Microfacet model
and Microfacet-based BRDF

2019.05.16
20186413 Murat Gaspard
Physically based microfacet BRDFs

\[ L_o = \int_{\Omega_+} L_i \cdot f_r \cdot \cos \theta_i \cdot d\omega_i \]
Microfacet Theory

From Hakyeong Kim's talk:
Materials = microsurfaces

Microsurfaces properties can be manipulated

\[ f_r = \frac{F \cdot D \cdot G}{4 \cdot \cos \theta_i \cdot \cos \theta_o} \]
Papers

• **Extracting Microfacet-based BRDF Parameters from Arbitrary Materials with Power Iterations**
  Jonathan Dupuy  Eric Heitz  Pierre Poulin  Victor Ostromoukhov
  Eurographics Symposium on Rendering 2015

• **Fast Global Illumination with Discrete Stochastic Microfacets Using a Filterable Model**
  Beibei Wang  Lu Wang  Pierre Poulin  Nicolas Holzschuch
  Pacific Graphics 2018
Extracting Microfacet-based BRDF Parameters from Arbitrary Materials with Power Iterations

Jonathan Dupuy  Eric Heitz  Pierre Poulin  Victor Ostromoukhov

Eurographics Symposium on Rendering 2015
Context

Real
(real materials)

Digital
(microfacet BRDFs)
Context

Real
(real materials)

Digital
(microfacet BRDFs)

How to retrieve the microsurface from real material?
Microfacet BRDF Fitting

Approach:
- Fitted microsurface
- Minimize fitting metrics

Current limitations:
- Robustness / Speed
- Arbitrary metrics
- Reproducibility
Contribution

Idea:
• Find the NDF
• Approximize the Fresnel term

Properties:
• Robustness
• Simplicity
• Speed
• Reproducibility

Input
NDF
Fresnel
- Tabulated
- GGX
- Beckmann
Microfacet theory

Assumption:
Single-bounce mirror reflection dominates on the microsurface

\[ f_r = \frac{F(\theta_d) D(h) G(i,o)}{4 \cos \theta_i \cos \theta_0} \]
Microfacet theory

Assumption:

Single-bounce mirror reflection dominates on the microsurface

\[ f_r = \frac{F(\theta_d)D(h)G(i,o)}{4 \cos \theta_i \cos \theta_0} \]

Halfway vector,

\[ h = \frac{i + o}{||i + o||} \]
Microfacet theory

Assumption:

Single-bounce mirror reflection dominates on the microsurface

\[ f_r = \frac{F(\theta_d)D(h)G(i,o)}{4 \cos \theta_i \cos \theta_0} \]

\( \theta_d = \text{difference angle in the BRDF parameterization of Rusinkiewicz} \)

\( \theta_d = \arccos(i \cdot h) \in [0, \pi/2] \)
Microfacet slopes

Goal:

Simplified the search of the NDF

Normalization constraint on the NDF:

\[ \int_{\Omega_+} D(h) \cos \theta_h \, d\omega_h \]
Microfacet slopes

Goal:
Simplified the search of the NDF

Idea:
Instead of searching in the horizontal space ($\Omega_+$), we search into the slopes space ($\mathbb{R}^2$)

In the $\Omega$ set, normals and slopes are linked through the bijection

$$\tilde{\mathbf{h}} = \begin{bmatrix} -\tan \theta \cos \phi = \tilde{x}_k \\ -\tan \theta \sin \phi = \tilde{y}_k \end{bmatrix}, \quad \tilde{\mathbf{h}} \in \mathbb{R}^2$$
Microfacet slopes

Goal:
Simplified the search of the NDF

Idea:
Instead of searching in the horizontal space ($\Omega_+$), we search into the slopes space ($\mathbb{R}^2$)
In the $\Omega+$ set, normals and slopes are linked through the bijection

$$\tilde{h} = \begin{bmatrix} -\tan \theta \cos \phi h = \tilde{x}_k \\ -\tan \theta \sin \phi h = \tilde{y}_k \end{bmatrix}, \quad \tilde{h} \in \mathbb{R}^2$$

Normal Distribution function

$$f_r = \frac{F(\theta_d) D(h) G(i, \theta)}{4 \cos \theta_i \cos \theta_0} \quad D(h) = P(\tilde{h}) \sec^4 \theta_h$$

Probability distribution function $P$

Normalisation constraint:
$$\int_{\mathbb{R}^2} P(\tilde{h}) d\tilde{h} = 1$$
**Microfacet theory**

Geometric attenuation factor

\[
f_r = \frac{F(\theta_d)D(h)G(i,o)}{4 \cos \theta_i \cos \theta_0}
\]

\[
G(i, o) = \frac{G_1(i)G_1(0)}{G_1(i)+G_1(0)-G_1(i)G_1(0)} \quad G \in [0,1]
\]

Smith monostatic shadowing function:

\[
G_1(k) = \frac{\cos \theta_k}{\int_{\Omega_+} kh D(h) \, d\omega_k} \quad G_1 \in [0,1]
\]
Backscattering Equation

Mathematical (previous work)

We focus on backscattering configuration which reduce the dimensionality of the BRDF

\[ \mathbf{l} = \mathbf{o} = \mathbf{h} \quad \theta_d = 0 \]

\[ f_r = \frac{F_0 D(o) G(o,o)}{4 \cos^2 \theta_0} \]

\[ G(o, o) = G_1(o) \]

\[ f_r = \frac{F_0 D(o) G_1(o)}{4 \cos^2 \theta_0} \]
Eigensystem construction

\[ f_r = \frac{F_0 D(o) G_1(o)}{4 \cos^2 \theta_0} \]

(Fredholm equation of the second kind)

\[ F_0 P(\tilde{\sigma}) = \int_{\Omega^+} K(o,h) P(\tilde{h}) \, d\omega_h \]

\[ K(o, h) = 4f_r(o, o) \cos^5 \theta_o \, oh \sec^4 \theta_h. \]
Eigensystem construction

\[ f_r = \frac{F_0 D(o) G_1(o)}{4 \cos^2 \theta_0} \]

\[ F_0 P(\tilde{\omega}) = \int_{\Omega^+} K(o,h) P(\tilde{h}) \, d\omega_h \]

Numerically solved by discretizing the equation with a quadrature rule

\[ F_0 P(\tilde{\omega}_i) = \sum_{j=1}^{N} w_j K(o_i, h_j) P(\tilde{h}_j) \]
Eigensystem construction

\[ F_0 P(\bar{o}_i) = \sum_{j=1}^{N} w_j K(o_i, h_j) P(\bar{h}_j) \]

\[ F_0 p = K \cdot p \]

\[ p = (P(\bar{o}_1), \cdots, P(\bar{o}_N))^t \]

\[ K = \begin{bmatrix} w_1 K(o_1, h_1) & \cdots & w_N K(o_1, h_N) \\ \vdots & \ddots & \vdots \\ w_1 K(o_N, h_1) & \cdots & w_N K(o_N, h_N) \end{bmatrix} \]
Eigensystem construction

\[ f_r = \frac{F_0 D(o) G_1(o)}{4 \cos^2 \theta_0} \]

\[ F_0 \cdot p(\tilde{o}) = K \cdot p \]

Perron-Frobenius theorem
the solution is always the eigenvector with the largest magnitude
Backscattering Equation

\[ f_r = \frac{F_0 D(o) G(o, o)}{4 \cos^2 \theta_0} \]

Algorithm 1 Extract \( P \)

```plaintext
function EXTRACT_P(f_r, N)
    for each \( i, j \in [1, N] \) do
        \( K_{i,j} \leftarrow w_j 4 f_r(o_i, o_l) \cos^5 \theta_{o_i, o_j} h_j \sec^4 \theta_{h_j} \)
    end for
    \( p \leftarrow (1, \ldots, 1)^T \)
    for \( 0 \leq i < M \) do
        \( p \leftarrow K \cdot p \)
    end for
    \( P \leftarrow \text{normalize}(p) \)
end function
```
Ideal Mirrors

Special microfacet BRDF:
- Fresnel term is 1
- Independent of wavelength

\[ f_{r, id} = \frac{D(o)G_1(o)}{4 \cos \theta_0 \cos \theta_i} \]
Fresnel Extraction

We compute an average response:
- Fully automatic
- Simple implementation
- Fast evaluation
- Works well in practice

\[ F(\theta_d) = \mathbb{E} \left[ \frac{f_r}{f_{r,id}} \middle| ih = \cos \theta_0 \right] \]
Fresnel Extraction

We compute an average response:
- Fully automatic
- Simple implementation
- Fast evaluation
- Works well in practice

\[ F(\theta_d) = \mathbb{E} \left[ \frac{f_r}{f_{r, id}} \bigg| \mathbf{i} \mathbf{h} = \cos \theta_0 \right] \]
Validation

29 gold-metallic-paint

Renderings

SGD  Reference  Ours: Tabulated

Ours: Gaussian  Reference  Ours: GGX
Validation

29 gold-metallic-paint

Renderings

SGD fitting optimization failed and resulted in flawed images

SGD
Reference
Ours: Tabulated

Ours: Gaussian
Reference
Ours: GGX
Accuracy

Mean delta-E difference image on the MERL database
Speed

(Intel i5-2500 @ 3.30 GHz)
Microfacet BRDFs

\[ f_r = \frac{F \cdot D \cdot G}{4 \cdot \cos \theta_i \cdot \cos \theta_o} \]

Modular components

- Fresnel term \( F \)
- Distribution of normals \( D \)
- Roughness \( \alpha \)
- Geometric factor \( G \)

Artistic Control

\[ D = P \left( \frac{\tilde{x}_h}{\alpha_x}, \frac{\tilde{y}_h}{\alpha_y} \right) \frac{\sec^4 \theta_h}{\alpha_x \alpha_y} \]

Tabulate the slope PDF

- Roughness \( \propto \) stretch\(^{-1}\)
- Efficient BRDF evaluation
- Efficient BRDF sampling
Fast Global Illumination with Discrete Stochastic Microfacets Using a Filterable Model
Beibei Wang  Lu Wang  Pierre Poulin  Nicolas Holzschuch

Pacific Graphics 2018
Goal

Original

Without Glints
Previous work

Jakob et al.

Discrete Stochastic Microfacet Models

Flakes-like small mirror facets
Previous work

Jakob et al.

**Discrete Stochastic Microfacet Models**

These specular patches are organized in a hierarchy.
Discrete Stochastic Microfacet Model

Extend the microsurface BRDF model to take into account a finite extend in \textit{space} and \textit{angle}

\[ \frac{\hat{f}_r(A, \omega_i, \Omega_0)}{a(A)\sigma(\Omega_0)} \int_A \int_{\Omega_0} f_r(x, \omega_i, \omega_o) d\omega_o dx \]

\( A = \text{finite area around } x \)
\( \Omega_0 = \text{finite solid angle around outgoing direction } \omega_o \)

\[ f_r(x, \omega_i, \omega_o) = \frac{F(\omega_i, \omega_o)D(x, \omega_o)G(\omega_i, \omega_o, \omega_h)}{4 \cos \theta_i \cos \theta_0} \]
Discrete Stochastic Microfacet Model

We now assume the surface is made of discrete small mirror particles instead of a continuous microfacets

\[
\hat{f}_r(A, \omega_i, \Omega_0) = \frac{1}{a(A)\sigma(\Omega_0)} \int_A \int_{\Omega_0} f_r(x, \omega_i, \omega_o) d\omega_o dx
\]

\[
\hat{f}_r(A, \omega_i, \Omega_0) = \frac{(\omega_i \cdot \omega_o) F(\omega_i, \omega_o, \Omega)}{a(A)\sigma(\Omega_0)4(\omega_i \cdot n)(\omega_i \cdot n)} \tilde{D}(x, \omega_o) G(\omega_i, \omega_o, \omega_h)
\]

\[
\tilde{D}(x, \Omega_h) = \frac{1}{N} \sum_i^N \Omega_h(\omega_h^k)1_A(x^k)
\]

Sum of a finite set of particles

A = pixel footprint
Discrete Stochastic Microfacet Model

\[
\tilde{D}(x, \Omega_h) = \frac{1}{N} \sum_{i}^{N} 1_{\Omega_h}(\omega^i_h) 1_A(x^i)
\]

Sum of a finite set of particles

A = pixel footprint

4 dimensional normal distribution
Count the number of particles in the 4 dimensional domain A × Ω_h
Filterable Discrete Stochastic Microfacet Model

Replace the particle count with a particle probability function
Introduce a Directional Probability Function (DPF)

\[ P(\omega_i, \omega_o, \gamma) = \text{probability a particle exist that reflect lights incoming from direction } \omega_i \text{ into a coe centered around direction } \omega_o \text{ with half angle } \gamma \]

\( x \) and \((\omega_i, \omega_o)\) are independant variables
Filterable Discrete Stochastic Microfacet Model

Replace the particle count with a particle probability function

Introduce a Directional Probability Function (DPF)

\[ P(\omega_i, \omega_o, \gamma) \approx \frac{1}{N} \sum_{k=1}^{N} 1_{\Omega_h}(\omega_h^k) \]

\[ \hat{D}(A, \omega_i, \omega_o) = \left( \frac{1}{N} \sum_{k=1}^{N} 1_A(x^k) \right) P(\omega_i, \omega_o, \gamma) \]

\[ \hat{D}(A, \omega_i, \omega_o) = \frac{1}{N} \sum_{k=1}^{N} 1_A(x^k) H(\lambda(x) - P(\omega_i, \omega_o, \gamma)) \]

\[ H(u) = \begin{cases} 
1, & \text{if } u > 0, \\
0, & \text{otherwise.} 
\end{cases} \]

x and \((\omega_i, \omega_o)\) are independant variables
Filterable Discrete Stochastic Microfacet Model

\[ \hat{D}(A, \omega_i, \omega_o) = \frac{1}{N} \sum_{k=1}^{N} 1_{A}(x^k) H(\lambda(x) - P(\omega_i, \omega_o, \gamma)) \]

\[ H(u) = \begin{cases} 
1, & \text{if } u > 0, \\
0, & \text{otherwise.} 
\end{cases} \]

\[ \lambda(x) = \text{uniformly Distributed random value from Tiny Encryption Algorithm} \]

\[ \lambda(x) \text{ is used to assign a random value between 0 and 1 to each positio x} \]
Filterable Discrete Stochastic Microfacet Model

**Preprocess**
compute and store $P$ for $\gamma = 1^\circ$, by sampling regularly in each dimension, computing and storing the value for $P$. 

![Diagram](image)
Filterable Discrete Stochastic Microfacet Model

Preprocess

- Build a hierarchical representation of the Directional Probability Function, using Gaussian blur
- Each level is generated by blurring the finest level
- 9 hierarchical levels, for 1°, 2°, 5°, 10°, 20°, 30°, 45°, 60° and 90°
Filterable Discrete Stochastic Microfacet Model

2 steps hierarchical traversal
- Angle
- Space

Look up in the precomputed table;
Select the appropriate hierarchical level;
Extract the precomputed value for P
Filterable Discrete Stochastic Microfacet Model

2 steps hierarchical traversal
- Angle
- Space

Traversal in texture space;
Assume flakes are uniformly distributed in space;

Algorithm 1 Spatial Traversal

```plaintext
function D_r(A, \omega_1, \omega_2)
    query = A
    queue = node(N_{_nm})
    count = 0
    p = P(\omega_1, \omega_2)
    while queue \neq \emptyset do
        node = queue.pop()
        if node \cap query == \emptyset or [node] = 0 then pass
        elseif node \subseteq query then n_k = [node]
            for i \in n_k do \psi = \Lambda(k)
                if \psi > p then count = count + 1
            end if
        end for
        else if error criterion satisfied then
            overlap = (node \cap query).vol()/node.vol()
            n_k = [node]
            for i \in n_k do \psi = \Lambda(k)
                if \psi > p then count = count + overlap
            end if
        end if
        else
            for c in node.split() do queue.push(c)
        end for
    end while
    return count
end function
```
Filterable Glint Computation

- For each path, compute the path footprint at each light bounce
- Compute the average glint contribution at this footprint

\[ \hat{D}(A, \omega_i, \omega_o) = a(A) \times P(\omega_i, \omega_o, \gamma), \]

- If the glint count is larger than a given threshold, use the average contribution.
- Otherwise, use the separable model described
Discrete Stochastic Microfacets Using a Filterable Model

Idea:
If a footprint cover a large surface,
- individual glint are not noticeable;
- average contribution

In practice:
Filterable model preafer for
- material far from camera;
- Several bounce in global illumination
Validation

Our model (with filtering)
Total: 4.6 min, Cost: +21%

Our model (no filter)
Total: 7.4 min, Cost: +95%

Original [JHY*14],
Total: 24.4 min, Cost: +542%

Without Glints,
Total: 3.8 min
Speed

(Intel i7 (40 cores) with 32 GB of main memory @ a 2.20GHz)
Speed

(Intel i7 (40 cores) with 32 GB of main memory @ a 2.20GHz)

![Graph 1: Rendering Time vs. Particle Count](image1)

![Graph 2: Rendering Time vs. Floor Material Roughness](image2)
Speed

(Intel i7 (40 cores) with 32 GB of main memory @ a 2.20GHz)
Quiz

- In ‘Extracting Microfacet-based BRDF Parameters from Arbitrary Materials with Power Iterations’, a bijection is used to link ( ) space and the slopes space ( )

  a. \( \Omega_d \) and \( \mathbb{R}^4 \)  
  b. \( \Omega_+ \) and \( \mathbb{R}^2 \)  
  c. \( \mathbb{R}^2 \) and \( \Omega_+ \)  
  d. \( \mathbb{R}^2 \) and \( \Omega_d \)

- In ‘Fast Global Illumination with Discrete Stochastic Microfacets Using a Filterable Model’, which variables do we consider for indépendance?

  a. \( x \) and \( (\omega_i, \omega_o) \)  
  b. \( x \) and \( \gamma \)  
  c. \( \gamma \) and \( (\omega_i, \omega_o) \)  
  d. \( \omega_h \) and \( \gamma \)