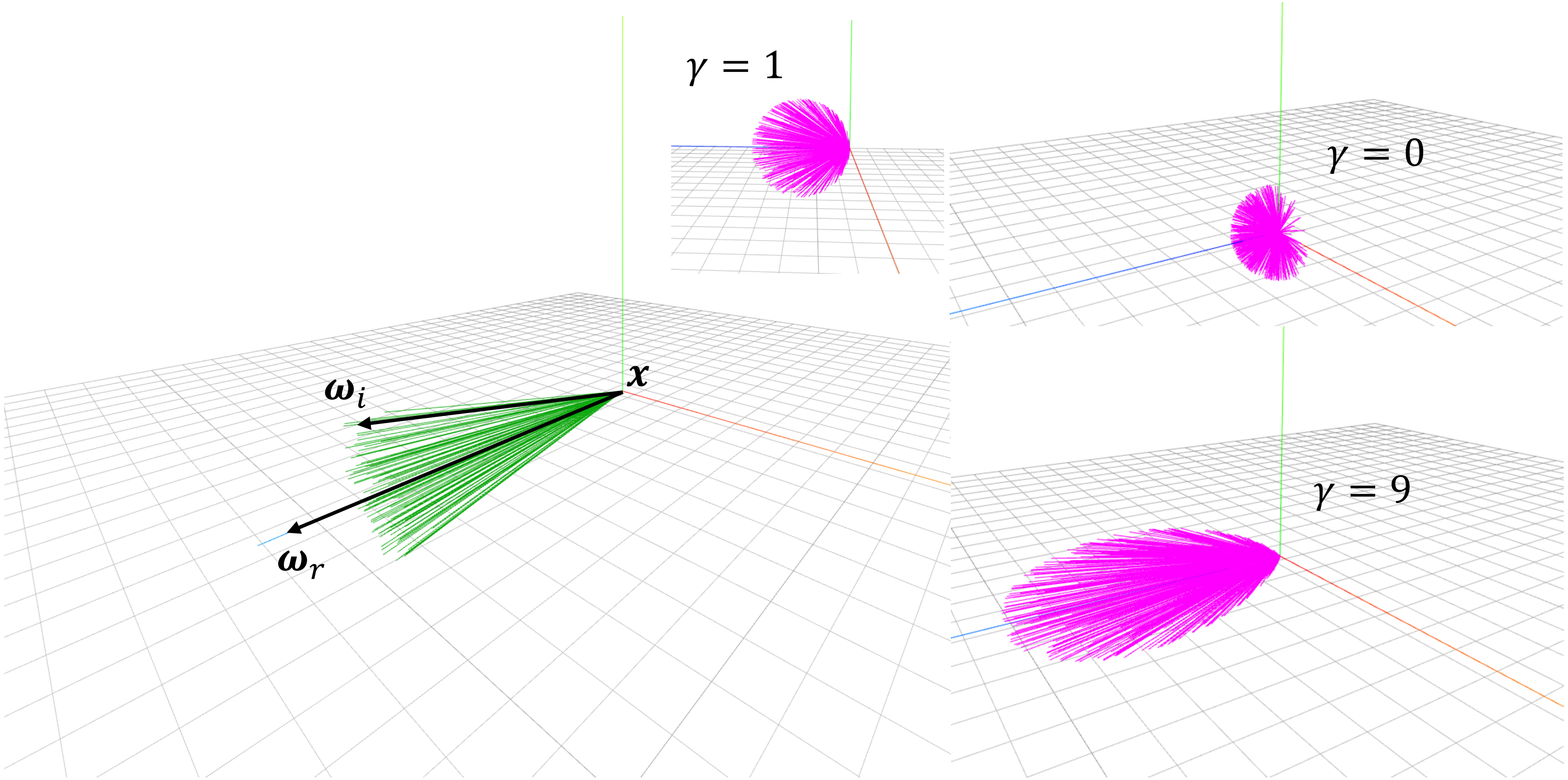


Sampled hemisphere of specular lobe

PDF of specular (cosine)-lobe around $\boldsymbol{\omega}_r$

Some samples may be directed toward the surface, in which case $f_r^{Phong} = 0$ but do not remove these samples

Specular/Cosine-lobe Sampling



Note that $\int_H f_r(\omega_i, \omega_o) \cos\theta_o \, \mathrm{d}\omega_o \le 1$ for $\forall \omega_i$

Original Phong BRDF Energy-conserving

- The derivation of energy conservation term for the original Phong BRDFs is widely known
- $I_P = ?$ when $f_r^P = \frac{\rho_s}{I_P} \frac{\max(0,(\cos\theta_r)^\gamma)}{\cos\theta_i}$ (original formulation)

$$1 = \int_H \frac{\rho_s}{I_P} \frac{\max(0,(\cos\theta_r)^\gamma)}{\cancel{\cos\theta_i}} \cancel{\cos\theta_i} \, \mathrm{d}\omega_i \left| \begin{array}{c} \text{set } \rho_s = 1 \\ \text{for } \omega_o = \hat{\boldsymbol{n}}\text{: } \theta_r = \theta_i \end{array} \right\}$$

simplified conditions maximizing the integral

$$= \frac{1}{I_P} \int_H (\cos\theta_i)^\gamma \mathrm{d}\omega_i = \frac{1}{I_P} \int_0^{2\pi} \int_0^{\frac{\pi}{2}} (\cos\theta_i)^\gamma \sin\theta_i \, \mathrm{d}\theta_i \mathrm{d}\varphi_i = \frac{1}{I_P} \frac{2\pi}{\gamma + 1} \Rightarrow$$

$$\frac{1}{I_P} = \frac{\gamma + 1}{2\pi}$$

Note that $\int_H f_r(\omega_i, \omega_o) \cos\theta_o \, \mathrm{d}\omega_o \leq 1$ for $\forall \omega_i$

Modified Phong BRDF Energy-conserving

- The derivation of energy conservation term for the modified Phong BRDFs is also widely known
- $I_M = ?$ when $f_r^M = \frac{\rho_s}{I_M} \max(0, (\cos\theta_r)^\gamma)$ (modified formulation)

$$1 = \int_H \frac{\rho_s}{I_M} \max(0, (\cos\theta_r)^\gamma) \cos\theta_i \, \mathrm{d}\omega_i \Bigg| \begin{array}{c} \text{set } \rho_s = 1 \\ \text{for } \omega_o = \hat{\boldsymbol{n}}\text{: } \theta_r = \theta_i \end{array}$$

simplified conditions maximizing the integral

$$= \frac{1}{I_M} \int_H (\cos\theta_i)^{\gamma+1} \mathrm{d}\omega_i = \frac{1}{I_M} \int_0^{2\pi} \int_0^{\frac{\pi}{2}} (\cos\theta_i)^{\gamma+1} \sin\theta_i \, \mathrm{d}\theta_i \mathrm{d}\varphi_i = \frac{1}{I_M} \frac{2\pi}{\gamma+2} \Rightarrow$$

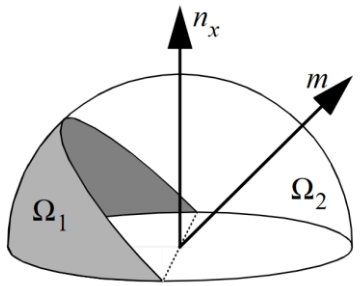
$$\frac{1}{I_M} = \frac{\gamma+2}{2\pi}$$

Modified Phong BRDF Energy-normalization

- It is well-known how to make modified Phong BRDF conserve energy (never gain energy) but making it energy-normalized (never lose nor gain energy) is far more difficult. Note that energy normalization is a stronger criterion than energy conservation, and is arguably part of the definition of the BRDF itself
- Some methods for energy-normalization are available
  - Method described in [1] requires the specular exponent n (or γ) to be integer-valued, and have $O(n)$ runtime cost
  - Method devised in [2] generalizes to the real-valued specular exponent case and attain $O(1)$ runtime cost

[1] ARVO, James. Applications of irradiance tensors to the simulation of non-lambertian phenomena. In: *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. 1995. p. 335-342.
[2] MALLETT, Ian; YUKSEL, Cem. Constant-time energy-normalization for the Phong specular BRDFs. *The Visual Computer*, 2020, p. 2029-2038.

Arvo's Method for (Modified) Phong BRDF



$$\int_{\Omega_2} \cos^n(\Theta, m)\cos(\Theta, n_x)\,d\omega_\Theta$$

procedure ARVOINTEGRATEMODIFIEDPHONG(N, ω_i, n)
  $c \leftarrow (N \bullet \omega_i)$ c is $\cos\theta_o$ $n = \text{round}(\gamma)$
  $s \leftarrow \sqrt{1-c^2}$
  if n even **then**
    $A \leftarrow \pi/2$
    $k \leftarrow 0$
    $T_k \leftarrow \pi/2$
  else
    $A \leftarrow \pi - \arccos(c)$
    $k \leftarrow 1$
    $T_k \leftarrow s$
  end if
  $t \leftarrow 0$
  $\boxed{\textbf{while } k \le n-2 \textbf{ do}}$ $O(n)$
    $t \leftarrow t + T_k$
    $k \leftarrow k+2$
    $T_k \leftarrow s^2\,((k-1)/k)\,T_k$
  end while
  $I_M \leftarrow 2\,(\underbrace{T_k + A\,c + c^2\,t}_{\le \pi})\,/\,(n+2)$
  return $\boxed{I_M}$
end procedure

$1/I_M$

Use $1/I_M$ value instead of the standard normalizer (red part) which is energy-conserving only

$$\rho_s \boxed{\frac{(\gamma+2)}{2\pi}} (\cos\theta_r)^\gamma$$

Mallett's and Yuksel's Method for (Modified) Phong BRDF

- To use this method, you need to provide the full (non-normalised) incomplete beta function implementation
- For instance, it is available in the Boost library

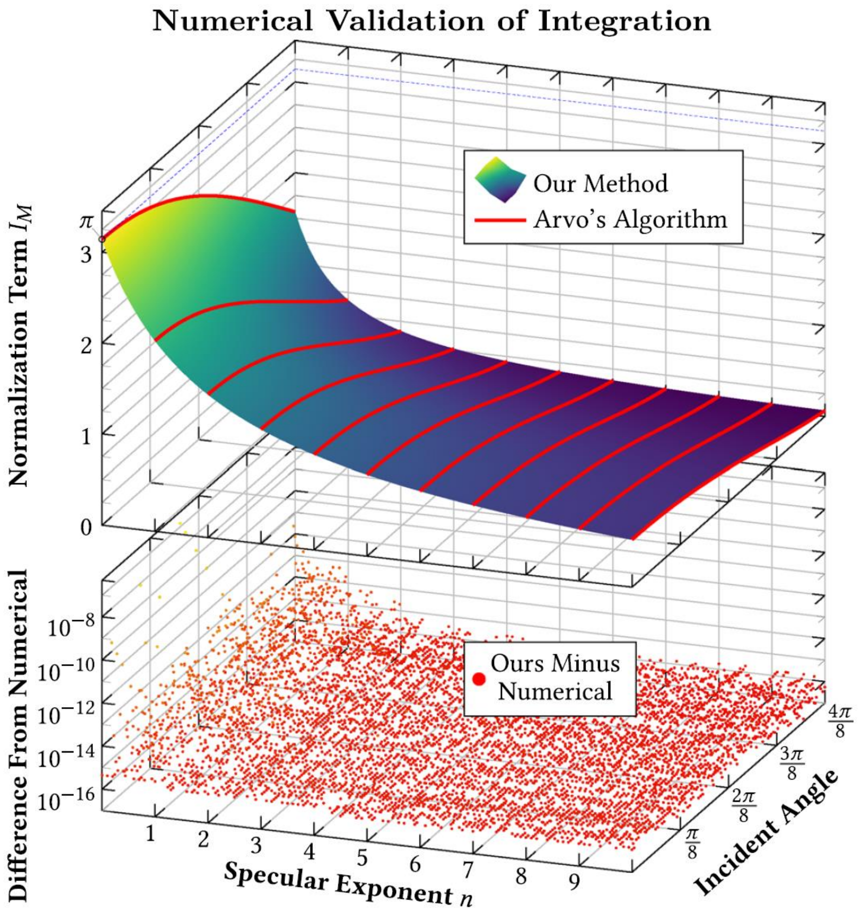
```
#include <boost/math/special_functions/beta.hpp >
float ibeta( float x, float a, float b ) {
  return boost::math::beta( a, b, x );
}
```

- Quick but dirty solution is to provide a numerical integration of

$$\boxed{B(x,a,b) = \int_0^x u^{a-1}(1-u)^{b-1}\mathrm{d}u}$$

```
#include <cmath>

#define TWO_PI   6.2831853f
#define SQRT_PI 1.77245385f

//Incomplete Beta function
float ibeta( float x, float a,float b );

//Computes Γ(a)/Γ(b)
inline static float gamma_quot( float a, float b ) {
    return std::exp( std::lgamma(a) - std::lgamma(b) );
}

inline static float calc_I_M( float NdotV, float n ) {
    float const& costerm     = NdotV;
    float        sinterm_sq = 1.0f - costerm*costerm;
    float        halfn       = 0.5f * n;

    float negterm = costerm;
    if (n>=1e-18f)
        negterm*=halfn*ibeta( sinterm_sq, halfn,0.5f );

    return (
        TWO_PI*costerm +
        SQRT_PI*gamma_quot( halfn+0.5f, halfn+1.0f )*
            ( std::pow(sinterm_sq,halfn) - negterm )
    ) / (n+2.0f);
}
```

Mallett's and Yuksel's Method for (Modified) Phong BRDF

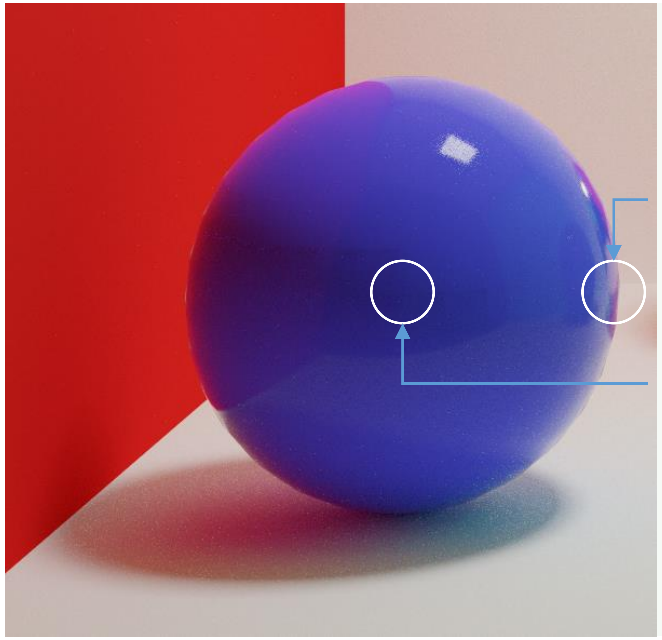


Arvo's algorithm
can normalize only at integer specular exponents. Second method works for
any specular exponent

Both methods ensure that the (modified) Phong BRDF passes the white furnace test

Fresnel Reflectance

- Predict the reflectance of smooth surfaces, which depends solely on the refractive index and the angle of incidence. For conductive materials (i.e. metals), the effect of Fresnel reflectance is subtle and for dielectrics is very strong but for both types of materials, at grazing angles, the reflectance reaches 100 %

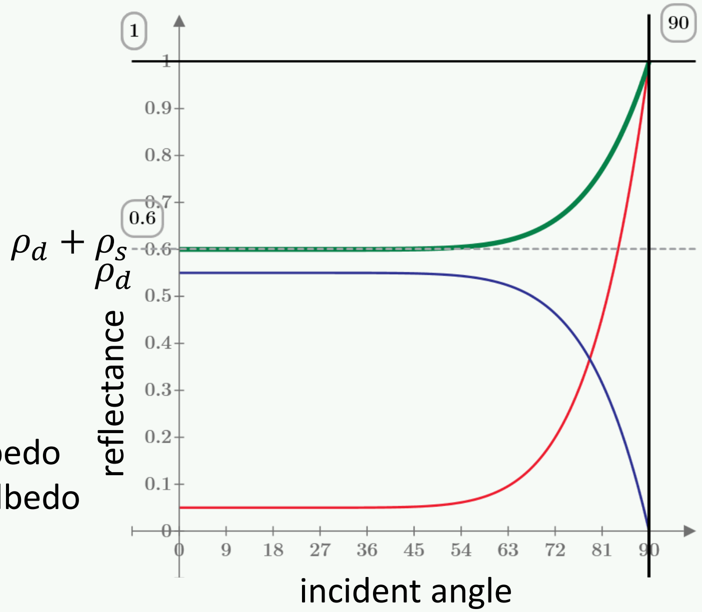


100 %

specular reflectance

5 %

$\rho_d = 0.55$ diffuse albedo
$\rho_s = 0.05$ specular albedo
energy conserving iff
$\rho_d + \rho_s = 0.6 \leq 1$



$$F(F_0, \theta) = F_0 + (1 - F_0)(1 - \cos(\theta))^5$$

$$R_d(F_0, \theta) = \frac{1 - \max(F(F_0, \theta))}{1 - \max(\rho_s)} \rho_d$$

$F(\rho s, \theta)$ specular reflectance

$Rd(\rho s, \theta)$ diffuse reflectance

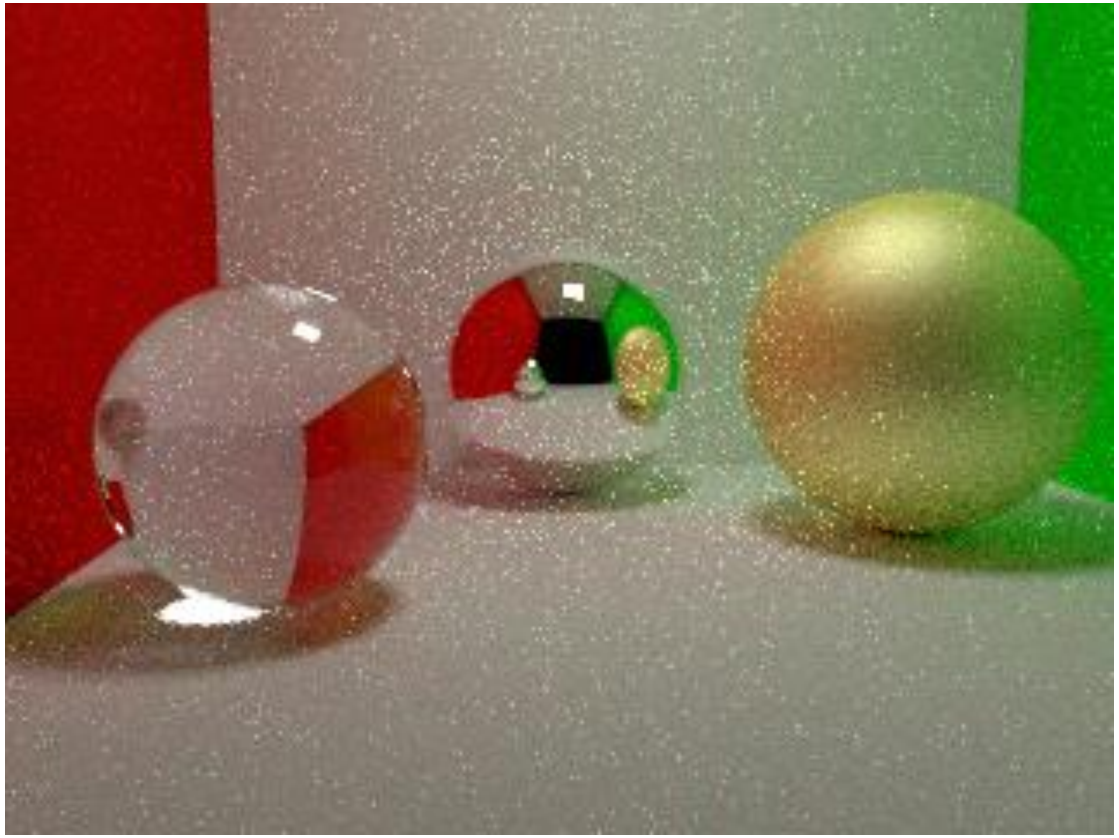
$F(\rho s, \theta) + Rd(\rho s, \theta)$ total reflectance

Modified Phong with Fresnel Term

- The final energy-normalized formula of the modified Phong BRDF with Fresnel reflectance

$$f_r^M(\omega_i, \omega_o) = k_d + k_s^M = \frac{R_d(\rho_s, \theta_o)}{\pi} + \frac{F(\rho_s, \theta_o)}{I_M}(\cos\theta_r)^\gamma$$

Phong BRDF (PT + RR + NEE + BRDF IS)

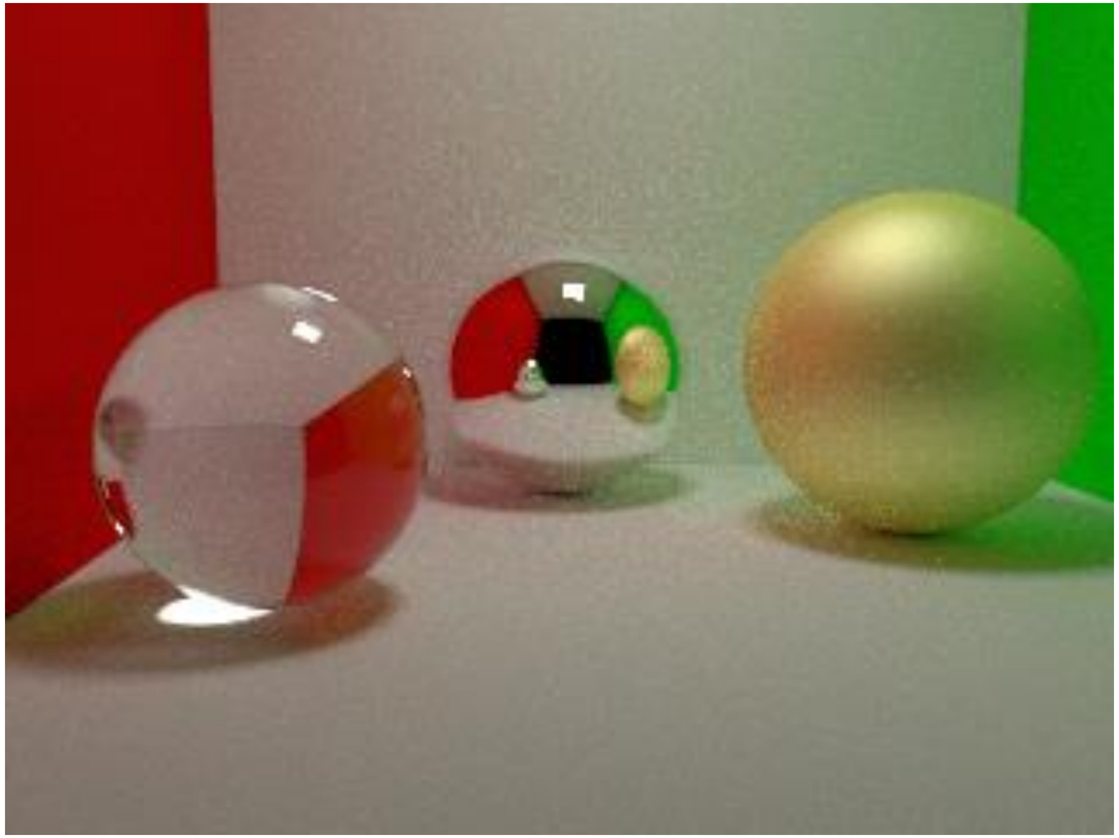


Note the white dots scattered all around the rendered image.

So called fireflies are caused by paths hitting the diffusive surfaces near light sources or caustics.

200 spp

Phong BRDF (PT + RR + NEE + BRDF IS)



Fireflies will disappear with the (much) higher amount of samples.

Faster solution:

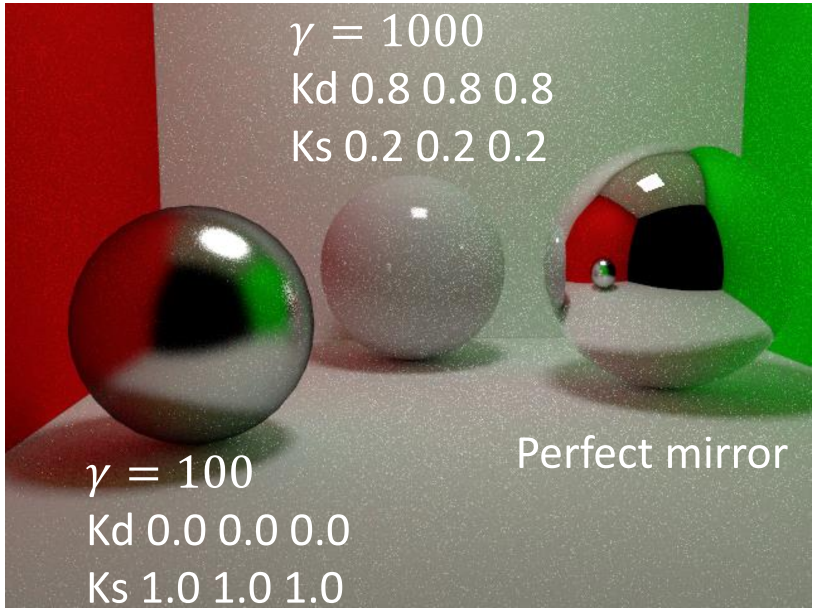
```
if len(pixel) > T then
pixel = normalize(pixel) * T
```

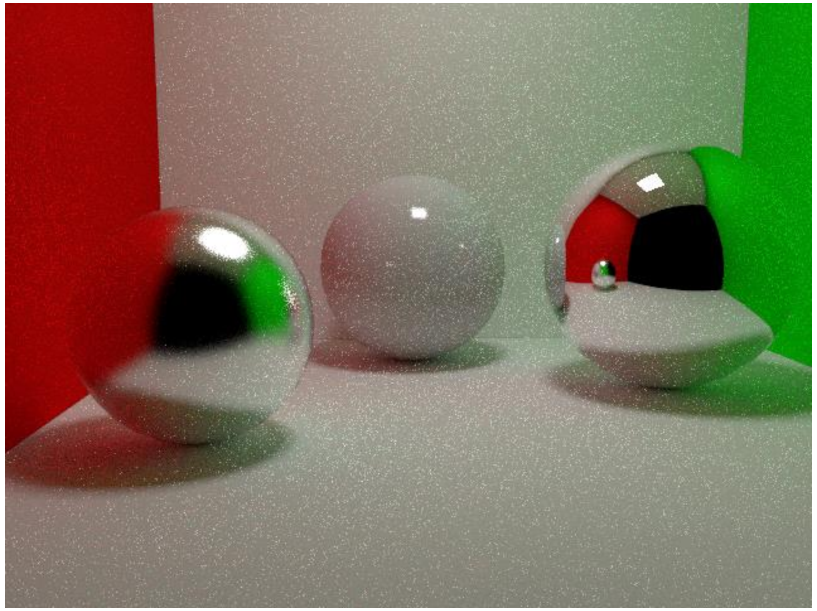
1.000 spp

Artefacts and Errors

Here, the bright spots aligned with the triangle edges are caused by non-unit normals.

Artefacts and Errors





When not using a proper normalizer, we may add the cosine denominator to the specular part of the Phong BRDF (as in the original unmodified formula) to remove the unwanted darkening around the sphere borders

$$k_s^{Phong}(\omega_i, \omega_o) = \frac{\rho_s(\gamma+1)}{2\pi}\frac{(\cos\theta_r)^\gamma}{\cos\theta_i}$$

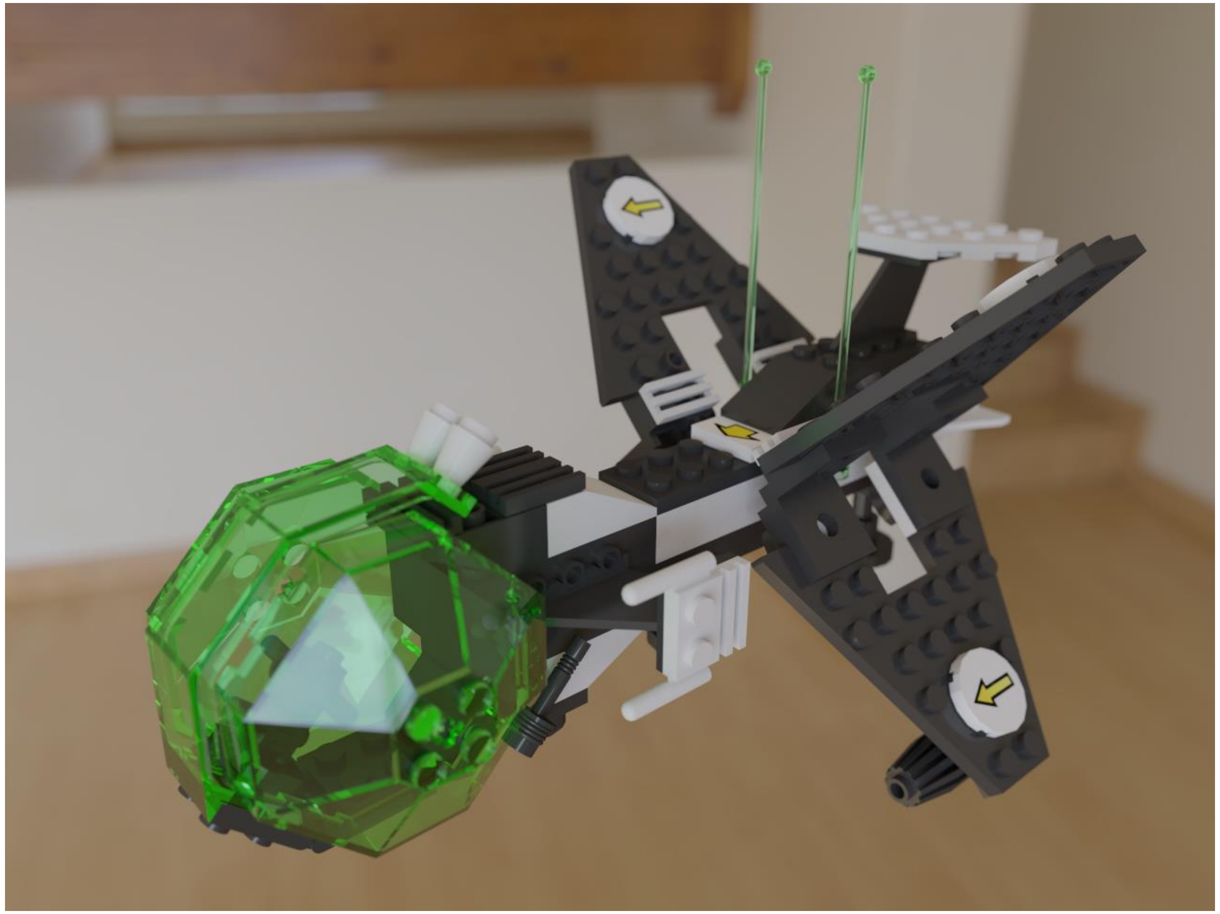
Examples of 4 BRDFs and 1 BSDF

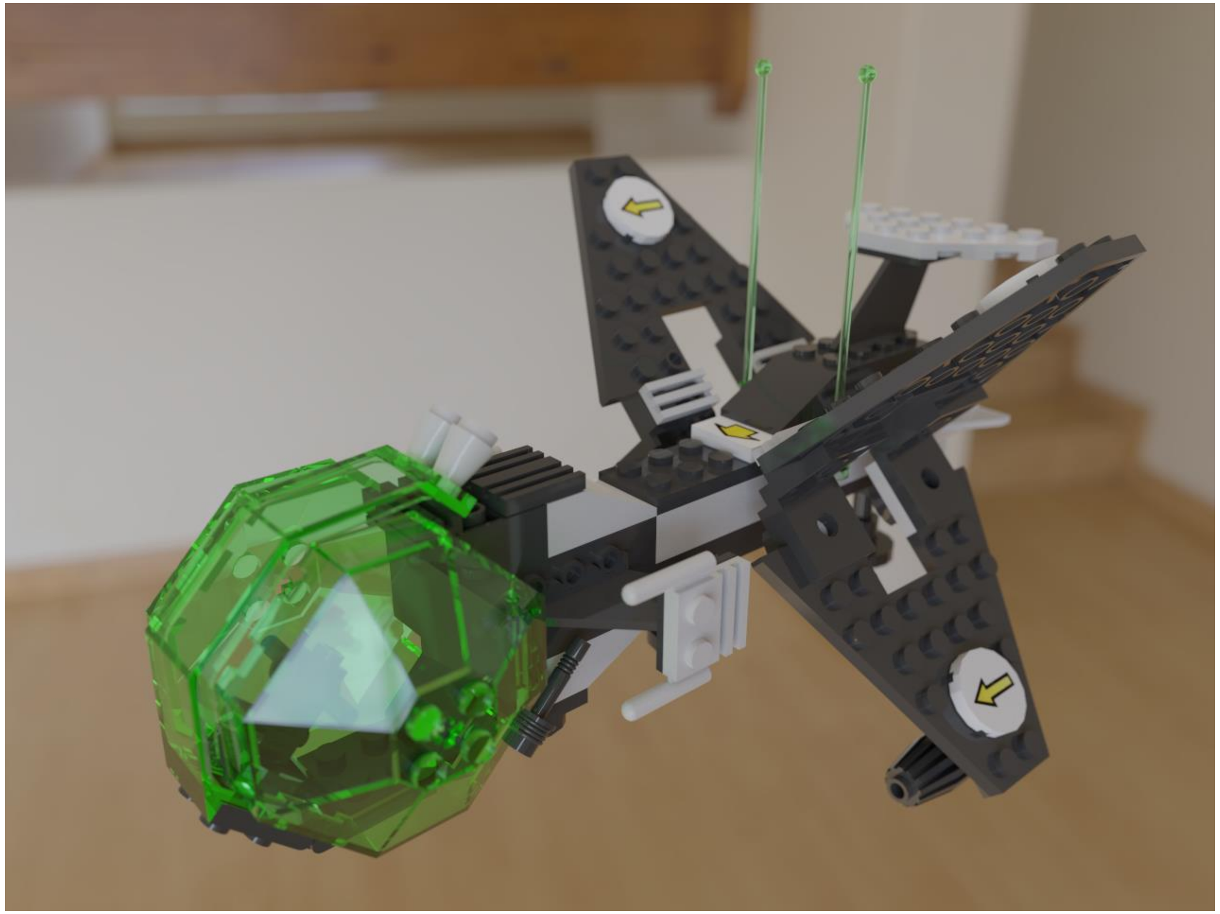
```
newmtl Mat001
  Ni 1.491
  Tf 0.01 0.01 0.01
  Ke 0 0 0
  shader 4

newmtl Mat002
  Ni 1.491
  Tf 0.75 0.75 0.75
  Ke 0 0 0
  shader 4

newmtl Mat003
  Ni 1.491
  Tf 1.5 1.5 1.5
  Ke 0 0 0
  shader 4

newmtl Mat004
  Ni 1.491
  Tf 3.0 3.0 3.0
  Ke 0 0 0
  shader 4

newmtl Mat005
  Ni 1.491
  Tf 0.15 0.01 0.96
  Ke 0 0 0
  shader 4

newmtl Mat006
  Kd 0.1 0.1 0.1
  Ke 0 0 0
  shader 2

newmtl Mat007
  Kd 0.5 0.5 0.5
  Ke 0 0 0
  shader 2

newmtl Mat008
  Kd 0.99 0.99 0.99
  Ke 0 0 0
  shader 2

newmtl Mat009
  Ns 0
  Ks 0.25 0.25 0.25
  Kd 0.25 0.25 0.25
  Ke 0 0 0
  shader 5

newmtl Mat010
  Pr 0.99
  Ks 0.5 0.5 0.5
  Kd 0.5 0.5 0.5
  Ke 0 0 0
  shader 8

newmtl Mat011
  Ns 1
  Kd 0.9 0.05 0.05
  Ks 0.05 0.05 0.05
  Ke 0 0 0
  shader 5

newmtl Mat012
  Ns 10
  Kd 0.9 0.05 0.05
  Ks 0.05 0.05 0.05
  Ke 0 0 0
  shader 5

newmtl Mat013
  Ns 100
  Kd 0.9 0.05 0.05
  Ks 0.05 0.05 0.05
  Ke 0 0 0
  shader 5

newmtl Mat014
  Ns 1000
  Kd 0.9 0.05 0.05
  Ks 0.05 0.05 0.05
  Ke 0 0 0
  shader 5

newmtl Mat015
  Ns 10000
  Kd 0.9 0.05 0.05
  Ks 0.05 0.05 0.05
  Ke 0 0 0
  shader 5

newmtl Mat016
  Pr 1.0
  Kd 0.9 0.05 0.05
  Ks 0.05 0.05 0.05
  Ke 0 0 0
  shader 8

newmtl Mat017
  Pr 0.7
  Kd 0.9 0.05 0.05
  Ks 0.05 0.05 0.05
  Ke 0 0 0
  shader 8

newmtl Mat018
  Pr 0.4
  Kd 0.9 0.05 0.05
  Ks 0.05 0.05 0.05
  Ke 0 0 0
  shader 8

newmtl Mat019
  Pr 0.1
  Kd 0.9 0.05 0.05
  Ks 0.05 0.05 0.05
  Ke 0 0 0
  shader 8

newmtl Mat020
  Pr 0.01
  Kd 0.9 0.05 0.05
  Ks 0.05 0.05 0.05
  Ke 0 0 0
  shader 8

newmtl Mat021
  Ks 0.99 0.99 0.99
  Ke 0 0 0
  shader 6

newmtl Mat022
  Ks 0.5 0.5 0.5
  Ke 0 0 0
  shader 6

newmtl Mat023
  Pr 0.7
  Kd 0.0 0.0 0.0
  Ks 1.0 0.766 0.336
  Ke 0 0 0
  shader 8

newmtl Mat024
  Pr 0.3
  Kd 0.0 0.0 0.0
  Ks 1.0 0.766 0.336
  Ke 0 0 0
  shader 8

newmtl Mat025
  Pr 0.1
  Kd 0.0 0.0 0.0
  Ks 1.0 0.766 0.336
  Ke 0 0 0
  shader 8
```

Examples of Metallic Materials

```
newmtl Mat001
  Ns 1
  Ks 1.0 0.766 0.336
  shader 5

newmtl Mat002
  Ns 10
  Ks 1.0 0.766 0.336
  shader 5

newmtl Mat003
  Ns 100
  Ks 1.0 0.766 0.336
  shader 5

newmtl Mat004
  Ns 1000
  Ks 1.0 0.766 0.336
  shader 5

newmtl Mat005
  Ns 10000
  Ks 1.0 0.766 0.336
  shader 5

newmtl Mat006
  Ns 1
  Ks 0.972 0.960 0.915
  shader 5

newmtl Mat007
  Ns 10
  Ks 0.972 0.960 0.915
  shader 5

newmtl Mat008
  Ns 100
  Ks 0.972 0.960 0.915
  shader 5

newmtl Mat009
  Ns 1000
  Ks 0.972 0.960 0.915
  shader 5

newmtl Mat010
  Ns 10000
  Ks 0.972 0.960 0.915
  shader 5

newmtl Mat011
  Ns 1
  Ks 0.955 0.637 0.538
  shader 5

newmtl Mat012
  Ns 10
  Ks 0.955 0.637 0.538
  shader 5

newmtl Mat013
  Ns 100
  Ks 0.955 0.637 0.538
  shader 5

newmtl Mat014
  Ns 1000
  Ks 0.955 0.637 0.538
  shader 5

newmtl Mat015
  Ns 10000
  Ks 0.955 0.637 0.538
  shader 5

newmtl Mat016
  Ns 1
  Ks 0.660 0.609 0.526
  shader 5

newmtl Mat017
  Ns 10
  Ks 0.660 0.609 0.526
  shader 5

newmtl Mat018
  Ns 100
  Ks 0.660 0.609 0.526
  shader 5

newmtl Mat019
  Ns 1000
  Ks 0.660 0.609 0.526
  shader 5

newmtl Mat020
  Ns 10000
  Ks 0.660 0.609 0.526
  shader 5

newmtl Mat021
  Ns 1
  Ks 0.560 0.570 0.580
  shader 5

newmtl Mat022
  Ns 10
  Ks 0.560 0.570 0.580
  shader 5

newmtl Mat023
  Ns 100
  Ks 0.560 0.570 0.580
  shader 5

newmtl Mat024
  Ns 1000
  Ks 0.560 0.570 0.580
  shader 5

newmtl Mat025
  Ns 10000
  Ks 0.560 0.570 0.580
  shader 5
```

Phong BRDF with Depth of Field


No Fresnel reflections


With Fresnel reflections

Phong BRDF with DoF and Fresnel Term

Split Direct and Indirect Illumination

- All paths that do not hit any light source will end up with zero contribution
- The solution is to connect each path vertex to a point on a light source – this technique is called next event estimation (NEE)
- We split the integral L_r of the reflected radiance as follows
  - Indirect illumination – sample hemisphere as before (RE in angular form)
  - Direct illumination – explicit sampling of light sources to ensure a non-zero contribution (RE in area form)

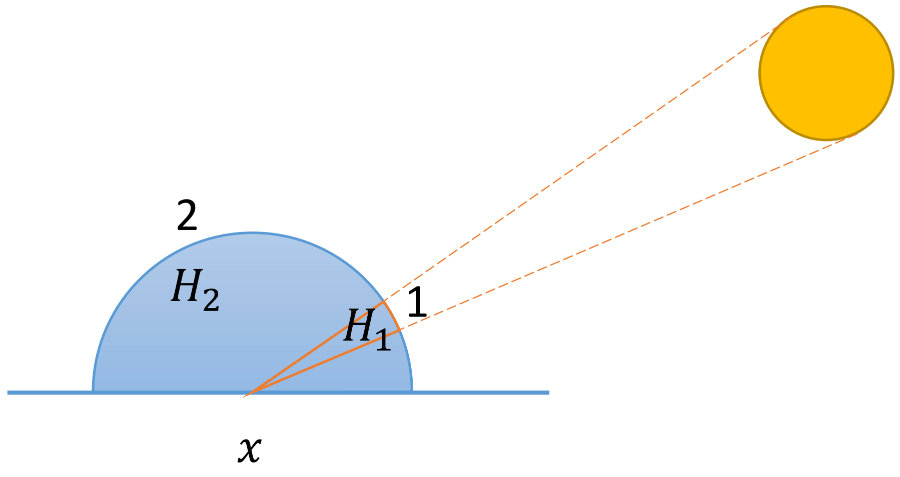
$$L_r(\boldsymbol{x}, \omega_0) = L_r^{ind}(\boldsymbol{x}, \omega_0) + L_r^{dir}(\boldsymbol{x}, \omega_0) =$$

$$= \int\limits_{H_2(\boldsymbol{x})} L_i(\boldsymbol{x}, \omega_i)\, f_r(\boldsymbol{x}, \omega_i, \omega_0) \cos\theta_i \,\mathrm{d}\omega_i + \int\limits_{H_1(\boldsymbol{x})} L_e(\boldsymbol{x}, \omega_i)\, f_r(\boldsymbol{x}, \omega_i, \omega_0) \cos\theta_i \,\mathrm{d}\omega_i$$

$$H_1 \cup H_2 = H$$

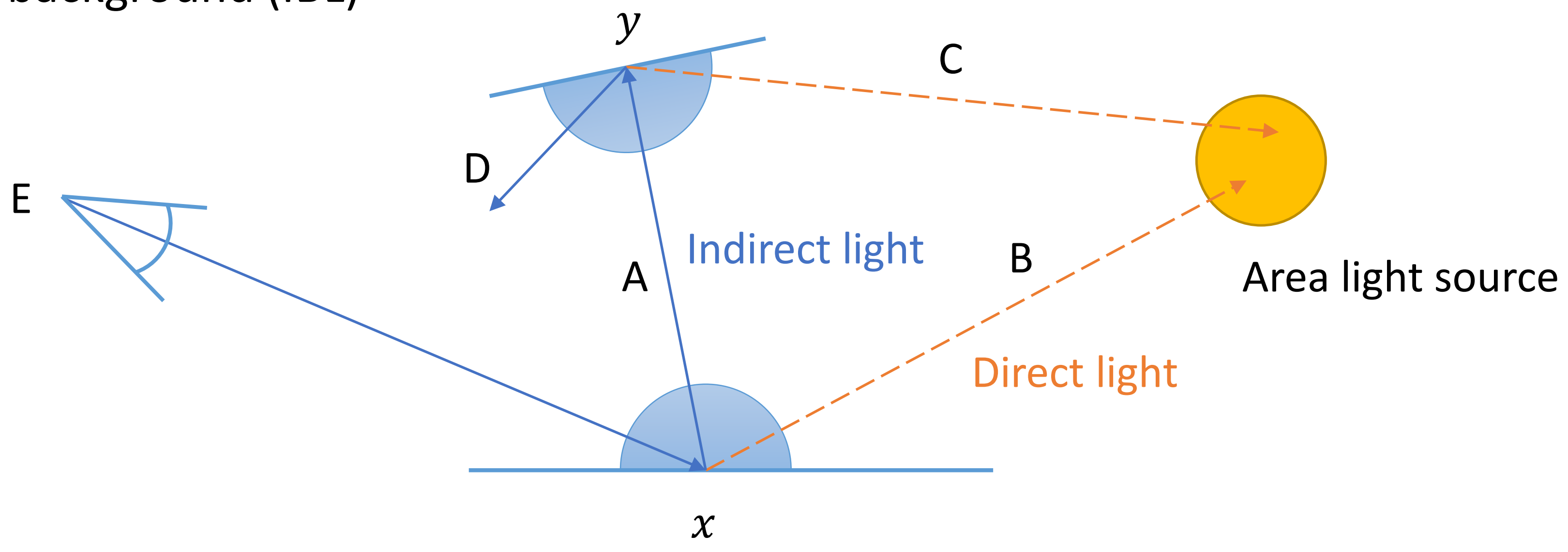
Next Event Estimation

- The whole process can be imagined as follows - sampled hemisphere is divided into two distinct areas
  - 1. An area that includes a projection of an area source
  - 2. A remaining area of the hemisphere



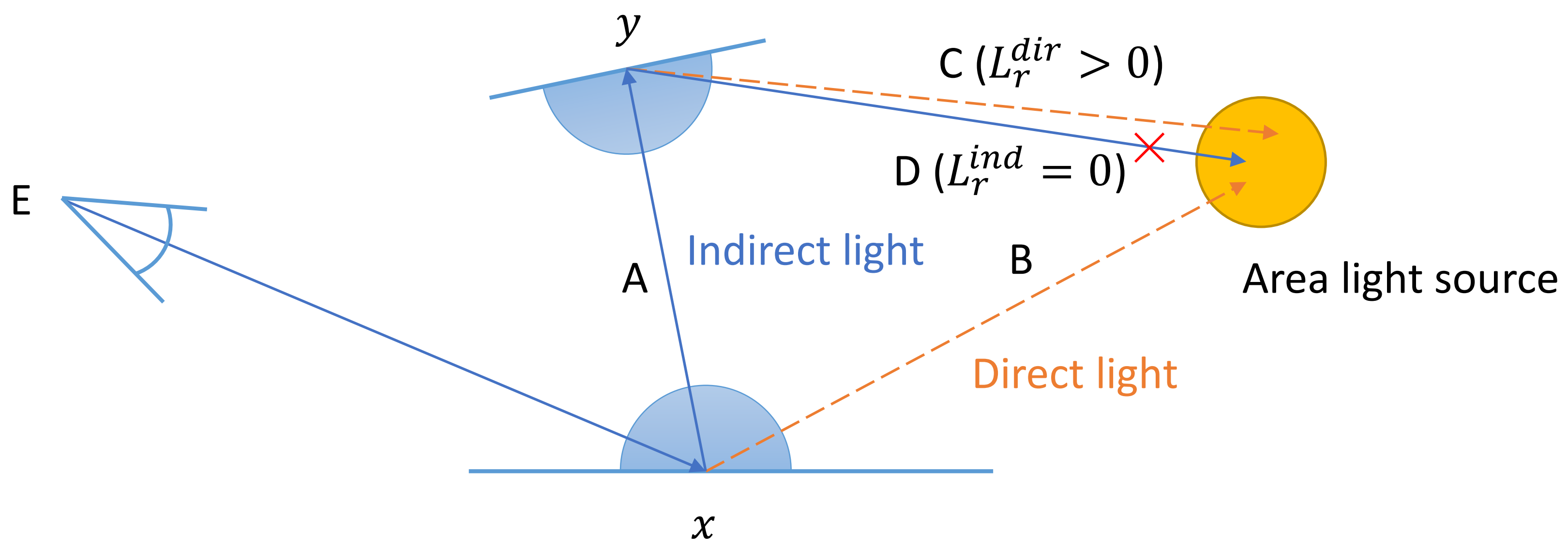
Next Event Estimation

- Ray A returns via point x the energy reflected by y (i.e. estimates indirect light at x)
- Ray B returns the radiance of the light source toward point x (i.e. estimates direct light at x)
- Ray C returns the radiance of the light source toward point y which will reach eye (pixel) via ray A
- Ray D hits the background (IBL)



Next Event Estimation

- When the ray D for indirect light sampling hits the light source, the path is terminated and **no energy** is returned through it. This will prevent accounting for direct illumination on point y twice

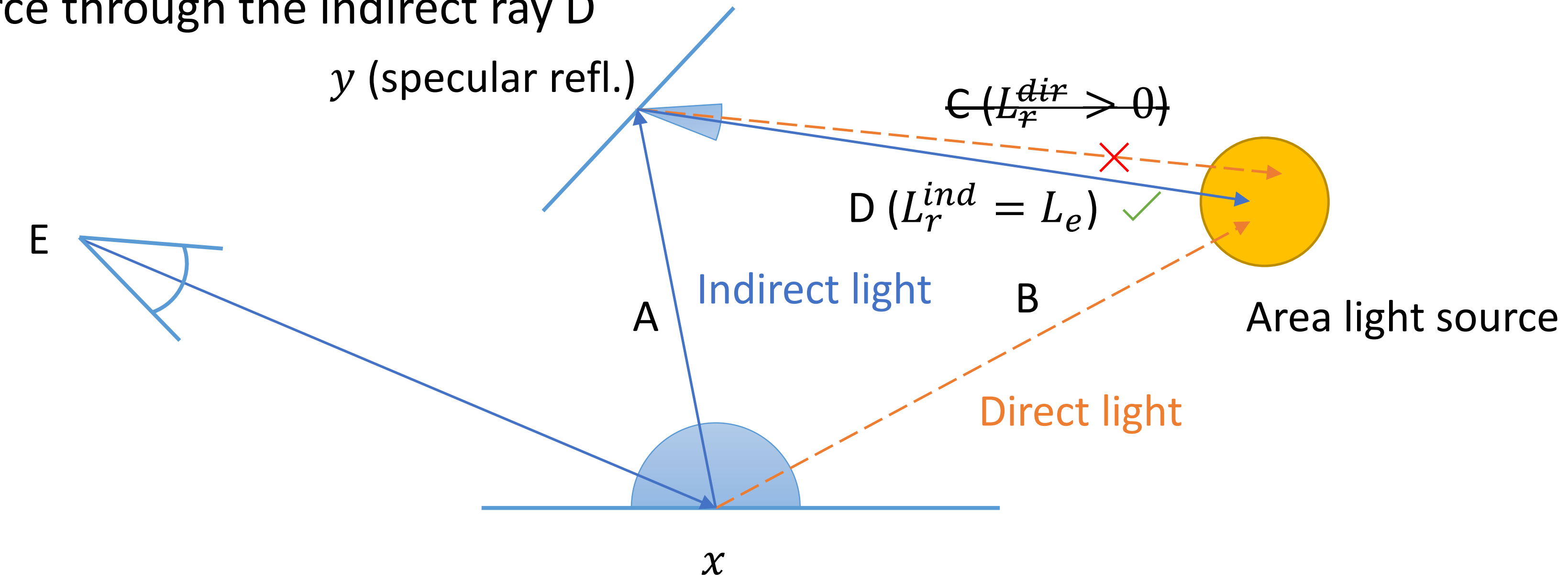


Next Event Estimation

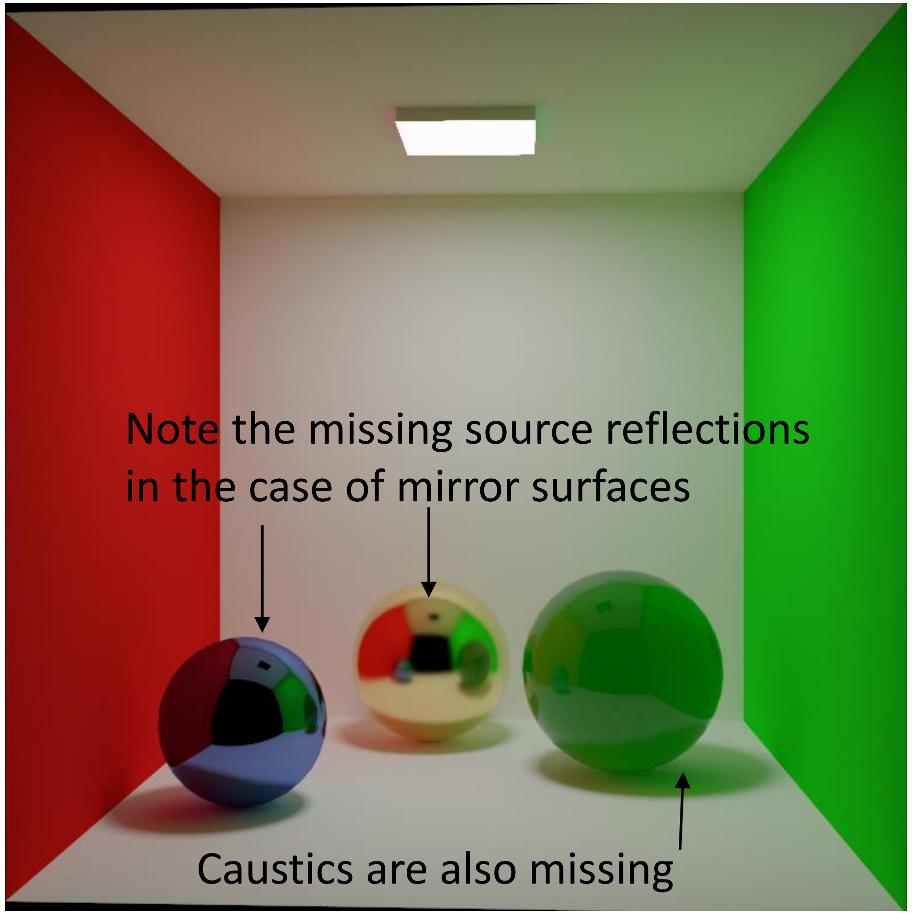
- Some vertices on light path require special attention
  - If the first vertex after the camera is emissive, its energy can’t be reflected to the camera (its energy must be returned directly)
  - For pure specular surfaces (mirrors), the BRDF always returns zero in the direction of a light source (i.e. no light ray)
- Since a light ray doesn’t make sense for specular vertices, we will include (possible) emission from a vertex directly following a specular vertex
- The same goes for the first vertex after the camera: if this is emissive, we will also include this
- This means we need to keep track of the type of the previous vertex during the path tracing

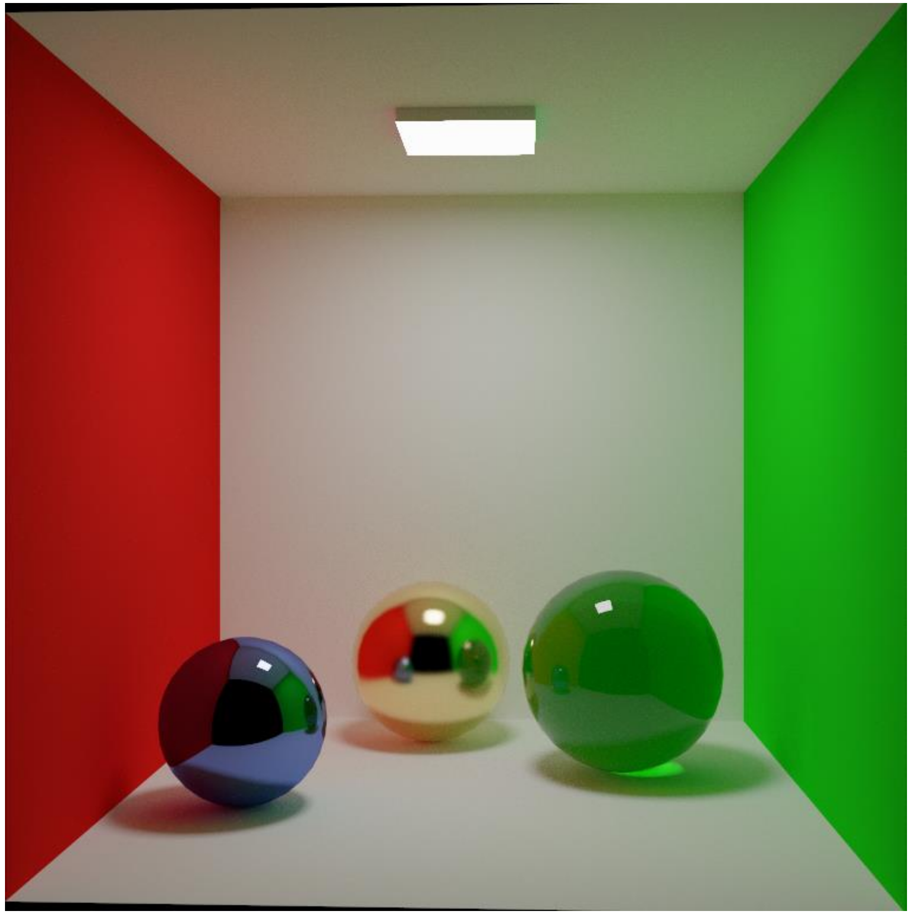
Next Event Estimation

- Once again, when the ray D for indirect light sampling hits the light source, the path is terminated and no energy is returned. It also holds that the mirror (specular) BRDF (almost) always returns zero in the direction of a light source. As a result, we will not see any specular reflection of a light source through the vertex y. Solution: iff the previous vertex is specular we pass the emission of the light source through the indirect ray D



Next Event Estimation





Remark About RE in Area Form

- 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)

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

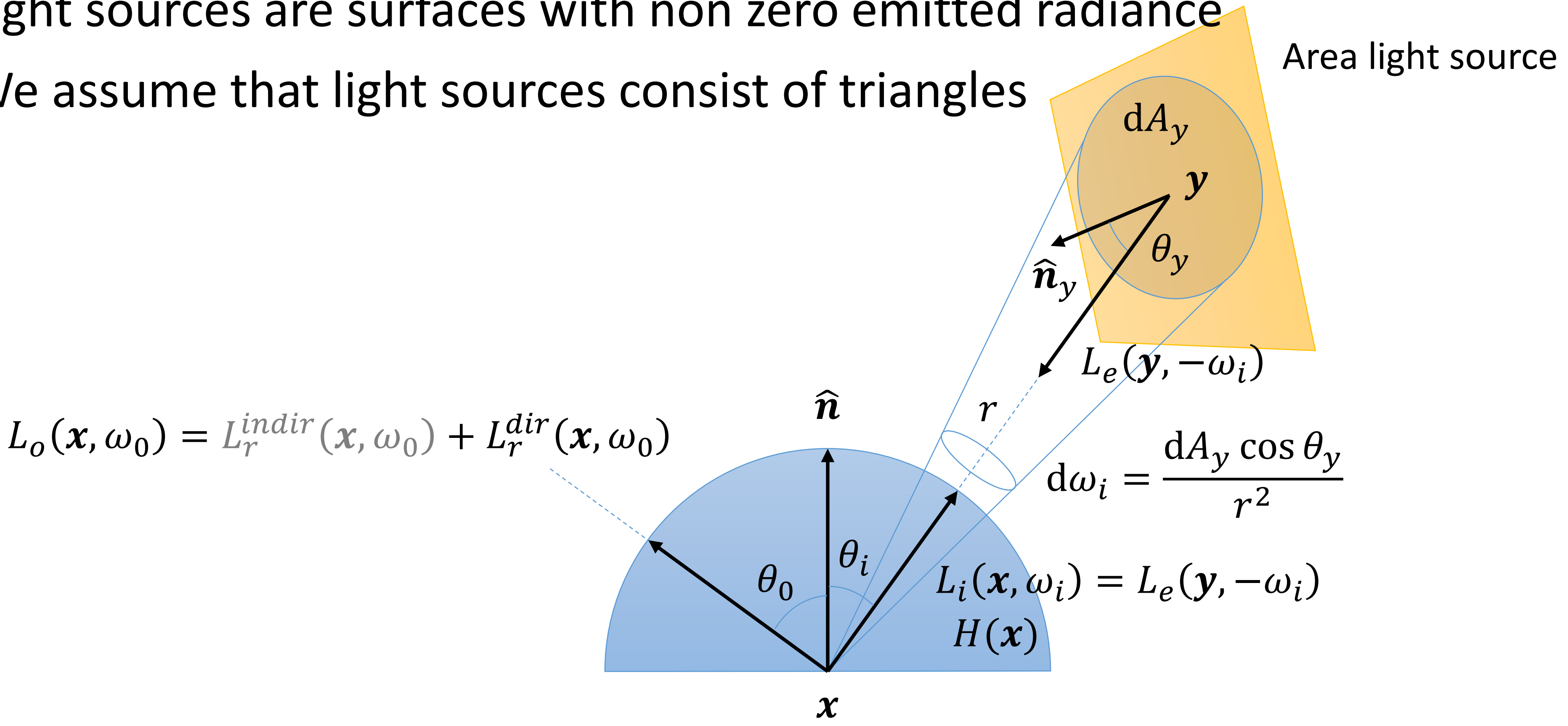
Geometry term $G(\boldsymbol{x}, \boldsymbol{y}) = \frac{\cos\theta_i \cos\theta_y}{\|\boldsymbol{x}-\boldsymbol{y}\|^2}$

Visibility term {0, 1}

Scene surface (e.g. a single triangle)

(Explicit) Sampling of Light Sources

- Light sources are surfaces with non zero emitted radiance
- We assume that light sources consist of triangles

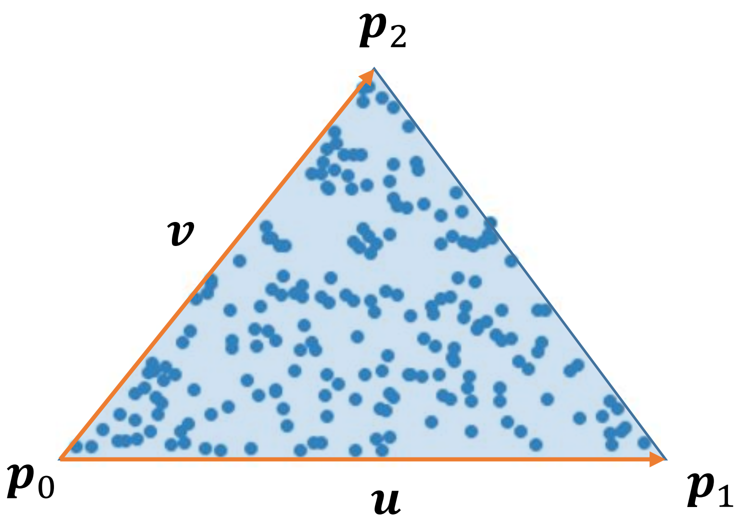


Triangle Sampling

- Uniform sampling of a triangle (given by $\boldsymbol{p}_0, \boldsymbol{p}_1, \boldsymbol{p}_2$) with pdf equal to $1/Area$

$$\boldsymbol{y} = \boldsymbol{p}_0\left(1 - \sqrt{\xi_1}\right) + \boldsymbol{p}_1\sqrt{\xi_1}(1 - \xi_2) + \boldsymbol{p}_2\sqrt{\xi_1}\xi_2$$

$$pdf(\boldsymbol{y}) = 1/Area$$



where $Area = \frac{1}{2}\|\boldsymbol{u} \times \boldsymbol{v}\|$ and $\boldsymbol{u} = \boldsymbol{p}_1 - \boldsymbol{p}_0, \boldsymbol{v} = \boldsymbol{p}_2 - \boldsymbol{p}_0$

or $Area = \frac{1}{2}\sqrt{|\det(A^T A)|}$ and $A = \begin{bmatrix} \vdots & \vdots \\ \boldsymbol{u} & \boldsymbol{v} \\ \vdots & \vdots \end{bmatrix}$

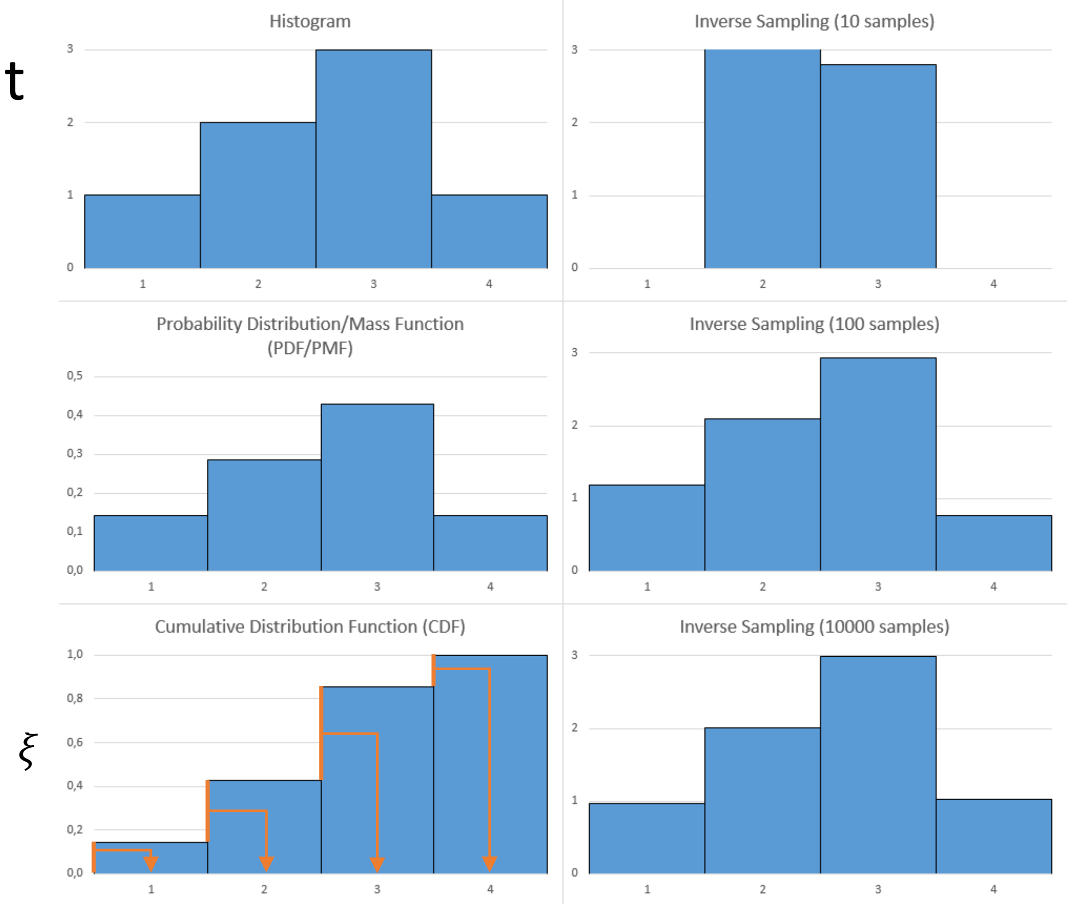
Sampling Many Light Triangles

- Problem: If the light source consists of many triangles, NEE will become slow (as we need to sum contributions from all light emitting triangles)
- Solution:
  - Trivial case – triangles are equally sized – pick randomly any triangle and weight the sample by inverse probability of choosing the particular triangle
  - Real case – triangles have different areas – almost the same technique like before but we need inverse sampling as we need random samples generated from custom probability distribution

Sampling From Arbitrary PDF/PMF

- Inverse transform sampling is a basic method for generating sample numbers at random from any probability distribution given its cumulative distribution function

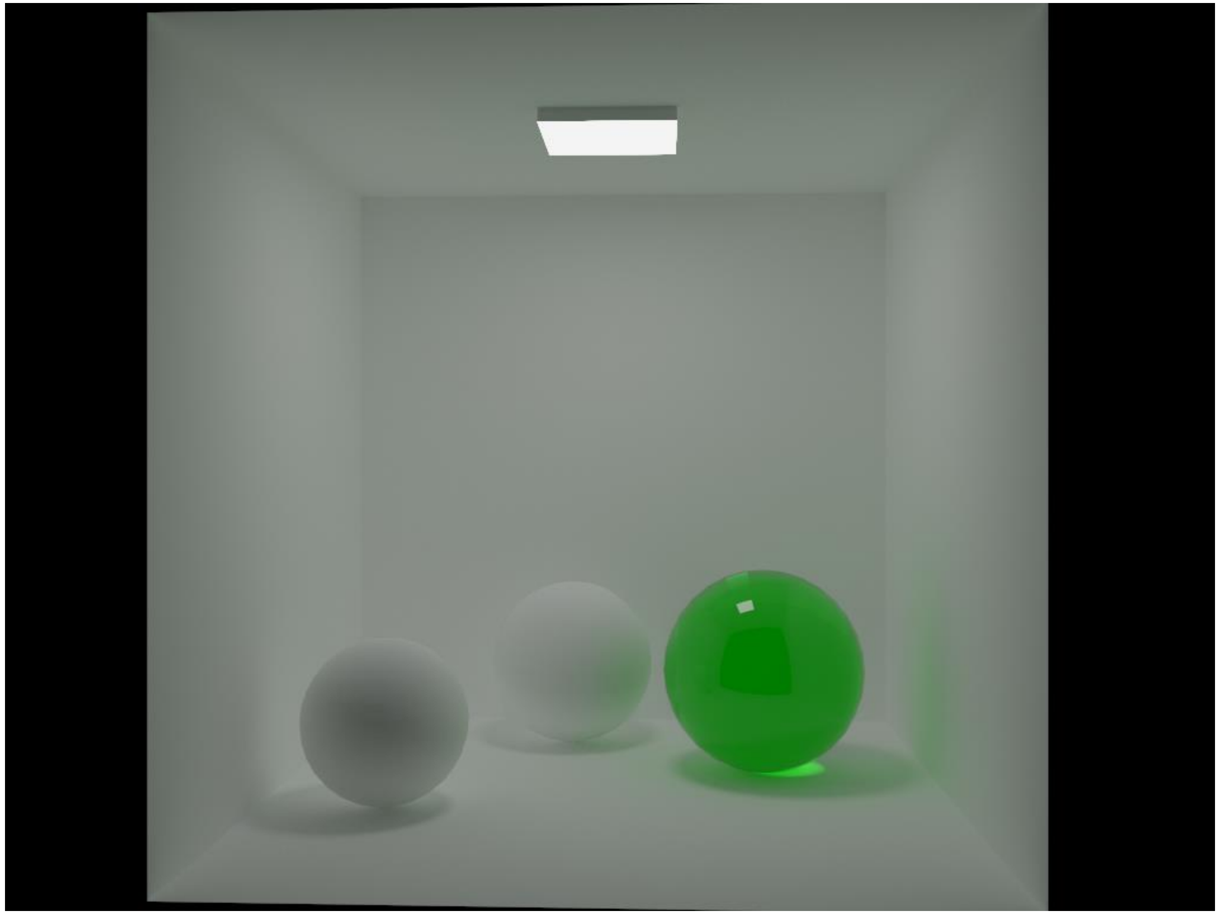
```
std::vector<float> cdf = { 0.143f, 0.429f, 0.857f, 1.0f };
std::vector<int> samples = { 0, 0, 0, 0 };
int N = 100;

for ( int i = 0; i < N; ++i ) {
  float ksi = random->next_float( 0.0f, 1.0f );
  auto lower = std::lower_bound( cdf.begin(), cdf.end(), ksi );
  int index = lower - cdf.begin();

  samples[index]++;
}

for ( auto s : samples ) {
  printf( "%0.3f\n", s / float( N ) );
}
```



Note About Density and Distribution

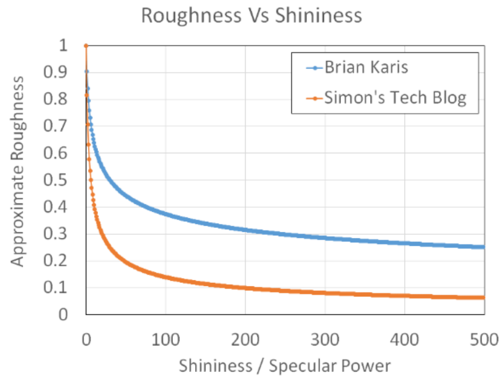
- Probability density function (PDF) contains information about probability of a continuous random variable but it is not a probability since can have any non-negative value, even larger than one. It has only to satisfty $f(x) \geq 0$ and $\int_{-\infty}^{\infty} f(x)\mathrm{d}x = 1$. Probability of a random variable X is then
$$P(a \leq X \leq b) = \int_a^b f(x)\mathrm{d}x = F_X(b) - F_X(a)$$
- Probability mass function (PMF) gives the probability that a discrete random variable is equal to some value
- Cumulative distribution function (CDF) gives the probability that a discrete or continuous random variable will take a value less than or equal to some value and its value is a number between zero and one
$$F_X(x) = P(X \leq x) = \int_{-\infty}^{x} f_X(t)\mathrm{d}t$$

Example (RR+NEE)



Dielectrics vs Metals

Workflow	MTL	Param. Name	Dielectrics	Metals
Specular-Glossines	Ks[1]	Specular reflectivity	cca 0.05 grayscale	<0.4, 1> tint
	Kd	Diffuse reflectivity	<0, 1>	0
	Ns[2]	Specular exponent	<0, cca 10000>	
	Ni	Index of refraction	<1, cca 4>	<0.15, cca 3>
	Tf	transmission filter	<0, inf>	-
	Ke	emission	<0, inf>	



Notes:

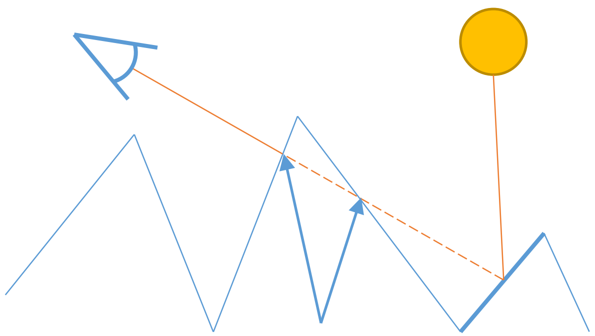
1. Specular reflectivity computed from IOR: dielectrics $R = \frac{(\eta_2-\eta_1)^2}{(\eta_2+\eta_1)^2}$; metallic $R = \frac{(\eta_2-\eta_1)^2+k^2}{(\eta_2+\eta_1)^2+k^2}$ where η_1 is ior of air, η_2 is ior of metal, and k is damping (extinction) constant . Also note that R is a scalar and Ks is a color. Consult https://refractiveindex.info for indices of various materials
2. Conversion between shininess (or specular power exponent) and roughness is not defined exactly

Torrance-Sparrow BRDF

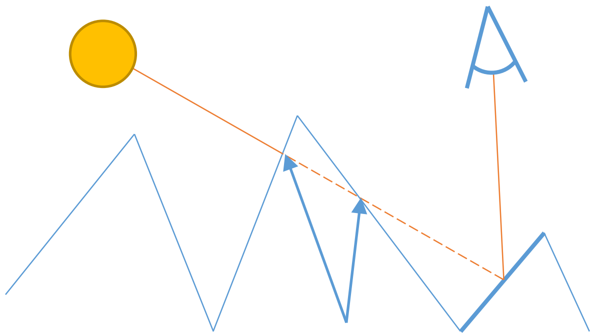
- One of the most complete physical reflection models for isotropic materials (precursor to other models)
- Validated by a ray-tracing-like simulation
- Considers polarized light and is used for rough surfaces
- The roughness is modelled using microscopic concavities
- Specularly reflecting V-cavities of equal length are called microfacets
- Their orientation is random and their distribution is controlled by parameters, so it is possible to simulate different degrees of roughness.

Microfacet Theory

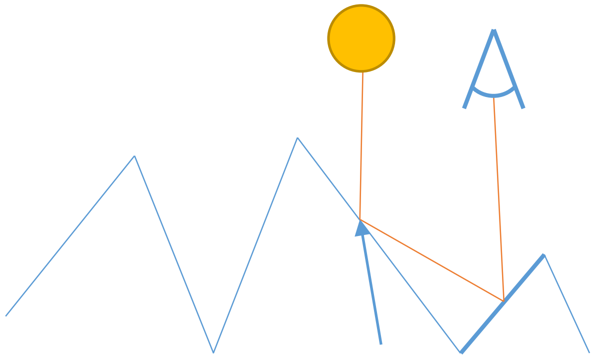
- Main idea: rough surfaces can be modeled as a collection of small microfacets (tiny planar areas of the surface used in reflection approximation)
- Surface is described by a distribution of microfacet normals
- Perfect mirror reflection is most commonly used for the microfacet BRDF
- Local lighting efects at the microfacet level are masking, shadowing, and interreflection



masking



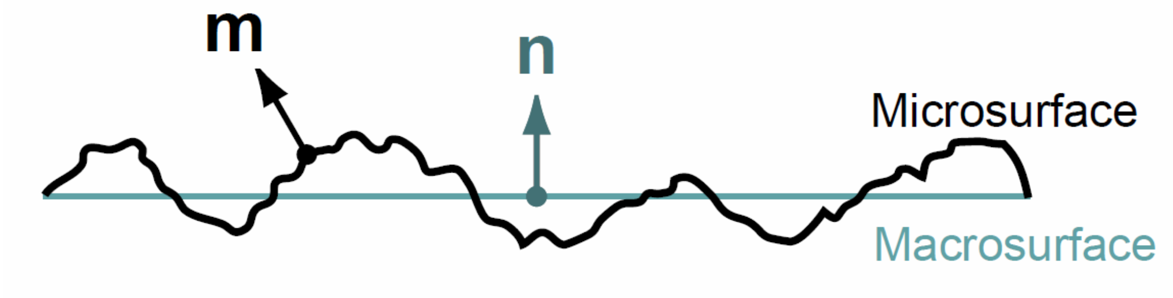
shadowing



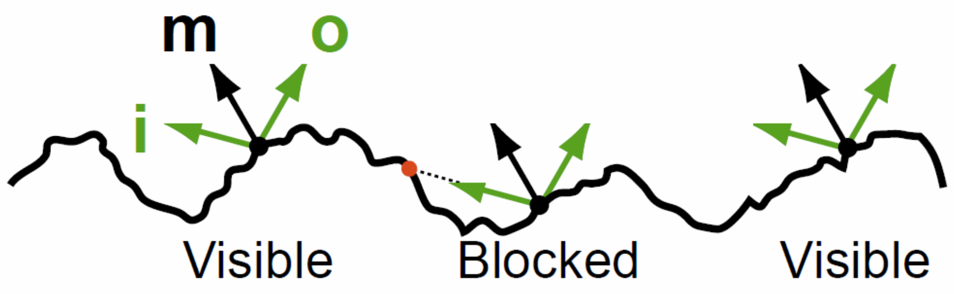
interreflection

Microfacet Theory

- Micro vs. macro surface



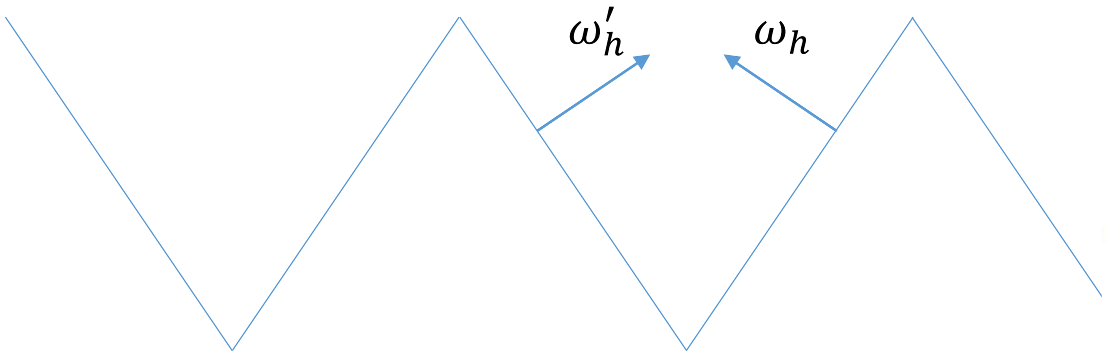
- Shadowing-masking geometry



Source: WALTER, Bruce, et al. Microfacet models for refraction through rough surfaces. In: *Proceedings of the 18th Eurographics conference on Rendering Techniques*. 2007. p. 195-206.

Microsurface Models

- Two common microsurface models
  - **V-cavity model** intruduced by Torrance et al.
  - Assumes a continuous distribution of separate microsurfaces rather than just one microsurface
  - Each microsurface is composed of two normals $\omega_m = (x_m, y_m, z_m)$ and $\omega_m' = (-x_m, -y_m, z_m)$



Masking function

$$G_1(\omega, \omega_m) = \chi^+\left(\frac{\omega \cdot \omega_m}{\omega \cdot \omega_g}\right) \min\left(1, 2\frac{|\omega_m \cdot \omega_g||\omega \cdot \omega_g|}{|\omega, \omega_m|}\right)$$

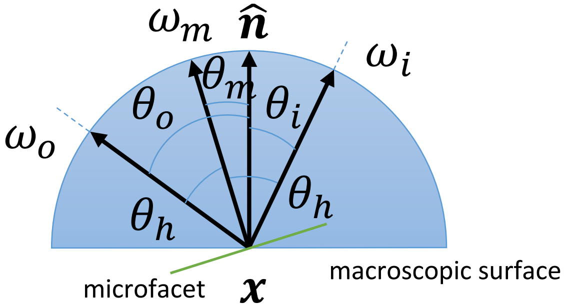
Masking-shadowing function

$$G_2(\omega_i, \omega_o, \omega_m) = \begin{cases} \min\left(G_1(\omega_i, \omega_m), G_1(\omega_o, \omega_m)\right), & \text{if } \omega_o \cdot \omega_g > 0, \\ \max\left(G_1(\omega_i, \omega_m) + G_1(\omega_o, \omega_m) - 1, 0\right), & \text{otherwise.} \end{cases}$$

Microsurface Models

- **Smith model** is the most accurate model for compactly describing geometrical-optics reflectance and transmission from explicit random height fields (such as an ocean surface), and is better than the V-cavity model for approximating reflectance of measured materials. However, specific masking functions G_1 must be derived for each new distribution of normals. Analytic solutions are available for Beckmann and GGX. Heitz showed that the same formulas can be used with the anisotropic extensions of D. Different forms of the Smith masking-shadowing function are available, the most simple is the non-correlated form $G_2(\omega_o, \omega_i) = G_1(\omega_o) G_1(\omega_i)$

Torrance-Sparrow BRDF



- BRDF takes form

Normal distribution function $\langle 0, \infty \rangle$ determines what percentage* of microfacets are oriented to reflect the incoming light in the viewer direction

* this is not exact, see slides below

Fresnel reflectance (RGB), depends on wavelength

Geometrical attenuation factor $\langle 0,1 \rangle$ reduces the output based on the amount of shadowing or masking

$$f_r^{TS}(\omega_i, \omega_o) = k_d + k_s^{TS} = \frac{\rho_d}{\pi} + \frac{D(\omega_m, \alpha)\, F(\theta_h)\, G(\omega_i, \omega_o)}{\mathbf{4} \cos\theta_o \cos\theta_i}$$

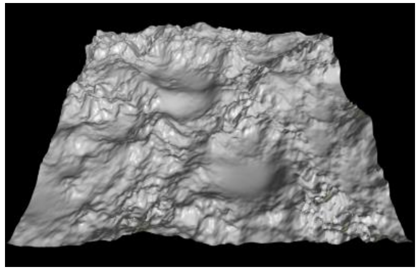
How much of the **macroscopic** surface is visible to the viewer
$\cos\theta_o = \omega_0 \cdot \hat{\boldsymbol{n}}$

How much of the **macroscopic** surface is visible to the light
$\cos\theta_i = \omega_i \cdot \hat{\boldsymbol{n}}$

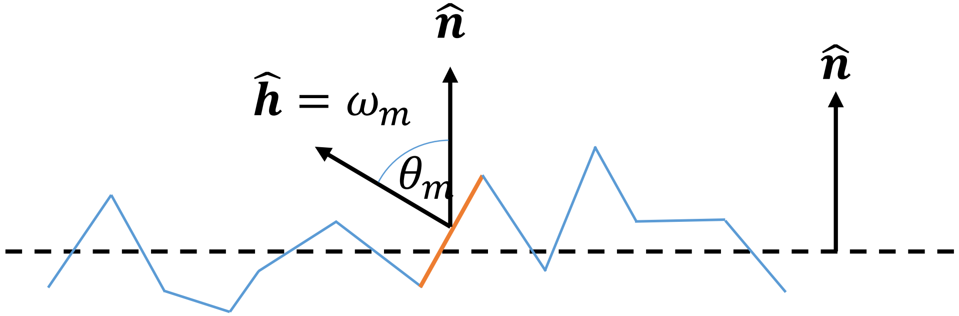
- where D is the microfacet distribution function, F is the Fresnel function, G is the geometric attenuation function responsible for masking and self-shadowing, and the direction $\hat{\boldsymbol{h}} = \hat{\boldsymbol{m}} = \omega_m = (\omega_i + \omega_o)/\|\omega_i + \omega_o\| = (\omega_i + \omega_o)/2$, a.k.a half-(way) vector, represents the microfacet normal
- Can be used for both conductors and dielectrics

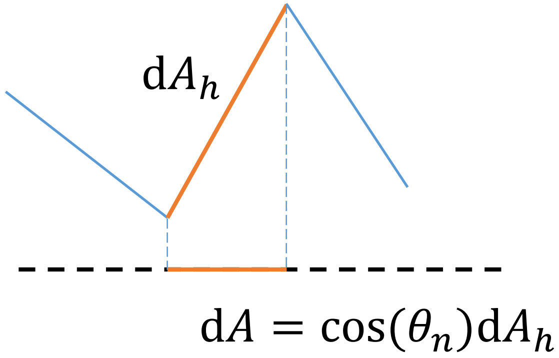
Normal Distribution Function

- Based on microfacet surface models
- Distribution of facets normals $\hat{\boldsymbol{h}}$ is described in a statistical manner (otherwise it would be very expensive to evaluate all microfacets on object surface)
- Distribution is often derived from some heightfield



- NDF for perfectly smooth surface: $D(\omega_m) = \delta(\theta_m)$
- NDF must be normalized to be physically plausible: $\int_H D(\omega_m)\cos(\theta_m)\mathrm{d}\omega_m = 1$





Normal Distribution Function

Source: http://www.reedbeta.com/blog/hows-the-ndf-really-defined/

- According to Walter at al., NDF obeys the equation

$$\mathrm{d}A_m = D(\omega_m)\mathrm{d}\omega_m A$$

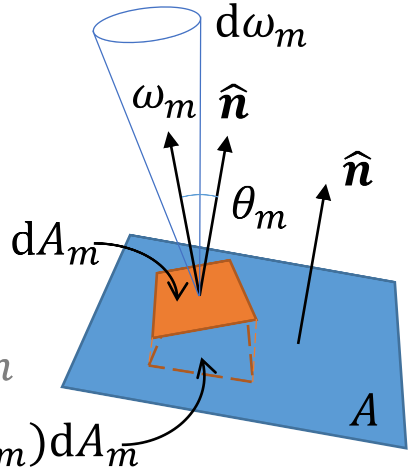
where A is a patch of macrosurface small enough to be considered flat, but much larger than an individual microfacet, and $\mathrm{d}A_m$ is the total area of all the microfacets within A that have normals within $\mathrm{d}\omega_m$

From this point of view NDF is not a PDF at all but it is the density of micro-area over the joint domain of macro-area and solid angle

$$D(\omega_m) = \frac{\mathrm{d}A_m}{\mathrm{d}\omega_m A}\left[\frac{\cancel{m^2}}{sr\cdot\cancel{m^2}}\right]$$

- The normalization condition can be derived from the above equation

$$1 = \underbrace{\frac{1}{A}\int \underbrace{\cos(\theta_m)\mathrm{d}A_m}_{\mathrm{d}A}}_{1} = \int_H D(\omega_m)\cos(\theta_m)\mathrm{d}\omega_m = \frac{1}{A}\int \frac{\cos(\theta_m)}{\cancel{\mathrm{d}\omega_m}}\cancel{\mathrm{d}\omega_m}\mathrm{d}A_m$$



Normal Distribution Function

- Beckmann NDF

$$D_{Beckmann}(\theta_m, m) = \frac{1}{\pi\, \alpha^2 (\cos\theta_m)^4} \exp\left(\frac{(\cos\theta_m)^2 - 1}{\alpha^2 (\cos\theta_m)^2}\right) =$$

$$= \frac{1}{\pi\, \alpha^2 (\cos\theta_m)^4} \exp\left(-\left(\frac{\tan\theta_m}{\alpha}\right)^2\right)$$

- α … roughness (or RMS slope of the surface microfacets)
- $\cos\theta_m = \hat{\boldsymbol{n}} \cdot \hat{\boldsymbol{h}}$ and $\hat{\boldsymbol{h}} = \frac{\omega_i + \omega_o}{\|\omega_i + \omega_o\|}$ and $\omega_i = \mathrm{reflect}(\omega_o, \hat{\boldsymbol{h}})$

Normal Distribution Function

- (Isotropic) Trowbridge-Reitz (a.k.a GGX) NDF

$$D_{TR-GGX}(\theta_m, \alpha) = \frac{\alpha^2}{\pi\left((\alpha^2 - 1)(\cos\theta_m)^2 + 1\right)^2} =_{\alpha=1} \frac{1}{\pi}$$

Note that for rough surfaces the GGX NDF equals to the Lambertian surface

where $\alpha = \mathrm{roughness}^2$ (aka Disney's reparametrization)

- Here presented NDF is a simplified version for the upper hemisphere, i.e. $\theta \in \langle 0, \pi/2\rangle$

Accounts for the projected area of microfacets onto the macro-surface

- Also holds that $\int_H D_{TR-GGX}(\theta_m, \alpha) \cos\theta_m \, \mathrm{d}\omega_m = 1$

Normal Distribution Function

- Equivalent formulation of (isotropic) Trowbridge-Reitz (a.k.a GGX) NDF

$$D_{TR-GGX}(\theta_m, \alpha) = \frac{\alpha^2}{\pi(\cos\theta_m)^4(\alpha^2 + (\tan\theta_m)^2)^2}$$

Equivalent formulation from Walter et al. 2007

- Equations for (anisotropic) Trowbridge-Reitz (a.k.a GGX) NDF

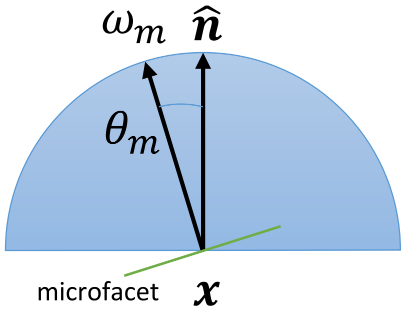
$$D_{TR-GGX}(\varphi_m, \theta_m, \alpha_x, \alpha_y)$$

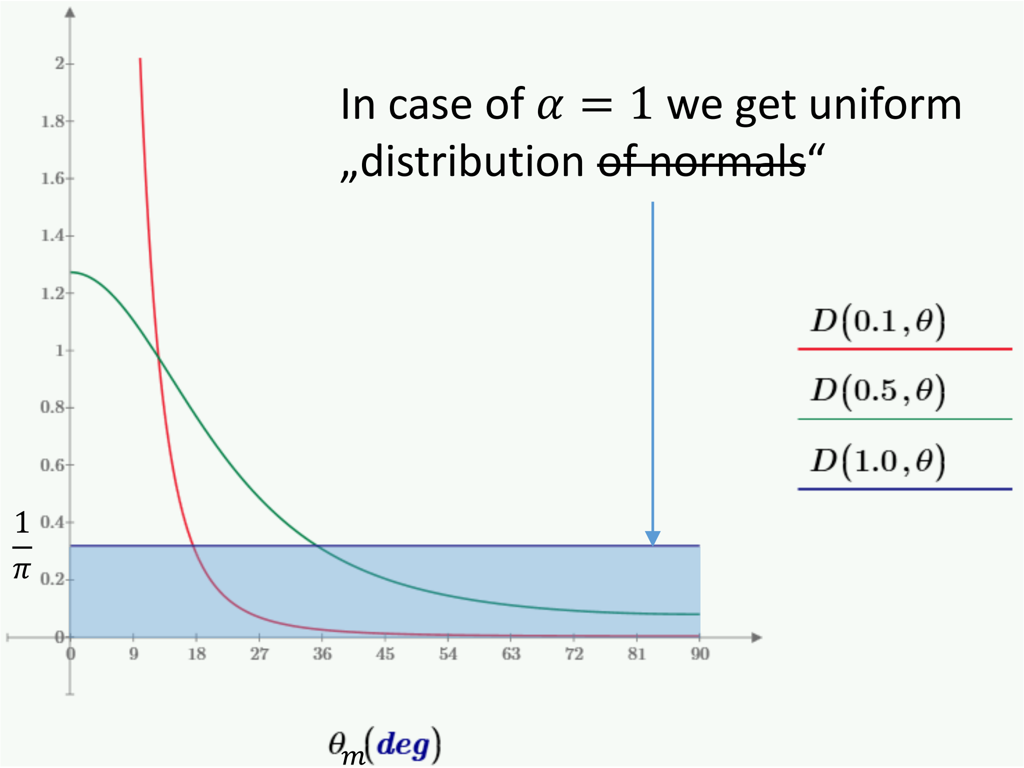
$$= \frac{1}{\pi\alpha_x\alpha_y(\cos\theta_m)^4\left(1 + (\tan\theta_m)^2\left(\frac{(\cos\varphi_m)^2}{\alpha_x^2} + \frac{(\sin\varphi_m)^2}{\alpha_y^2}\right)\right)^2} =$$

$$= \frac{1}{\pi\alpha_x\alpha_y\left(\frac{m_x^2}{\alpha_x^2} + \frac{m_y^2}{\alpha_y^2} + m_z^2\right)^2}$$

Normal Distribution Function

- Trowbridge-Reitz (isotropic GGX)





Normal Distribution Function

- Lets check that $\int_H D_{TR-GGX}(\theta_m, \alpha) \cos\theta_m \, \mathrm{d}\omega_m = 1$

```
from sympy import *
from scipy.integrate import nquad
def D_GGX(θm, φm, α=0.45):
  d_ggx = α**2 / (π *((α**2-1)*cos(θm)**2+1)**2)
  return d_ggx * cos(θm) * sin(θm) # see the next slide
I = nquad(D_GGX, [[0, π/2], [0, 2*π]]) # integral over the hemisphere
print(I)
Out: 1.0
```

Normal Distribution Function

- Lets check that $\int_H D_{TR-GGX}(\theta_m, \alpha = 1) \cos\theta_m \, \mathrm{d}\omega_m = 1$

$$\int_H D_{TR-GGX}(\theta_m, 1) \cos\theta_m \, \mathrm{d}\omega_m = \int_0^{2\pi} \int_0^{\frac{\pi}{2}} \frac{1}{\pi} \cos\theta_m \sin\theta_m \, \mathrm{d}\theta_m \mathrm{d}\varphi_m =$$

$$= -\frac{1}{2\pi} \int_0^{2\pi} [\cos^2\theta_m]_0^{\frac{\pi}{2}} \mathrm{d}\varphi_m = -\frac{1}{2\pi} \int_0^{2\pi} [0 - 1] \mathrm{d}\varphi_m =$$

$$= \frac{1}{2\pi} [2\pi - 0] = 1$$

Sampling Isotropic D_{TR-GGX}

Chain rule of probability:
$p(a,b) = p(a|b)p(b) = p(b|a)p(a)$

- The pdf of D_{TR-GGX}

$$p(\omega_m) = D_{TR-GGX}(\theta_m, \alpha)\cos\theta_m$$

from which we get the joint probability respecting the spherical coordinates (recall again that $\mathrm{d}\omega_m = \sin\theta_\mathrm{m}\mathrm{d}\theta_m\mathrm{d}\varphi_m$)

$$p(\theta_m, \varphi_m) = D_{TR-GGX}(\theta_m, \alpha)\cos\theta_m \sin\theta_m$$

Also recall that θ_m represens the polar angle and φ_m is the azimuthal angle of sampled microfacet normal ω_m. Also note that φ_m is not included in the above pdf making this BRDF isotropic and the marginal probability of θ_m equals to

$$p(\theta_m) = \int_0^{2\pi} p(\theta_m, \varphi_m)\mathrm{d}\varphi_m = \frac{2\alpha^2\cos\theta_m\sin\theta_m}{\left((\alpha^2-1)(\cos\theta_m)^2+1\right)^2}$$

Marginal probability is the probability of one event in the presence of all (or a subset of) outcomes of the other random variable

Sampling Isotropic D_{TR-GGX}

- For inverse transform sampling, we need to calculate the cdf as follows

$$cdf(\theta_m) = \int_0^{\theta_m} p(\theta)\mathrm{d}\theta =$$

$$= \int_0^{\theta_m} \frac{2\alpha^2 \cos\theta \sin\theta}{\left((\alpha^2-1)(\cos\theta)^2+1\right)^2}\mathrm{d}\theta \left| \begin{array}{c} x = \cos\theta \\ \mathrm{d}x = -\sin\theta\,\mathrm{d}\theta \rightarrow \mathrm{d}\theta = -\mathrm{d}x/\sin\theta \end{array} \right. =$$

$$= \int_{\cos(0)}^{\cos(\theta_m)} \frac{-2\alpha^2 x \cancel{\sin\theta\, x}}{\left((\alpha^2-1)x^2+1\right)^2 \cancel{\sin\theta}}\mathrm{d}x = \left[\frac{\alpha^2}{(\alpha^2-1)^2x^2+\alpha^2-1}\right]_1^{\cos(\theta_m)} =$$

$$= \frac{\alpha^2}{(\alpha^2-1)^2(\cos\theta)^2+\alpha^2-1} - \frac{1}{\alpha^2-1} = \frac{1}{\alpha^2\left((\sin\theta)^{-2}-1\right)+1}$$

Sampling Isotropic D_{TR-GGX}

- Now we need to solve the following equation for θ_m having an uniform random number ξ_1

$$cdf(\theta_m) = \xi_1$$

$$\frac{1}{\alpha^2\left((\sin\theta_m)^{-2} - 1\right) + 1} = \xi_1$$

$$(\sin\theta_m)^{-2} = \frac{1-\xi_1}{\alpha^2\xi_1} + 1;\text{ note that } (\sin\theta)^{-2} = \frac{1}{1-(\cos\theta)^2}$$

$$(\cos\theta_m)^2 = \frac{1-\xi_1}{\xi_1(\alpha^2-1)+1} \Rightarrow \theta_m = \cos^{-1}\left(\sqrt{\frac{1-\xi_1}{\xi_1(\alpha^2-1)+1}}\right)$$

Sampling Isotropic D_{TR-GGX}

- Now we need to solve the following equation for θ_m having an uniform random number ξ_1

$$cdf(\theta_m) = \xi_1$$

what gives us

$$\theta_m = \cos^{-1}\left(\sqrt{\frac{1-\xi_1}{\xi_1(\alpha^2-1)+1}}\right) = \tan^{-1}\left(\alpha\sqrt{\frac{\xi_1}{1-\xi_1}}\right)$$

Alt., more effective form for later conversion to Cartesian coordinates of ω_m

$$\cos(\theta_m) = \frac{1}{\sqrt{1+\xi_1(\alpha^2-1)}}$$

$$\sin(\theta_m) = \sqrt{1-(\cos(\theta_m))^2}$$

$$\varphi_m = 2\pi\xi_2$$

Remark on Probability of Sampling D_{TR-GGX}

- Now we know how to sample ω_m proportional to D_{TR-GGX} NDF
- However, the way we calculate the PDF given an incident direction is different from the above ones respecting the half-vector, whether it's solid angle or spherical coordinate. What we have so far is the PDF for half-vector, a following transformation is necessary to get the correct PDF for θ_i

$$p(\theta_i) = p(\omega_m)\frac{\mathrm{d}\omega_m}{\mathrm{d}\omega_i} = p(\omega_m)\frac{1}{4\left(\omega_{\{o,\,i\}}\cdot\omega_m\right)} = \frac{D_{TR-GGX}(\theta_m,\alpha)\cos\theta_m}{4\cos\theta_h}$$

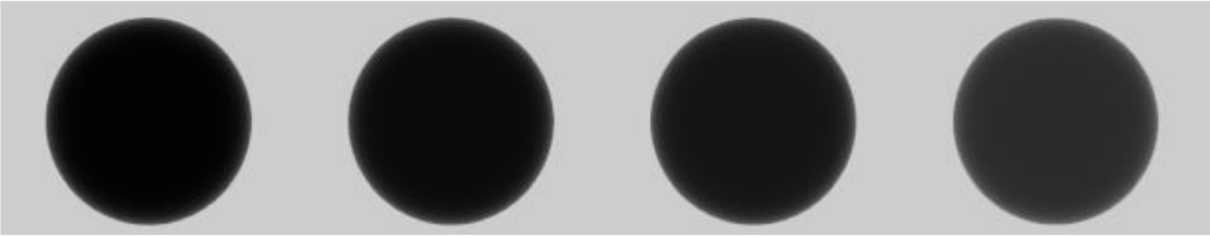
The solid angle transformation of the reflection operation is the Jacobian

For more details, see Section 1.3 in Linear Efficient Antialiased Displacement and Reflectance Mapping: Supplemental Material

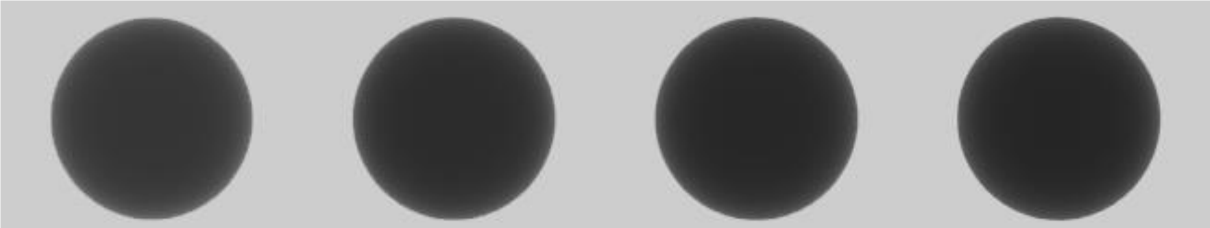
Fresnel Factor

- The Fresnel factor $F \in \langle 0, 1 \rangle$ gives the fraction of light that is reflected from the entire surface. Its computation is a linear combination of the coefficient for perpendicular light polarization and the coefficient for parallel light polarization
- It is quite usual to use the Schlick's approximation as we did before

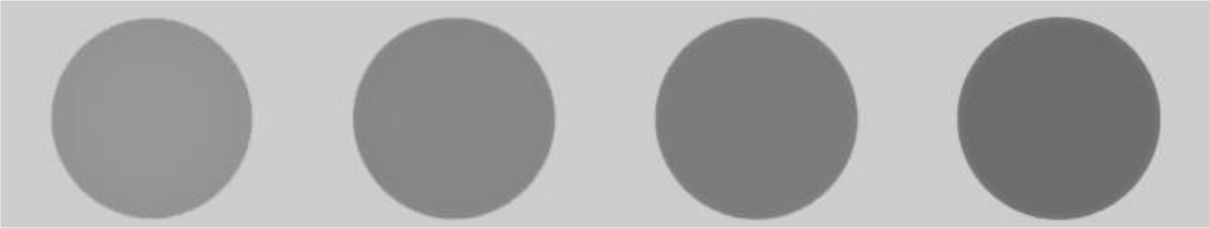
Fresnel Factor



Dielectric Fresnel: From left to right the index of refraction is 1.2, 1.5, 1.8, 2.4

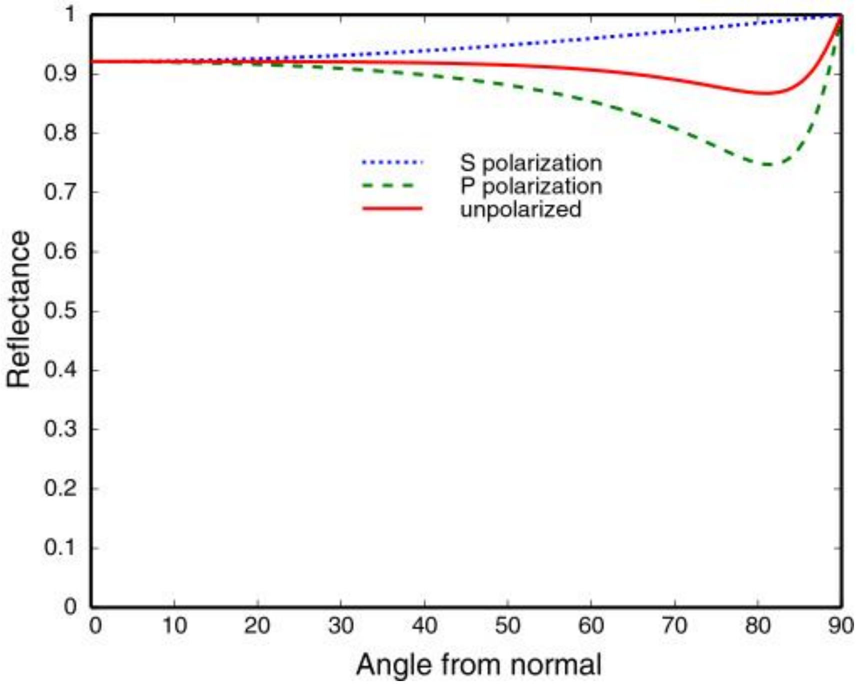


Conductor Fresnel with Absorption k = 2: From left to right the index of refraction is 1.2, 1.5, 1.8, 2.4

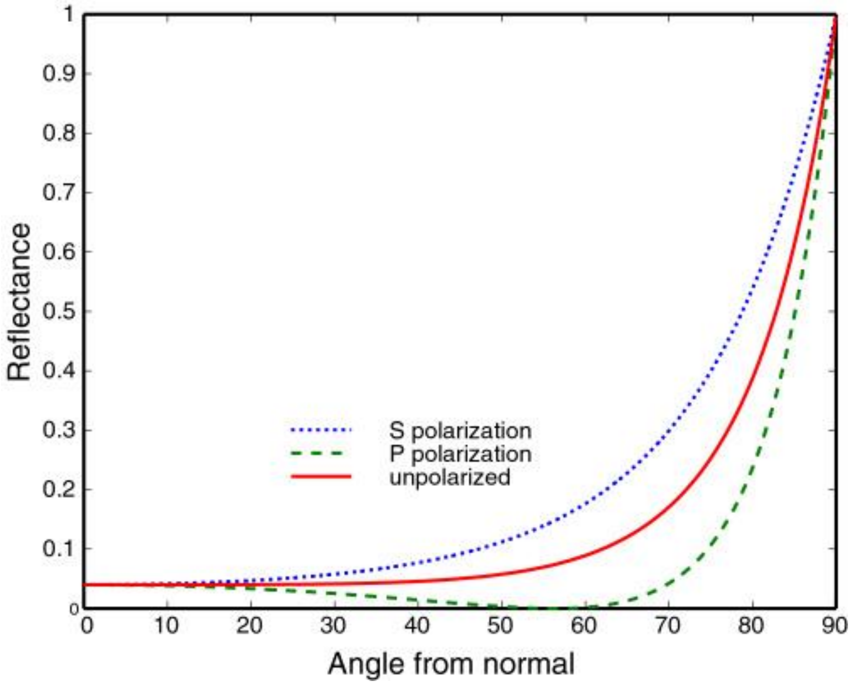


Conductor Fresnel with Absorption k = 4: From left to right the index of refraction is 1.2, 1.5, 1.8, 2.4

Fresnel Reflectance



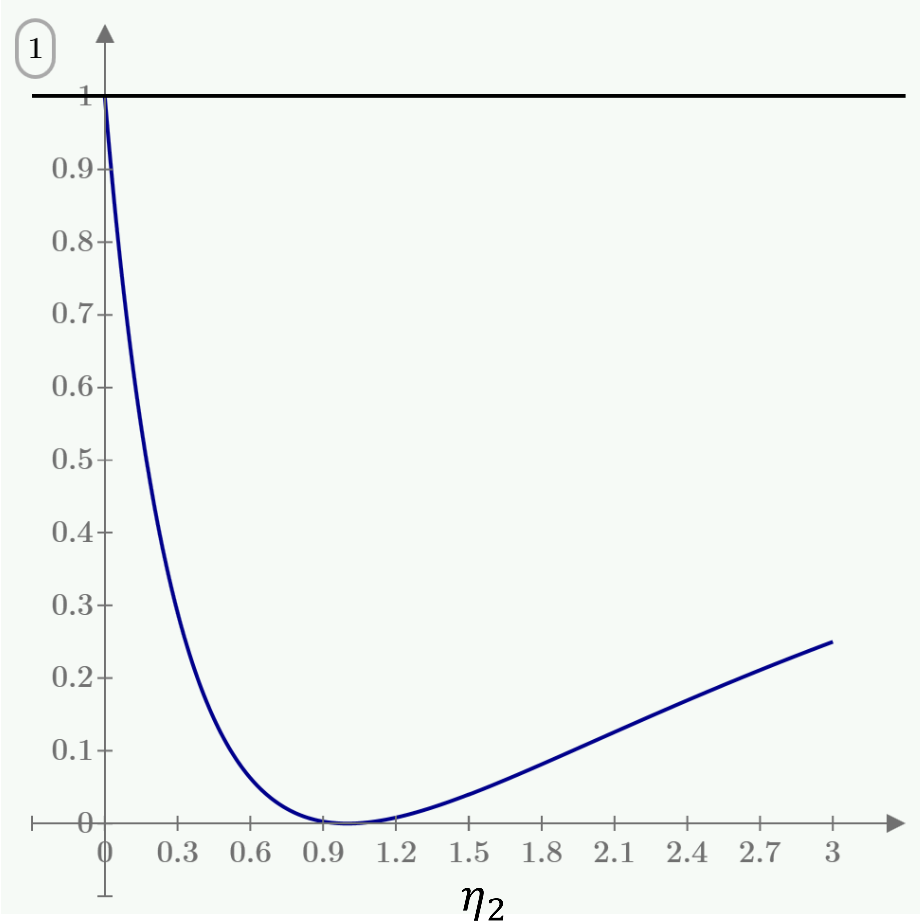
For conductive materials (i.e. metals), the effect of Fresnel reflectance is subtle



For dielectric, or non-conducting, materials, the effect is very strong: reflectance of only ~4% at normal incidence, but 100% at grazing angle

Reflectance $F(0°)$ or $F0$

$$\left(\frac{\eta_1 - \eta_2}{\eta_1 + \eta_2}\right)^2 = F(0°)$$



$\eta_1 = 1.000293$

Schlick's Approximation

For the case of normal incidence there is no distinction between s and p polarization

- Formula for approximating the contribution of the Fresnel factor in the specular reflection of light from a **non-conducting** surface

Sometimes parameterized as F_{90}

$$F(F_0, \theta_h) = F_0 + (\mathbf{1} - F_0)(1 - \cos\theta_h)^5$$

$$\text{where } F_0 = \left(\frac{\eta_1 - \eta_2}{\eta_1 + \eta_2}\right)^2, \cos\theta_h = \omega_i \cdot \hat{\boldsymbol{m}} = \omega_o \cdot \hat{\boldsymbol{m}}$$

- η_1 (mostly air in our case) and η_2 (e.g. 1.46 in case of plastic) are the indices of refraction of the two media at the interface and F_0 is the reflection coefficient for light incoming parallel to the half-way vector $\hat{\boldsymbol{m}}$ (i.e., the value of the Fresnel term when $\theta_h = 0$ or minimal reflection)

Smith Masking-shadowing Function G for TR-GGX

- If masking and shadowing are statistically independent (not correlated) events even in close proximity (severe approximation)

$$G(\omega_0, \omega_i) = G_1(\omega_0)G_1(\omega_i) \text{ where } G_1(\omega) = \frac{1}{1+\Lambda(\omega)} \text{ is the masking function}$$

- A more physically plausible formulation assuming that masking and shadowing are (height) correlated events

$$G(\omega_0, \omega_i) = \frac{1}{1 + \Lambda(\omega_0) + \Lambda(\omega_i)}$$

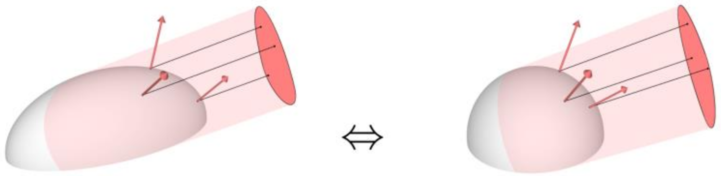
- In both cases, the auxiliary function Λ for the TR-GGX NDF reads

$$\Lambda(\omega) = \frac{\sqrt{1 + \alpha^2(\tan(\theta))^2} - 1}{2} = \frac{\sqrt{1 + \alpha^2\left(\frac{1}{(\cos(\theta))^2} - 1\right)} - 1}{2}$$

GGX VNDF Sampling Routine

Note that this method of sampling a microfacet normal assumes all directions in the local coordinate system

```
// Input Ve: view direction
// Input alpha_x, alpha_y: roughness parameters
// Input U1, U2: uniform random numbers
// Output Ne: normal sampled with PDF D_Ve(Ne) = G1(Ve) * max(0, dot(Ve, Ne)) * D(Ne) / Ve.z
vec3 sampleGGXVNDF(vec3 Ve, float alpha_x, float alpha_y, float U1, float U2) {
  // Section 3.2: transforming the view direction to the hemisphere configuration
  vec3 Vh = normalize(vec3(alpha_x * Ve.x, alpha_y * Ve.y, Ve.z));

  // Section 4.1: orthonormal basis (with special case if cross product is zero)
  float lensq = Vh.x * Vh.x + Vh.y * Vh.y;
  vec3 T1 = lensq > 0 ? vec3(-Vh.y, Vh.x, 0) * inversesqrt(lensq) : vec3(1,0,0);
  vec3 T2 = cross(Vh, T1);

  // Section 4.2: parameterization of the projected area
  float r = sqrt(U1);
  float phi = 2.0 * M_PI * U2;
  float t1 = r * cos(phi);
  float t2 = r * sin(phi);
  float s = 0.5 * (1.0 + Vh.z);
  t2 = (1.0 - s)*sqrt(1.0 - t1*t1) + s*t2;

  // Section 4.3: reprojection onto hemisphere
  vec3 Nh = t1*T1 + t2*T2 + sqrt(max(0.0, 1.0 - t1*t1 - t2*t2))*Vh;

  // Section 3.4: transforming the normal back to the ellipsoid configuration
  vec3 Ne = normalize(vec3(alpha_x * Nh.x, alpha_y * Nh.y, std::max<float>(0.0, Nh.z)));

  return Ne;
}
```



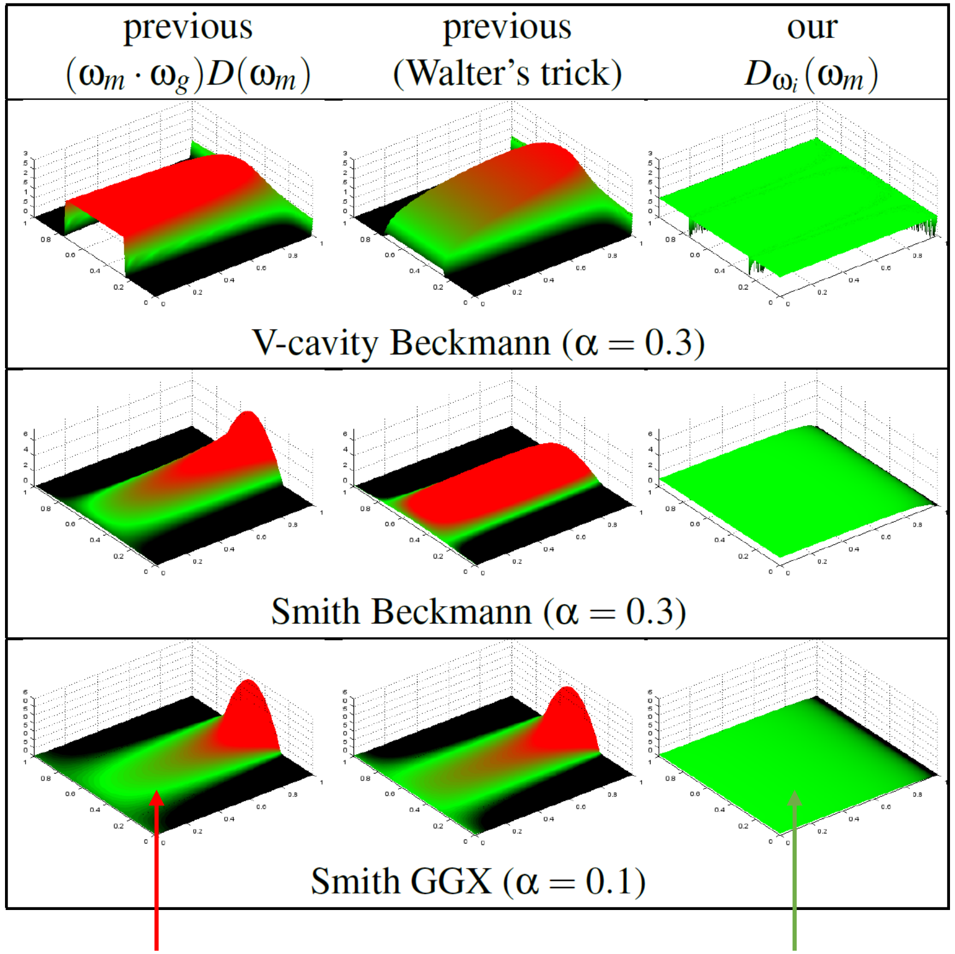
Sampling the GGX distribution of visible normals (VNDF) is equivalent to sampling the projected area of an ellipsoid, which can be mapped to sampling the projected area of a hemisphere

Source: HEITZ, Eric. Sampling the GGX distribution of visible normals. *Journal of Computer Graphics Techniques Vol*, 2018, 7.4.

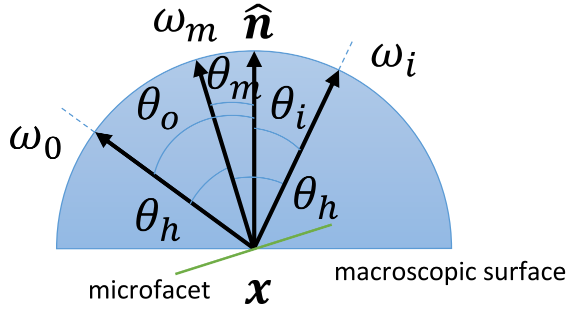
GGX VNDF Sampling Routine

- Note that the GGX VNDF sampling scheme completely fills the sample space with weights very near 1.0 and firefly artifacts do not appear. Because this method never generate backfacing normals, it does not waste the sampling space like previous methods

- HEITZ, Eric; D'EON, Eugene. Importance sampling microfacet-based BSDFs using the distribution of visible normals. In: *Computer Graphics Forum*. 2014. p. 103-112.



GGX Microfacet BRDF



- Typical (e.g. GGX) specular component of microfacet BRDF has the following form

$$f_r^{GGX}(\omega_i, \omega_o) = \frac{D(\theta_n)\, F(\theta_h)\, G(\omega_i, \omega_o)}{4 \cos\theta_o \cos\theta_i}$$

- GGX VNDF sampling routine generates microfacet normals with probability

Distribution of visible normals (VNDF) Jacobian of the reflection operator

$$p(\omega_i) = \frac{D(\theta_n) G_1(\omega_o) \cos\theta_h}{\cos\theta_o} \frac{1}{4\cos\theta_h}$$

- Then the MC sample of surface reflectance multiplied by $\cos\theta_i$ from RE equals to

$$\frac{f_r^{GGX}(\omega_i, \omega_o) \cos\theta_i}{p(\omega_i)} = \frac{\frac{\cancel{D(\theta_n)}\, F(\theta_h)\, G(\omega_i, \omega_o)}{4 \cancel{\cos\theta_o} \cancel{\cos\theta_i}} \cancel{\cos\theta_i}}{\frac{\cancel{D(\theta_n)} G_1(\omega_o) \cancel{\cos\theta_h}}{\cancel{\cos\theta_o}} \frac{1}{4 \cancel{\cos\theta_h}}} = \frac{F(\theta_h)\, G(\omega_i, \omega_o)}{G_1(\omega_o)}$$

GGX Microfacet BRDF

- The final and simplified formulation of GGX VNDF-sampling BRDF after canceling most of the microfacet BRDF terms with the PDF terms is as follows

$$f_r^{GGX}(\omega_i, \omega_o) = \frac{F(\theta_h)\, G(\omega_i, \omega_o)}{G_1(\omega_o)}$$

provided that $\cos(\theta_i) = 1$ and $pdf(\omega_i) = 1$ in our MC estimator of reflected radiance in rendering equation

$$L_r^{GGX}(\boldsymbol{x}, \omega_0) = \frac{1}{N}\sum_{i=1}^{N} \frac{L_i(\boldsymbol{x}, \omega_i) f_r^{GGX}(\omega_i, \omega_o) \cos\theta_i}{pdf(\omega_i)}$$

$$= \frac{1}{N}\sum_{i=1}^{N} L_i(\boldsymbol{x}, \omega_i) f_r^{GGX}(\omega_i, \omega_o)$$

Roughness vs. Shininess

- To convert between the roughness m in Beckmann distribution and shininess γ in (Blinn-)Phong distribution, the following formula is used

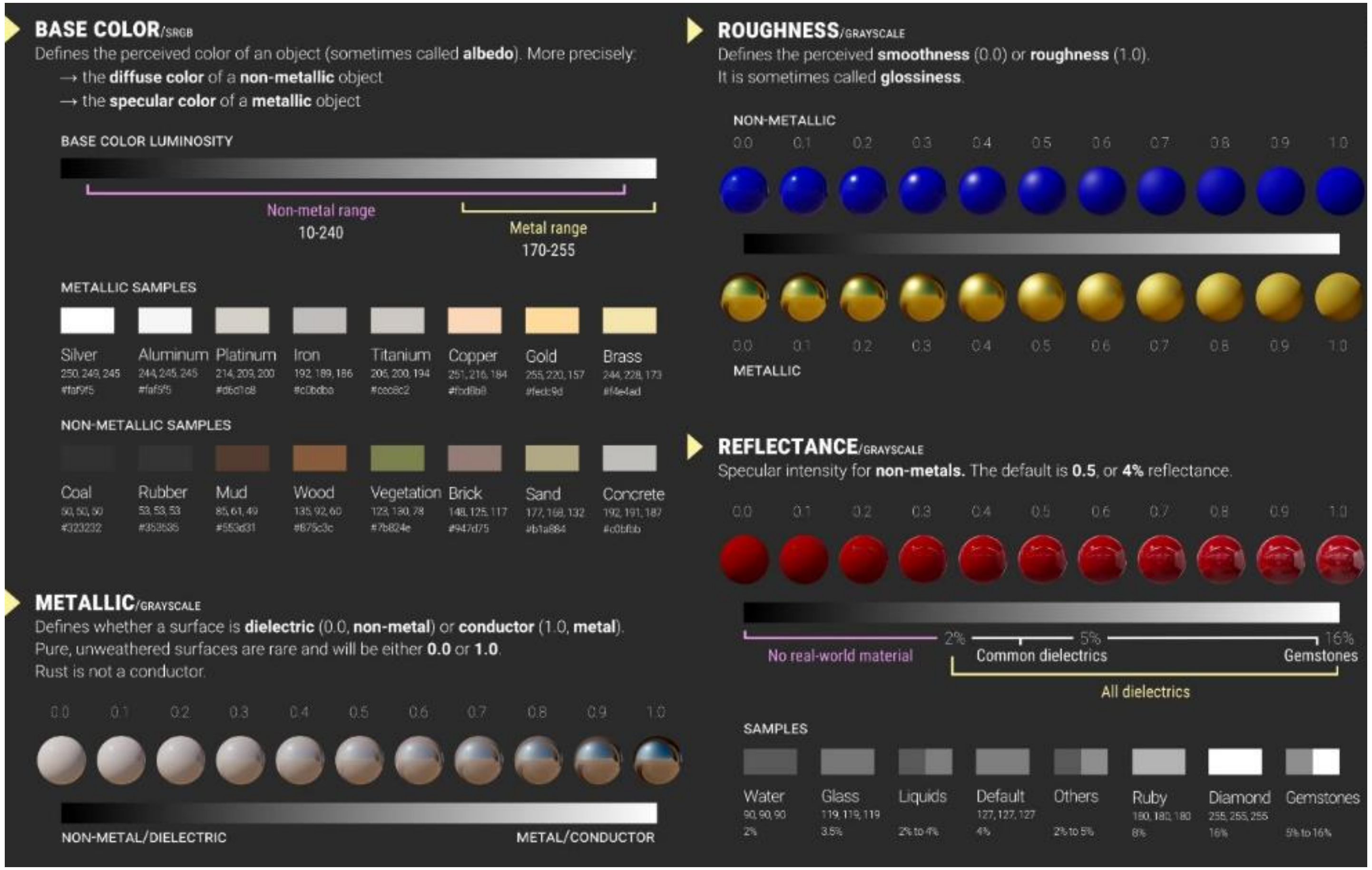
$$m = \sqrt{\frac{2}{\gamma + 2}}$$

which gives a very similar result when both refractive index and roughness m are small. When ior > 10 and $m > 0.5$ then the two distributions start to show difference and the difference will get larger when both m and ior are getting larger.

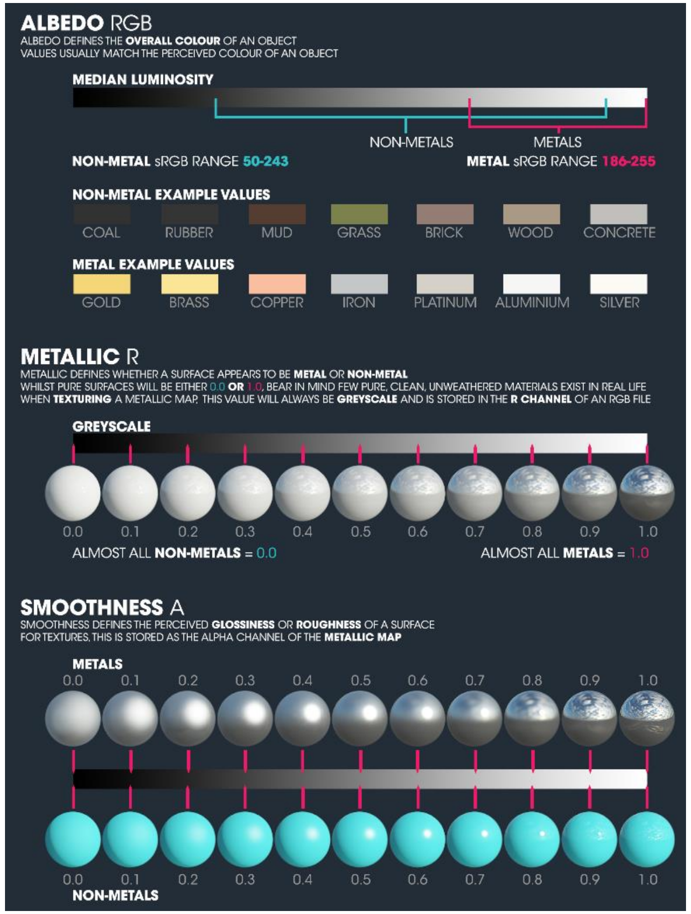
PBR Workflow

- Physically-based material workflows:
  - Metallic-Roughness workflow
    - **Base color (albedo) ≠ diffuse** is represented as a color map without any lighting in the range 30-240 sRGB (for dielectrics) or pure black color (for conductors)
    - **Metalicity** is typically a binary (or linearly interpolated grayscale) texture containing 0‘s (dielectrics) and 1‘s (metals)
    - **Roughness** – a grayscale linear texture in the range 0 (smooth) and 1 (rough)
  - Specular-Glossines workflow
    - **Diffuse** (Albedo) – RGB map
    - **Specular** – RGB map
    - **Glossines** – a grayscale linear texture that describes the surface irregularities that cause light diffusion. It is the inverse of the roughness map

Filament PBR Materials



Unity PBR Materials



PBR Textures and Materials

- On-line sources of free seamless PBR textures with Diffuse, Normal, Displacement, Occlusion, Specularity and Roughness maps:
    - https://cc0textures.com
    - https://texturehaven.com
    - https://www.poliigon.com
    - https://freepbr.com
    - https://3dtextures.me