Importance Sampling Microfacet-Based BSDFs using the Distribution of Visible Normals

Eric Heitz¹ & Eugene d'Eon²

¹INRIA ; CNRS ; Univ. Grenoble Alpes ²The Jig Lab



Dielectric material (e.g. water, plastic)



Rough dielectric material





Conductor material (e.g. metal)





Rough conductor material





Microfacet theory: statistical model of the rough interface and its interaction with light.



microfacet normals

State of the art in microfacet theory

Walter et al., Microfacet Models for Refraction through Rough Surfaces, EGSR 2007

• Physical model \rightarrow the equations.

 $\begin{aligned} \mathsf{BRDF} \quad & f_r(\omega_i, \omega_o) = \frac{F(\omega_i, \omega_{h_r}) G_2(\omega_i, \omega_o, \omega_{h_r}) D(\omega_{h_r})}{4 |\omega_i \cdot \omega_g| |\omega_o \cdot \omega_g|} \\ \mathsf{BTDF} \quad & f_t(\omega_i, \omega_o) = \frac{|\omega_i \cdot \omega_{h_t}| |\omega_o \cdot \omega_{h_t}|}{|\omega_i \cdot \omega_g| |\omega_o \cdot \omega_g|} \frac{n_0^2 (1 - F(\omega_i, \omega_{h_t})) G_2(\omega_i, \omega_o, \omega_{h_t}) D(\omega_{h_t})}{(n_i (\omega_i \cdot \omega_{h_t}) + n_o (\omega_o \cdot \omega_{h_t}))^2} \end{aligned}$

• Importance sampling \rightarrow how to solve the equations.



Choose a direction at random.

Available in Mitsuba (Physically Based Renderer)

8. Plugin reference	8.2. Surface scattering models
8.2.5. Rough dielectric material (roughdielectric)	
8. Plugin reference	8.2. Surface scattering models
8.2.7. Rough conductor material (roughconductor)	

Mitsuba's documentation

Rendering with Mitsuba (path tracing+MIS)

32 spp

128 spp



A dielectric glass plate (n = 1.5) with anisotropic Beckmann roughness ($\alpha_x = 0.05$, $\alpha_y = 0.4$).

Rendering with Mitsuba (path tracing+MIS)

512 spp

4096 spp



A dielectric glass plate (n = 1.5) with anisotropic Beckmann roughness ($\alpha_x = 0.05$, $\alpha_y = 0.4$).

What if we had perfect BSDF importance sampling?



What if we had perfect BSDF importance sampling?

Previous (512 spp)



Perfect importance sampling with precomputed data (512 spp)



 \triangle This is NOT what we propose to do. But this shows that there is clearly room for improvement.

OK, so why not just use perfect BSDF importance sampling?



Per-BSDF precomputed data?

• OK for "toy-scenes" with a low number of BSDFs.



only 1 BSDF in this scene

 Not affordable for complex scenes with a huge number of different BSDFs.

Example: Textured Assets

- Parametric BRDF models (roughness parameters α_x, α_y)
- Per-texel parameters
- GigaBytes of textures



 ∞ number of BSDFs in this scene

Courtesy of Christian Bense

Example: Textured Assets

- Parametric BRDF models (roughness parameters α_x, α_y)
- Per-texel parameters
- GigaBytes of textures



Roughness map (α_x, α_y)



 ∞^2 number of BSDFs in this scene

We propose

A better BSDF importance sampling scheme with lower variance, not perfect, but that can be used in production rendering.

Fulfilled requirements

- Parametric BSDF models used in production (typically Beckmann and GGX)
- Varying anisotropic roughness parameters α_x, α_y
- No assumption on the kind of integrator (bidir, MLT, etc.)
- Plug & Play (trivial update of an existing implementation of the previous method)

Our approach

We investigate the physical "meaning" of the BSDF model and use it to our advantage.

$$f_{r}(\omega_{i},\omega_{o}) = \frac{F(\omega_{i},\omega_{h_{r}})G_{2}(\omega_{i},\omega_{o},\omega_{h_{r}})D(\omega_{h_{r}})}{4|\omega_{i}\cdot\omega_{g}||\omega_{o}\cdot\omega_{g}|}$$
$$f_{t}(\omega_{i},\omega_{o}) = \frac{|\omega_{i}\cdot\omega_{h_{t}}||\omega_{o}\cdot\omega_{h_{t}}|}{|\omega_{i}\cdot\omega_{g}||\omega_{o}\cdot\omega_{g}|}\frac{n_{0}^{2}\left(1-F(\omega_{i},\omega_{h_{t}})\right)G_{2}(\omega_{i},\omega_{o},\omega_{h_{t}})D(\omega_{h_{t}})}{\left(n_{i}(\omega_{i}\cdot\omega_{h_{t}})+n_{o}(\omega_{o}\cdot\omega_{h_{t}})\right)^{2}}$$

Looks complicated.

But the model behind the equations is actually very simple and intuitive.



How to sample outgoing directions?



Available information: the distribution of normals.



Use it to generate a random normal sample.



Apply a transport operator on the normal ("reflect" or "transmit").



Importance sampling is based on the distribution of normals.



But in the physical model, the incident ray can only intersect normals that are *visible*.



In the physical model, the transport operators are applied only on *visible* normals.

distribution of visible normals distribution of normals





This difference is what makes importance sampling inefficient.

distribution of visible normals weight = 0.2 distribution of normals





samples

weighhs 0.2



distribution of normals

weight = 0.3



samples

weights 0.2 0.3



weights 0.2 0.3 0







Problem 1: sampling space wasting.






The Problem with the Previous Method Problem 2: high sampling weights.



23

Importance sampling with the -distribution of normals distribution of visible normals.



Our importance sampling method does exactly what is represented in this picture.



Only one component of the model is missing: most of the rays leave the surface...



...but other are occluded (microsurface shadowing).



Previous 512 spp (88.9s) Our 408 spp (87.1s)

A dielectric glass plate (n = 1.5) with anisotropic Beckmann roughness ($\alpha_x = 0.05$, $\alpha_y = 0.4$).

same rendering time

Previous 256 spp (194.5s)



Our 192 spp (198.9s)



A rough dielectric with isotropic GGX roughness ($\alpha = 0.05$). same rendering time



A rough conductor with isotropic GGX roughness ($\alpha = 0.10$).

same rendering time

Previous 8192 spp (31.1min)

Our 1024 spp (5.2m)



A rough conductor with isotropic GGX roughness ($\alpha = 0.15$).

8× less samples per pixel 6× less rendering time

The distribution of visible normals

$$D_{\omega_i}(\omega_m) = \frac{G_1(\omega_i, \omega_m)}{\cos \theta_i} \langle \omega_i, \omega_m \rangle D(\omega_m, \alpha_x, \alpha_y)$$

- $D(\omega_m, \alpha_x, \alpha_y)$ distribution of normals
- (α_x, α_y) roughness parameters
- $G_1(\omega_i, \omega_m)$ masking function
- $\langle \omega_i, \omega_m \rangle$ cosine projection factor

▲ Sampling this PDF is not simple!



microfacet normals ⇔ microsurface slope



scaling the roughness \Leftrightarrow stretching the microsurface



The distribution of visible normals is stretch-invariant.





Example: we want to sample D_{ω_i} for $\alpha = \frac{1}{2}$.



First, we stretch into a configuration where $\alpha = 1$.



Now, since $\alpha = 1$ we know how to sample D_{ω_i} .



Finally, we go back to the initial configuration.



Implementing the sampling scheme for $\alpha = 1$



Spherical coordinates

$$\begin{split} & \omega_m \Leftrightarrow (\theta_m, \phi_m) \\ \text{The PDF of } \theta_m \text{ depends on } \theta_i \text{ and } \phi_m \\ \text{The PDF of } \phi_m \text{ depends on } \theta_i \text{ and } \theta_m \end{split}$$

2D PDFs difficult 🔅









1. Main idea: sample the distribution of visible normals.

- In practice: sampling D_{ωi}(ω_m, α_x, α_y) is not simple ^(*)
 2.1 Stretch invariance: solve only for α_x = 1 and α_y = 1.
 2.2 Slope space: sample only two 1D PDFs.
 Now it's simple ^(*)
- Implementation (C++ code in our supplemental material)
 3.1 Beckmann and GGX distributions: analytic sampling.
 - 3.2 Other distributions: generic sampling with precomputed data.
- 4. Validation: passes χ^2 test in Mitsuba.

- 1. Main idea: sample the distribution of visible normals.
- In practice: sampling D_{ωi}(ω_m, α_x, α_y) is not simple ^(C)
 2.1 Stretch invariance: solve only for α_x = 1 and α_y = 1.
 2.2 Slope space: sample only two 1D PDFs.
 Now it's simple ^(C)
- Implementation (C++ code in our supplemental material)
 3.1 Beckmann and GGX distributions: analytic sampling.
 - $3.2\,$ Other distributions: generic sampling with precomputed data.
- 4. Validation: passes χ^2 test in Mitsuba.

- 1. Main idea: sample the distribution of visible normals.
- 2. In practice: sampling $D_{\omega_i}(\omega_m, \alpha_x, \alpha_y)$ is not simple B
 - 2.1 Stretch invariance: solve only for $\alpha_x = 1$ and $\alpha_y = 1$.
 - 2.2 Slope space: sample only two 1D PDFs.

Now it's simple ③

- 3. Implementation (C++ code in our supplemental material) 3.1 Beckmann and GGX distributions: analytic sampling.
 - 3.2 Other distributions: generic sampling with precomputed data.

4. Validation: passes χ^2 test in Mitsuba.

- 1. Main idea: sample the distribution of visible normals.
- 2. In practice: sampling $D_{\omega_i}(\omega_m, \alpha_x, \alpha_y)$ is not simple B
 - 2.1 Stretch invariance: solve only for $\alpha_x = 1$ and $\alpha_y = 1$.
 - 2.2 Slope space: sample only two 1D PDFs.

Now it's simple ③

- 3. Implementation (C++ code in our supplemental material) 3.1 Beckmann and GGX distributions: analytic sampling.
 - 3.2 Other distributions: generic sampling with precomputed data.
- 4. Validation: passes χ^2 test in Mitsuba.

Conclusion

Conclusion

Contribution

- Efficient BSDF Importance sampling
- Easy to implement
- Analytic for Beckmann and GGX distributions

Lessons Learned

- Physical models have often a "meaning"
- Spend time working out the intuitions behind the math

Conclusion

Contribution

- Efficient BSDF Importance sampling
- Easy to implement
- Analytic for Beckmann and GGX distributions

Lessons Learned

- Physical models have often a "meaning"
- Spend time working out the intuitions behind the math

Questions?

The distribution of visible normals

$$D_{\omega_i}(\omega_m) = \frac{G_1(\omega_i, \omega_m)}{\cos\theta_i} \langle \omega_i, \omega_m \rangle D(\omega_m, \alpha_x, \alpha_y)$$

- $D(\omega_m, \alpha_x, \alpha_y)$ distribution of normals
- $G_1(\omega_i, \omega_m)$ masking function

Is it actualy a PDF? I.e.

$$\int_{\Omega} \frac{G_1(\omega_i, \omega_m)}{\cos \theta_i} \langle \omega_i, \omega_m \rangle D(\omega_m, \alpha_x, \alpha_y) \, d\omega_m \stackrel{?}{=} 1$$

The distribution of visible normals

$$D_{\omega_i}(\omega_m) = \frac{G_1(\omega_i, \omega_m)}{\cos\theta_i} \langle \omega_i, \omega_m \rangle D(\omega_m, \alpha_x, \alpha_y)$$

- $D(\omega_m, \alpha_x, \alpha_y)$ distribution of normals
- $G_1(\omega_i, \omega_m)$ masking function

Is it actualy a PDF? I.e.

$$\int_{\Omega} \frac{G_1(\omega_i, \omega_m)}{\cos \theta_i} \langle \omega_i, \omega_m \rangle D(\omega_m, \alpha_x, \alpha_y) \, d\omega_m \stackrel{?}{=} 1$$

Masking functions G_1 in the CG literature

Smith Cook & Torrance V-cavities Implicit Kelemen Schlick-Smith


Masking functions G_1 in the CG literature



Does it work?

$$\int_{\Omega} \frac{G_1(\omega_i, \omega_m)}{\cos \theta_i} \langle \omega_i, \omega_m \rangle D(\omega_m, \alpha_x, \alpha_y) \, d\omega_m \stackrel{?}{=} 1$$

What does it mean for a masking function to be "correct" or "exact"?

The masking function is "correct" or "exact".

\Leftrightarrow

The distribution of visible normals is a PDF, i.e.

$$\int_{\Omega} \frac{G_1(\omega_i, \omega_m)}{\cos \theta_i} \langle \omega_i, \omega_m \rangle D(\omega_m, \alpha_x, \alpha_y) \, d\omega_m = 1$$

The distribution of visible normals is NOT a PDF, i.e.

$$\int_{\Omega} \frac{G_1(\omega_i, \omega_m)}{\cos \theta_i} \langle \omega_i, \omega_m \rangle D(\omega_m, \alpha_x, \alpha_y) \, d\omega_m \neq 1$$

 \Leftrightarrow

The masking function is NOT "correct" or "exact" and should NOT be called "physically based".

Name	D_{ω_i} is a PDF?	"physically based"?
Smith		
Cook & Torrance V-cavities		
Implicit		
Kelemen		
Schlick-Smith		

Name	D_{ω_i} is a PDF?	"physically based"?
Smith	1	1
Cook & Torrance V-cavities		
Implicit		
Kelemen		
Schlick-Smith		

Name	D_{ω_i} is a PDF?	"physically based"?
Smith	1	✓
Cook & Torrance V-cavities	1	1
Implicit		
Kelemen		
Schlick-Smith		

Name	D_{ω_i} is a PDF?	"physically based"?
Smith	✓	✓
Cook & Torrance V-cavities	1	1
Implicit	×	×
Kelemen	×	×
Schlick-Smith	×	×

Masking functions G_1 in the CG literature

Name	D_{ω_i} is a PDF?	"physically based"?
Smith	1	✓
Cook & Torrance V-cavities	1	1
Implicit	×	×
Kelemen	×	×
Schlick-Smith	×	×

More details in *"Understanding the Masking-Shadowing Function in Microfacet-Based BRDFs"*, Eric Heitz, JCGT 2014, to appear.