KAIST

Extending microfacet-based normal mapping

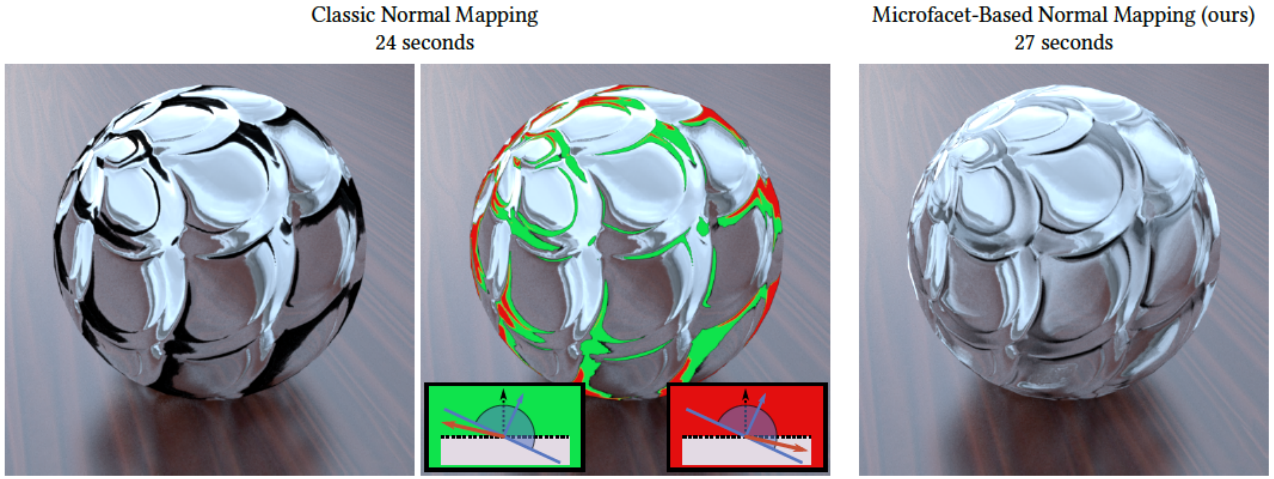
Team 4
20186413 Gaspard Murat
20193001 Dahyun Kang
20193164 Hakyeong Kim

VISUAL COMPUTING Lab

Table of Contents

- Introduction
- Problem Statement
- Goal
- Experiment & Result
- Summary

Introduction

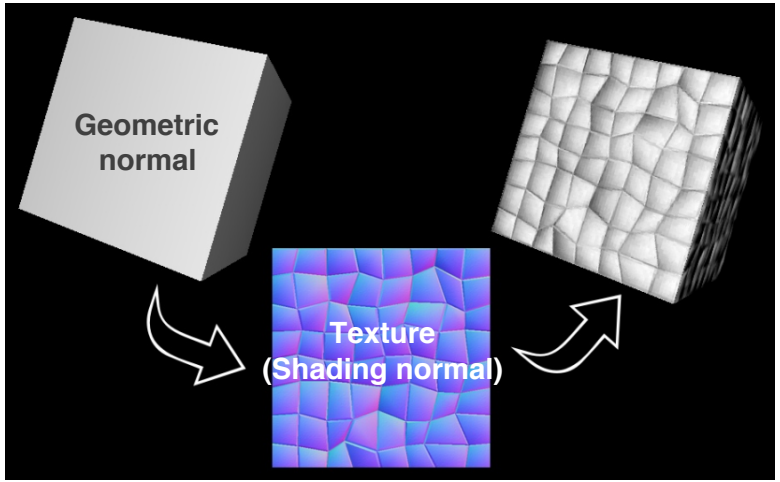


Microfacet-based Normal Mapping for Robust Monte Carlo Path Tracing

VINCENT SCHÜSSLER, ERIC HEITZ, JOHANNES HANIKA, CARSTEN DACHSBACHER

Introduction

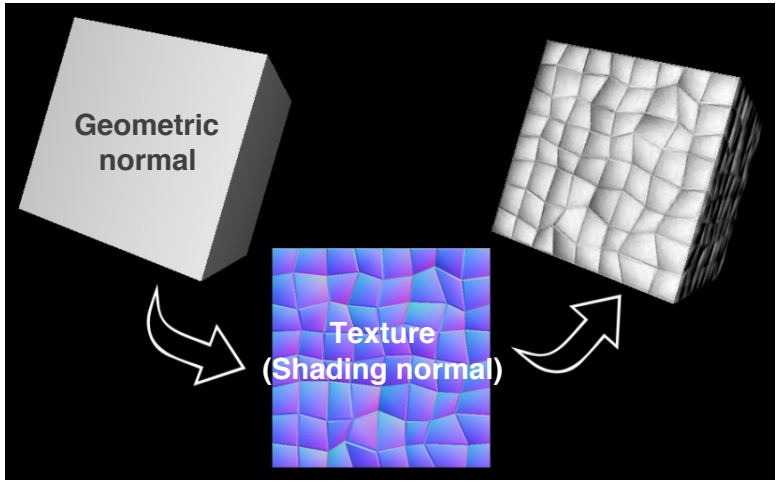
l Normal map



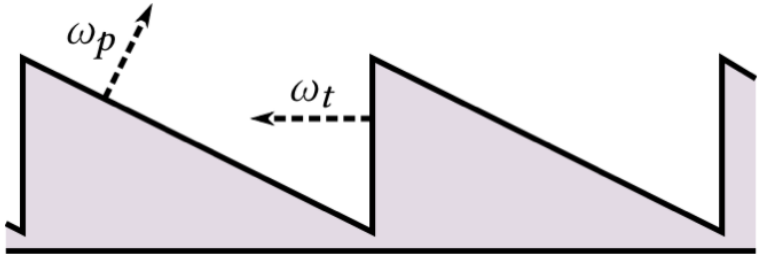
Problem:
Breaks the consistency of light transport
Cause flawed approach for PBRT(Monte Carlo Path Tracing)

Introduction

l Normal map



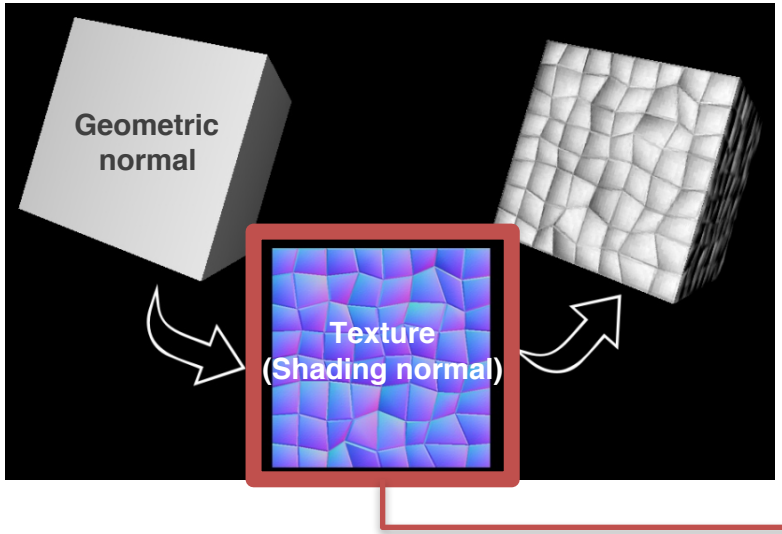
l Microfacet-based Normal mapping



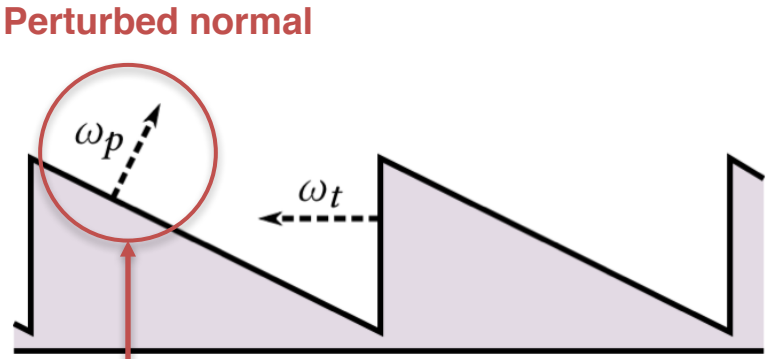
Problem:
Breaks the consistency of light transport
Cause flawed approach for PBRT

Introduction

l Normal map



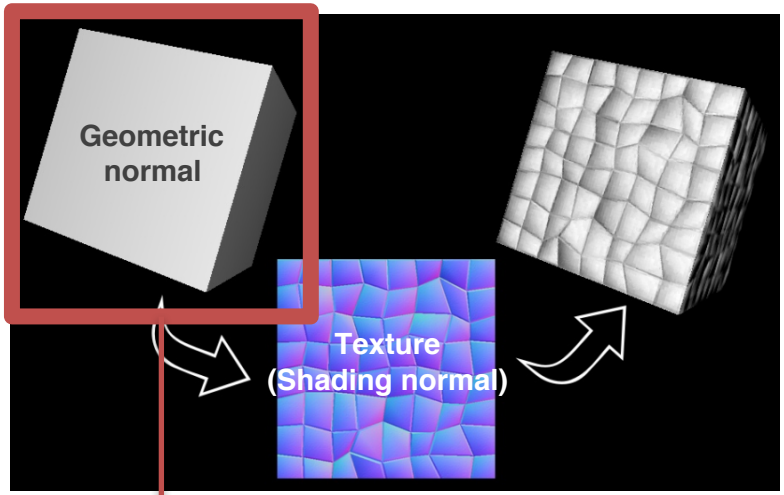
l Microfacet-based Normal mapping



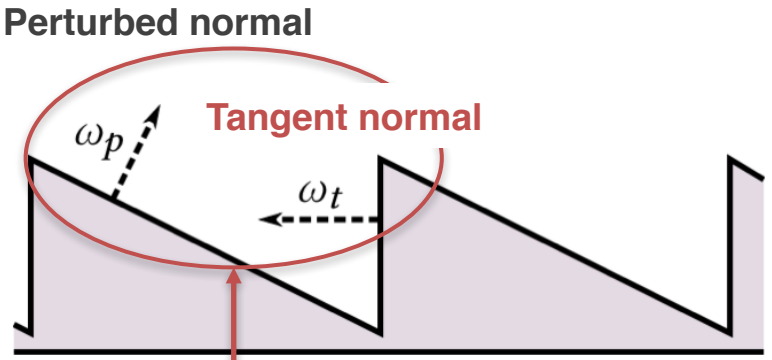
Introduction

KAIST

Normal map



Microfacet-based Normal mapping



Average normal of microsurface remains geometric normal

VISUAL COMPUTING Lab

Introduction

KAIST

l Paper Result

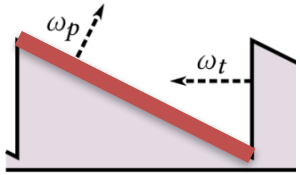
VISUAL COMPUTING Lab

Problem statement

Addressed problem.

1. Limited support of different material types

Micro-BRDF of perturbed face

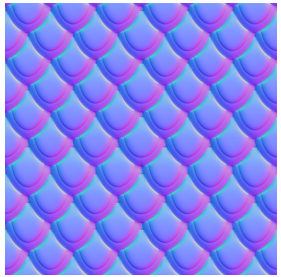


= Rough conductor

= Rough dielectric

Wanted image

Dielectric material
w/o normal mapping



Microfacet model that support
shading normal mapping

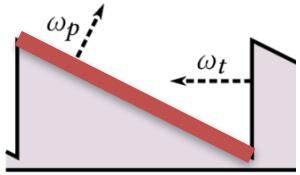
VISUAL COMPUTING Lab

Problem statement

Addressed problem.

1. Limited support of different material types

Micro-BRDF of perturbed face

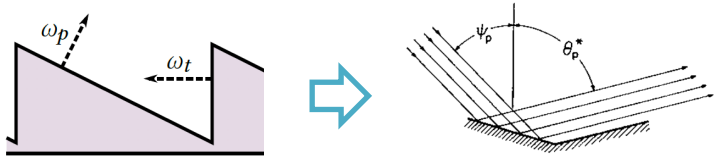


= Rough conductor

= Rough dielectric

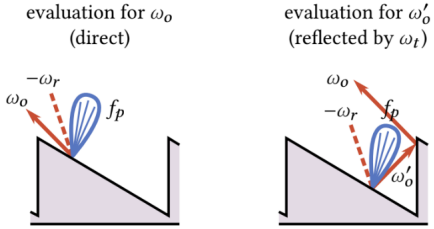
Further extent of implementation

2. Variation of microgeometry



3. Analytic solution of multiple scattering

- Speed enhancement



VISUAL COMPUTING Lab

Goal

Limited support of different material types

1. Support transmittance term in microfacet-based normal mapping

Variation of microgeometry

2. Experiment microgeometry with non-tangent microfacet

Analytic solution for multiple scattering

3. Analyze assumptions of the problem and explore for the solution

Study 1. Support Transmittance

KAIST

Author's implementation of microfacet-based normal mapping

Mitsuba-plugin : inherit BSDF class

```
43  class NormalMapMicrofacet : public BSDF {
44  public:
45      NormalMapMicrofacet(const Properties &props) : BSDF(props), rng(new Random) {
```

3 main function that implements the algorithm

```
119     Spectrum eval(const BSDFSamplingRecord &bRec_, EMeasure measure) const {
```

- Evaluate the **BSDF** $f(w_i, w_o)$

```
133     Float pdf(const BSDFSamplingRecord &bRec, EMeasure measure) const {
```

- Compute the probability of sampling ***bRec.wo*** (given ***bRec.wi***).

```
138     Spectrum sample(BSDFSamplingRecord &bRec, const Point2 &sample) const {
```

- Sample the BSDF and return the importance weight
- (i.e. the value of the BSDF divided by the probability density of the sample).

Study 1. Support Transmittance

KAIST

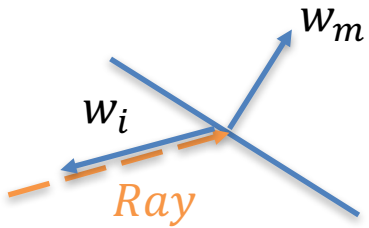
1) Modify function eval

```
119    Spectrum eval(const BSDFSamplingRecord &bRec_, EMeasure measure) const {
```

Calls evalMirror according to paper's algorithm implementation

```
Spectrum evalMirror(const BSDFSamplingRecord &bRec, EMeasure measure) const {
        if (measure != ESolidAngle
                || Frame::cosTheta(bRec.wi) <= 0
                || Frame::cosTheta(bRec.wo) <= 0)
                //|| Frame::cosTheta(bRec.wi) <= 0
                //|| Frame::cosTheta(bRec.wo) <= 0
                )
                return Spectrum(0.0f);
```

When ray hits the back-face of microfacet



Author's code stops evaluation
→ Delete to support transparency

VISUAL COMPUTING Lab

Study 1. Support Transmittance

KAIST

1) Modify function eval

```
119    Spectrum eval(const BSDFSamplingRecord &bRec_, EMeasure measure) const {
```

Calls evalMirror according to paper's algorithm implementation

```
Spectrum evalMirror(const BSDFSamplingRecord &bRec, EMeasure measure) const {
        if (measure != ESolidAngle
                || Frame::cosTheta(bRec.wi) <= 0
                || Frame::cosTheta(bRec.wo) <= 0)
```

After sampling of wo

```
if (!singleScattering) {
        float pdfWo = m_nested->pdf(queryEvalReflection, measure);
        pdfWo /= Frame::cosTheta(queryEvalReflection.wo);
        pdfWo /= abs(Frame::cosTheta(queryEvalReflection.wo));
```

Study 1. Support Transmittance

KAIST

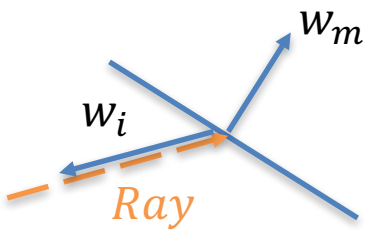
1) Modify function eval

```
119    Spectrum eval(const BSDFSamplingRecord &bRec_, EMeasure measure) const {
```

Calls evalMirror according to paper's algorithm implementation

```
Spectrum evalMirror(const BSDFSamplingRecord &bRec, EMeasure measure) const {
                .
                .
if (i >= maxScatteringOrder || generateRandomNumber() < G1_) break;
//if a ray goes down, stop traveling
if ( Frame::cosTheta(query_wp.wo)< 0.f || i >= maxScatteringOrder || generateRandomNumber() < G1_) break;
```

In case of **trasmission**
Stop randomwalk for multiple scattering and compute refracted ray



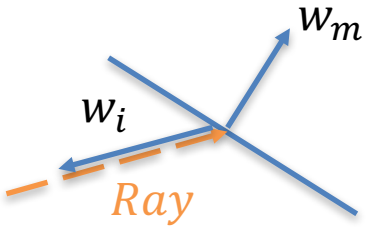
VISUAL COMPUTING Lab

Study 1. Support Transmittance

2) Modify function pdf and sample

```
133    Float pdf(const BSDFSamplingRecord &bRec, EMeasure measure) const {
138    Spectrum sample(BSDFSamplingRecord &bRec, const Point2 &sample) const {
```

Consider back-facing facet



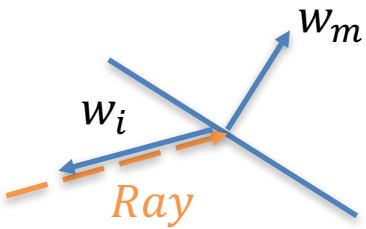

```
Float pdfMirror(const BSDFSamplingRecord &bRec, EMeasure measure) const {
        if (measure != ESolidAngle
                || Frame::cosTheta(bRec.wi) <= 0
                || Frame::cosTheta(bRec.wo) <= 0)
                //|| Frame::cosTheta(bRec.wi) <= 0
                //|| Frame::cosTheta(bRec.wo) <= 0
                )
                return 0.0f:
Spectrum sampleMirror(BSDFSamplingRecord &bRec, const Point2 &sample) const {
        if (Frame::cosTheta(bRec.wi) <= 0)
                return Spectrum(0.0f);

        //when a ray hit the backward surface, return 0
        //if (Frame::cosTheta(bRec.wi) <= 0)
        //      return Spectrum(0.0f);
```

Study 1. Support Transmittance

2) Modify function pdf and sample

```
133    Float pdf(const BSDFSamplingRecord &bRec, EMeasure measure) const {
138    Spectrum sample(BSDFSamplingRecord &bRec, const Point2 &sample) const {
```

Consider back-facing facet




```
Float pdfMirror(const BSDFSamplingRecord &bRec, EMeasure measure) const {
    BSDFSamplingRecord samplingQuery(*currentIts, NULL);
    BSDFSamplingRecord samplingQuery(*currentIts, sampler);
```

→ Set originally nullified sampler

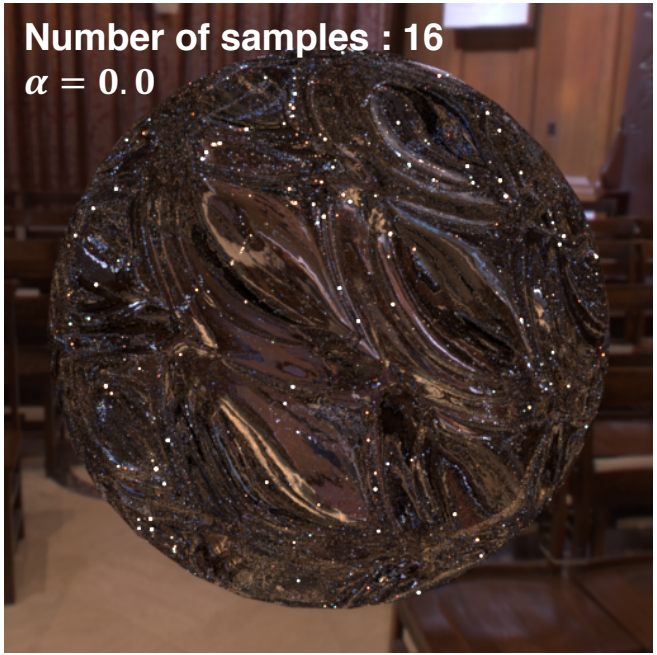
```
Spectrum sampleMirror(BSDFSamplingRecord &bRec, const Point2 &sample) const {
```

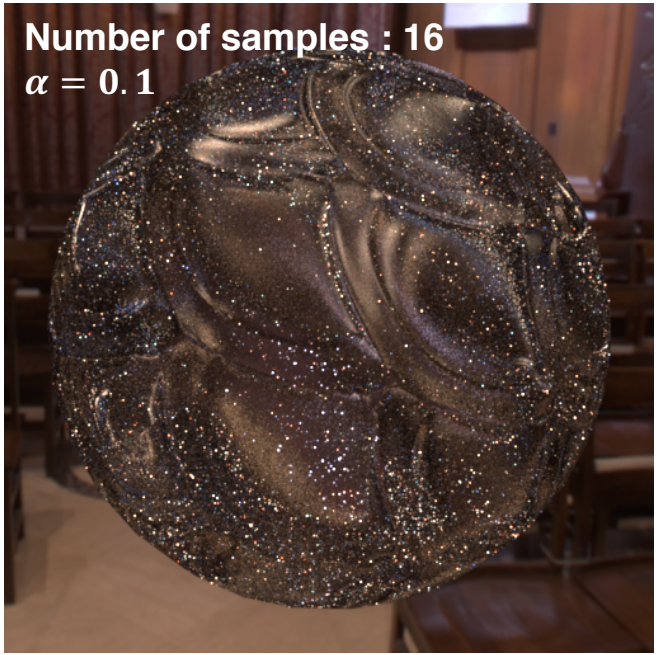
Sample for transmitting rays also

```
    //when a ray go under ground, return 0
    //if (Frame::cosTheta(bRec.wo) <= 0.) {
    //  return Spectrum(0.f);
    //}
    return energy;
```

Result 1. Support Transmittance

Microfacet-based normal mapping of **rough dielectric material**

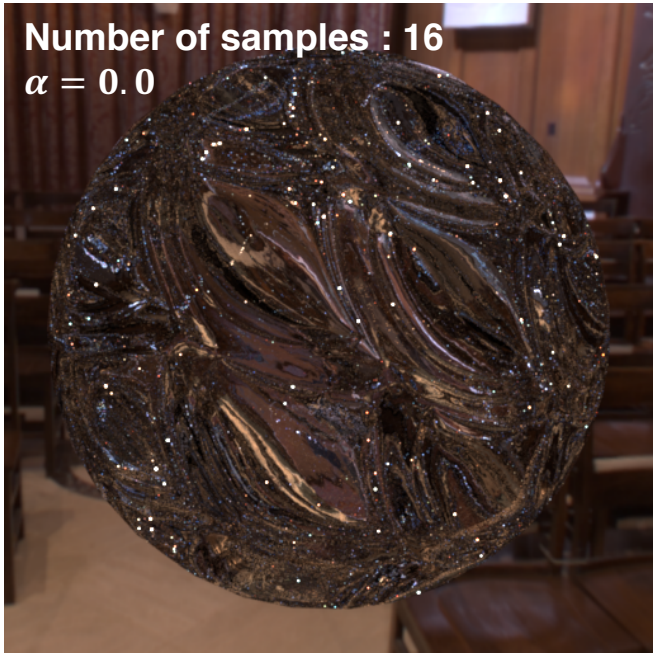


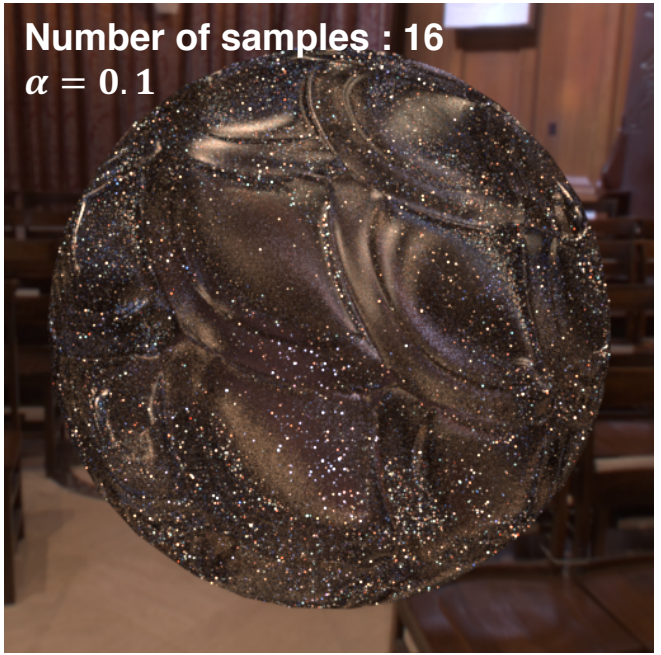


Result 1. Support Transmittance

KAIST

Microfacet-based normal mapping of **rough dielectric material**





→ Supported transmittance term of micro-BSDF in microfacet-based normal mapping

Study 2. Microgeometry

Goal:

- improving speed in the random walk algorithm
- Reducing variance

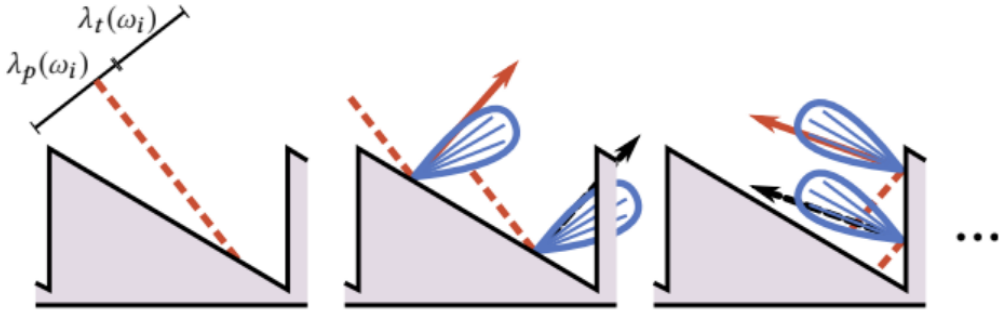


Figure x: Random walk on the microsurface

Study 2. Microgeometry

KAIST

Default of the random walk algorithm:

- Rays bounce against the surface of the microsurface.
- The more bounces, the more variance is increased and the more time consuming the algorithm is !

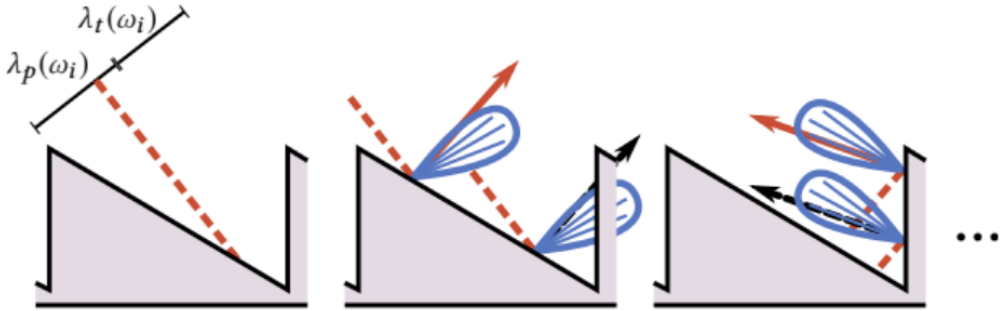
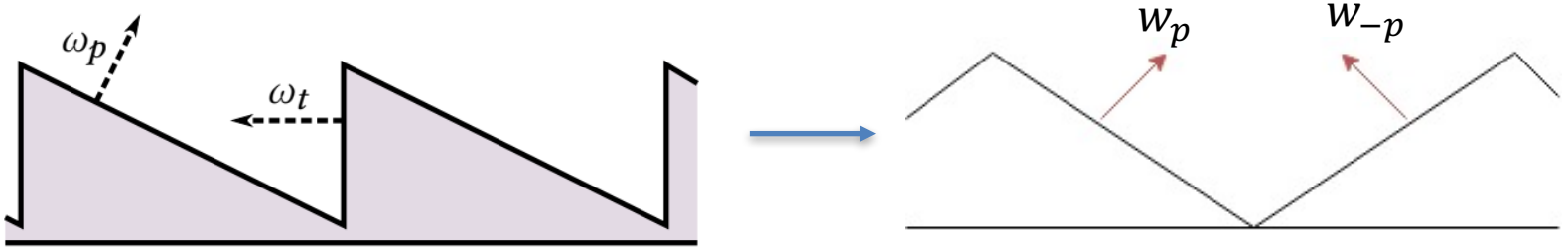


Figure x: Random walk on the microsurface

VISUAL COMPUTING Lab

Study 2. Microgeometry

First modification: classic v-groove



$$G1(\omega_i, \omega_m) = H(\langle \omega_i, \omega_m \rangle) \min[1, \frac{\langle \omega_i, \omega_g \rangle}{a_p(\omega_i) + a_t(\omega_i)}]$$

$$G1(\omega_i, \omega_m) = \min[1, \frac{\langle \omega_i, \omega_p \rangle \langle \omega_p, \omega_g \rangle}{\langle \omega_i, \omega_g \rangle}]$$

Result 2. Microgeometry

KAIST

First modification: classic v-groove

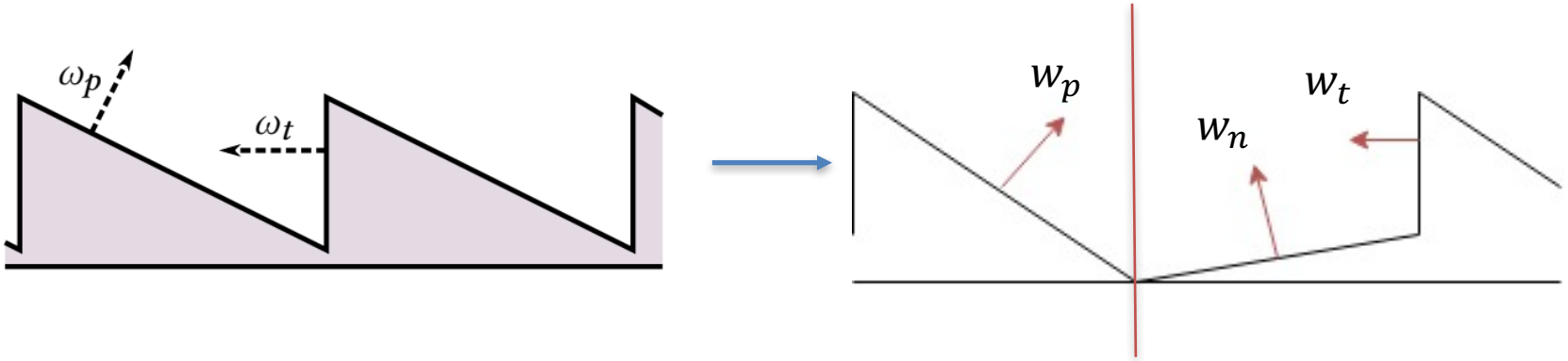
As expected, very bad result:

- not symmetric
- not energy conservative

Figure x: Golden sphere rendered using classic V-groove and random walk

Study 2. Microgeometry

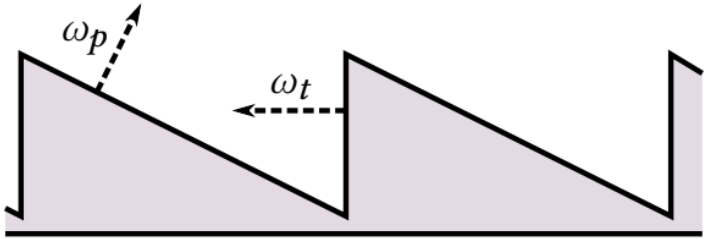
second modification: u-groove

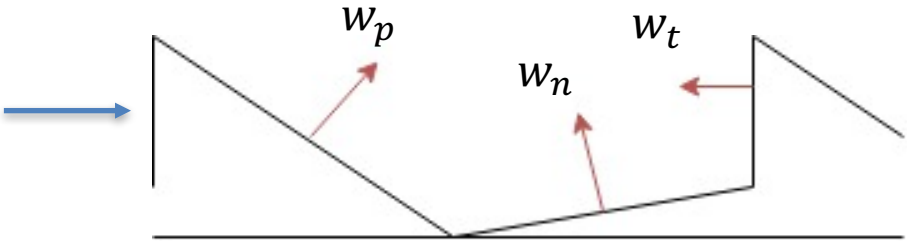


Mix of the classic V-groove model
and the V-groove model introduce by the author

Study 2. Microgeometry

second modification: u-groove



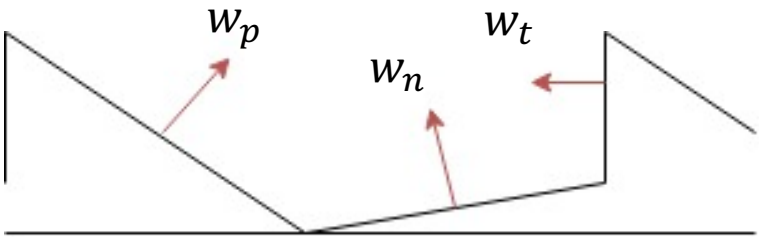


$$G1(\omega_i, \omega_m) = H(\langle \omega_i, \omega_m \rangle) \min[1, \frac{\langle \omega_i, \omega_g \rangle}{a_p(\omega_i) + a_t(\omega_i)}]$$

$$G1(\omega_i, \omega_m) = H(\langle \omega_i, \omega_m \rangle) \min\left[1, 2 \frac{\langle \omega_i, \omega_g \rangle}{\frac{\langle \omega_i, \omega_p \rangle}{\langle \omega_p, \omega_g \rangle} + \frac{\langle \omega_i, \omega_n \rangle}{\langle \omega_n, \omega_g \rangle} + \frac{\langle \omega_i, \omega_t \rangle \sqrt{1 - \langle \omega_n, \omega_g \rangle^2}}{\langle \omega_n, \omega_g \rangle}}\right]$$

Study 2. Microgeometry

second modification: u-groove



Design a **Normal Distribution Funtion** by mixing the NDF from classic V-groove and the NDF from the modified V-groove

$$D(w_m) = \frac{1}{2}\frac{\delta_p(\omega_m)}{\langle \omega_p, \omega_g \rangle} + \frac{1}{2}\left(\frac{\delta_n(\omega_m)}{\langle \omega_n, \omega_g \rangle} + \frac{\delta_t(\omega_m)\sqrt{1 - \langle \omega_n, \omega_g \rangle^2}}{\langle \omega_n, \omega_g \rangle}\right)$$

Study 2. Microgeometry

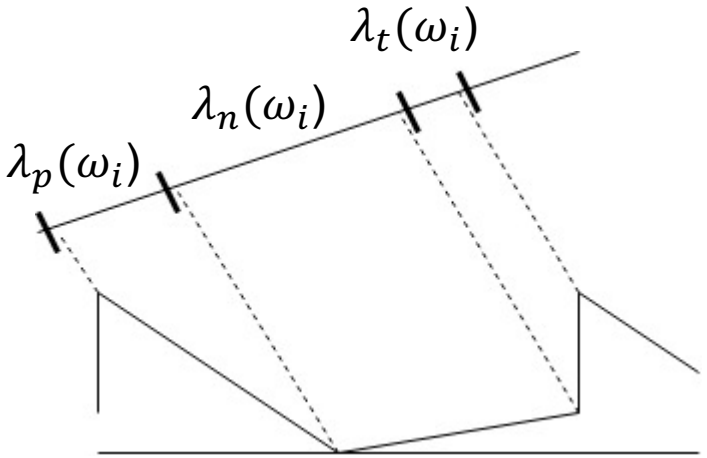
second modification: u-groove

Projected Areas and Intersection Probabilities

$$a_p(\omega_i) = \frac{1}{2}\frac{\langle \omega_i, \omega_p \rangle}{\langle \omega_p, \omega_g \rangle}$$

$$a_n(\omega_i) = \frac{1}{2}\frac{\langle \omega_i, \omega_n \rangle}{\langle \omega_n, \omega_g \rangle}$$

$$a_t(\omega_i) = \frac{1}{2}\frac{\langle \omega_i, \omega_p \rangle}{\langle \omega_p, \omega_g \rangle}\frac{\langle \omega_i, \omega_p \rangle\sqrt{1-\langle \omega_n, \omega_g \rangle^2}}{\langle \omega_n, \omega_g \rangle}$$



Study 2. Microgeometry

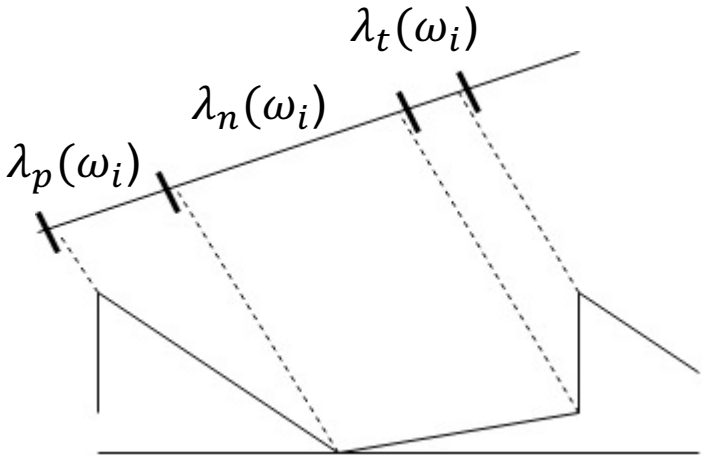
second modification: u-groove

Projected areas and Intersection Probabilities

$$\lambda_p(\omega_i) = \frac{a_p(\omega_i)}{a_p(\omega_i)+a_n(\omega_i)+a_t(\omega_i)}$$

$$\lambda_n(\omega_i) = \frac{a_n(\omega_i)}{a_p(\omega_i)+a_n(\omega_i)+a_t(\omega_i)}$$

$$\lambda_t(\omega_i) = \frac{a_t(\omega_i)}{a_p(\omega_i)+a_n(\omega_i)+a_t(\omega_i)}$$



Study 2. Microgeometry

second modification: u-groove

- With changing the microfacet model, the random walk algorithm has also to be modified.

  Assuming the ray is still in the microfacet, we need to determine which micronormal it will hit

```
213         if (generateRandomNumber() > lambda_p(wp, bRec.wi)) {
214             std::swap(currentIts, nextIts);
215             std::swap(currentWo, nextWo);
216         }
```

Generate a random number g

If (g > $\lambda_p(\omega_i)$)

Then ω_t is hit

Else

ω_p is hit

Just have to swap from one face to another

Study 2. Microgeometry

second modification: u-groove

- With changing the microfacet model, the random walk algorithm has also to be modified.

```
243         int facet = 1; //w_p is chosen
244         if (generateRandomNumber() > lambda_p(wp, bRec.wi)) {
245             facet = 2; //w_n is chosen
246             currentIts = nextIts_n;
247             currentWo = nextWo_n;
248             if (generateRandomNumber() > lambda_n(wp, bRec.wi)) {
249                 facet = 3; //w_t is chosen
250                 currentIts = nextIts_t;
251                 currentWo = nextWo_t;
252             }
253         }
```

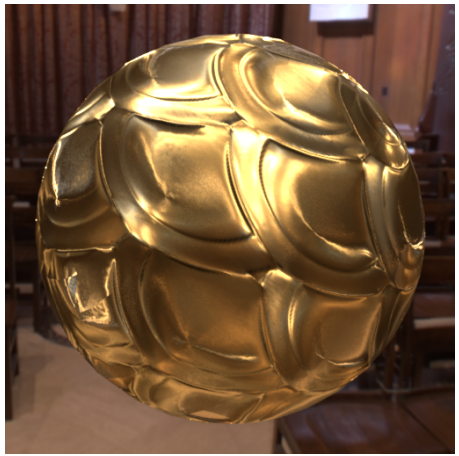
We need to keep track of the current facet, and we have to compute the next to be hit among the two other.

Study 2. Microgeometry

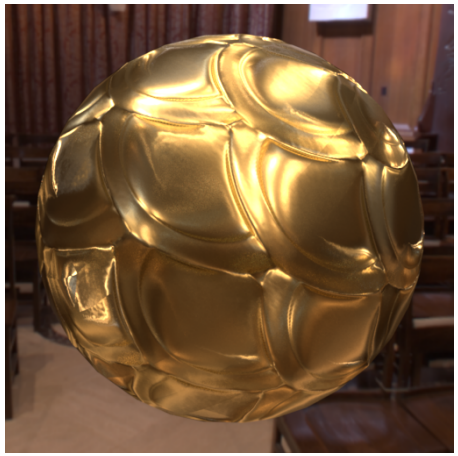
KAIST

second modification: u-groove

But the result were not good enough, and we lack time to estimate the improvement in speed or variance…



Our model



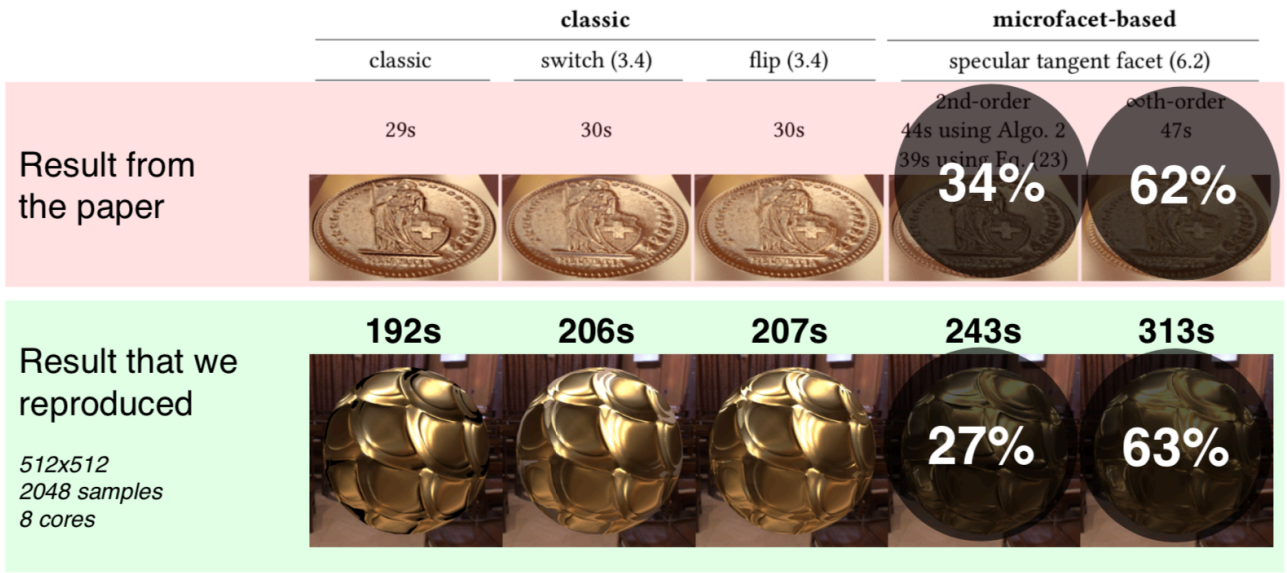
Model of Heitz et Al

Study 3. Multiple Scattering

Motivation:

The slowdown is mainly because of **random walk** process.

(used to evaluate BRDF value)



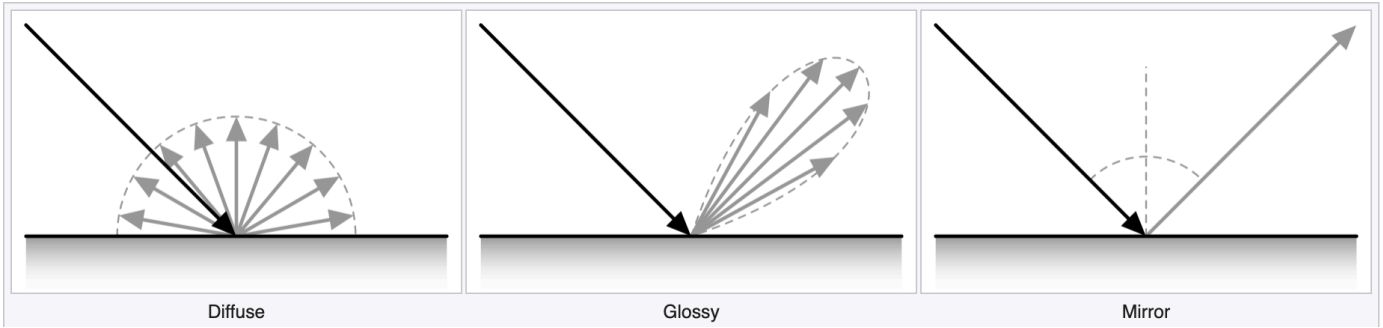
Study 3. Multiple Scattering

KAIST

Motivation:

How do we get the BRDF value at a fixed angles,

i.e. how do we evaluate the value of $f(\omega_i=\textit{fixed}, \omega_o=\textit{fixed})$?



Simple examples of BRDF

Study 3. Multiple Scattering

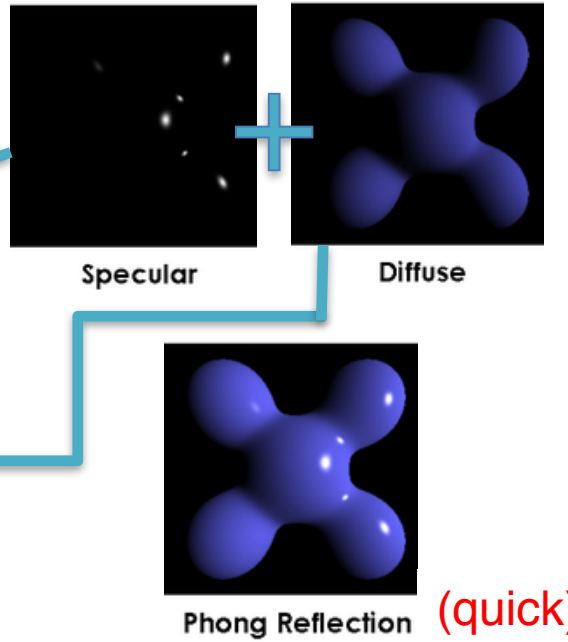
Motivation:

How do we get the BRDF value at a fixed angles,
i.e. how do we evaluate the value of $f(\omega_i=fixed, \omega_o=fixed)$?

```
Spectrum result(0.0f);
if (hasSpecular) {
    Float alpha    = dot(bRec.wo, reflect(bRec.wi)),
          exponent = m_exponent->eval(bRec.its).average();

    if (alpha > 0.0f) {
        result += m_specularReflectance->eval(bRec.its) *
            ((exponent + 2) * INV_TWOPI * std::pow(alpha, exponent));
    }
}

if (hasDiffuse)
    result += m_diffuseReflectance->eval(bRec.its) * INV_PI;

return result * Frame::cosTheta(bRec.wo);
```

O(1)

O(1)



(quick)

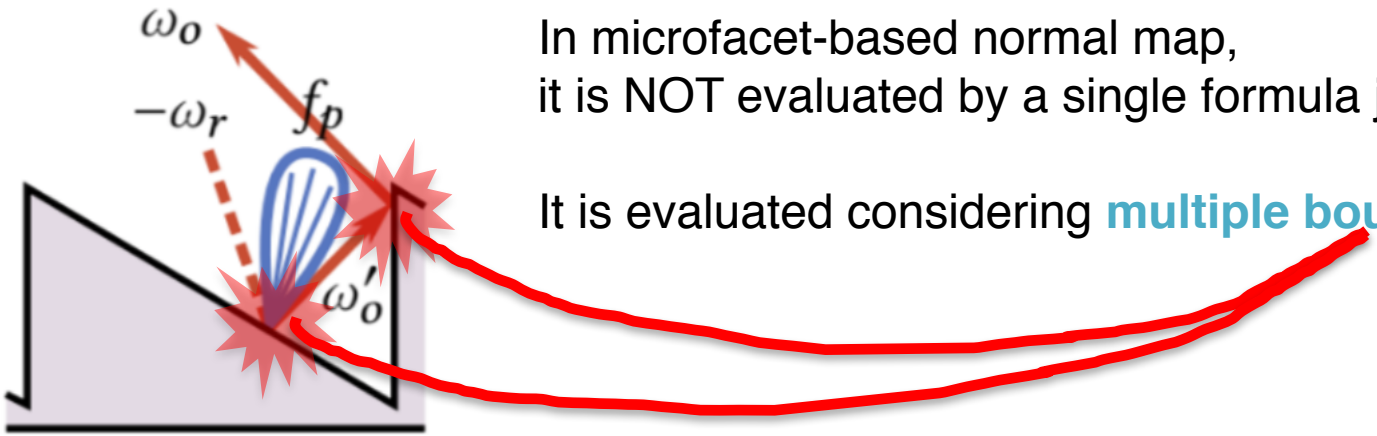
Study 3. Multiple Scattering

KAIST

Motivation:

How do we get the BRDF value at a fixed angles,

i.e. how do we evaluate the value of f(ω_i=*fixed*, ω_o=*fixed*)?



In microfacet-based normal map,
it is NOT evaluated by a single formula just as Phong.

It is evaluated considering **multiple bouncing**.

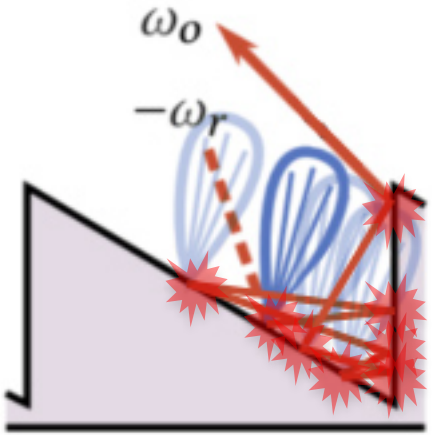
Study 3. Multiple Scattering

KAIST

Motivation:

How do we get the BRDF value at a fixed angles,

i.e. how do we evaluate the value of $f(\omega_i=\textit{fixed}, \omega_o=\textit{fixed})$?



In microfacet-based normal map,
it is NOT evaluated by a single formula just as Phong.

It is evaluated considering **multiple bouncing**.

… and even more bounces.

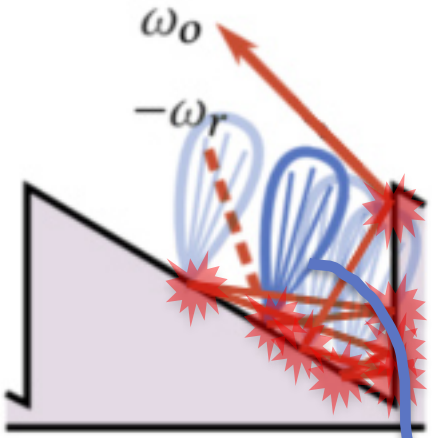
Study 3. Multiple Scattering

KAIST

Motivation:

How do we get the BRDF value at a fixed angles,

i.e. how do we evaluate the value of $f(\omega_i=\textit{fixed}, \omega_o=\textit{fixed})$?



In microfacet-based normal map,
it is NOT evaluated by a single formula just as Phong.

It is evaluated considering **multiple bouncing**.

In code, it loops through `do-while`,
and that loop includes micro-brdf sampling.(expensive)
Of course, this should be evaluated stochastically.

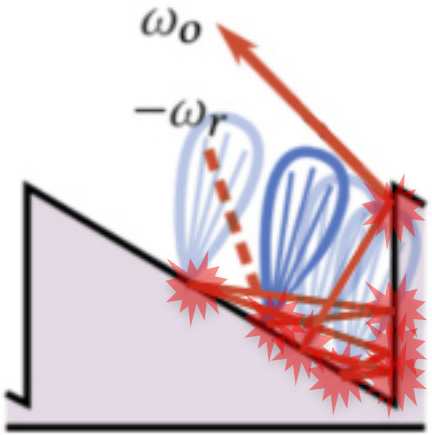
Study 3. Multiple Scattering

KAIST

Motivation:

How do we get the BRDF value at a fixed angles,

i.e. how do we evaluate the value of $f(\omega_i=\textit{fixed}, \omega_o=\textit{fixed})$?



But our model has strong assumptions:

1. **Microgeometry is super simple.**
   → Only two kinds of facets.(V-Groove)

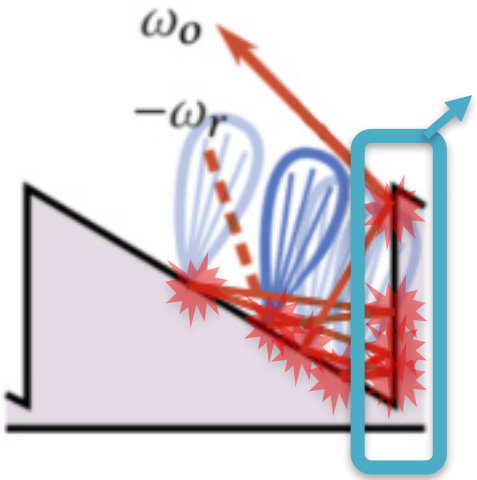
Study 3. Multiple Scattering

KAIST

Motivation:

How do we get the BRDF value at a fixed angles,

i.e. how do we evaluate the value of $f(\omega_i=\textit{fixed}, \omega_o=\textit{fixed})$?



But our model has strong assumptions:

1. **Microgeometry is super simple.**
2. **Tangent facets always do perfect mirror reflection.**
   - → Reflection on tangent facets is deterministic.
   - → Thus the `do-while` loop is just consecutive sampling of the same micro-BRDF.

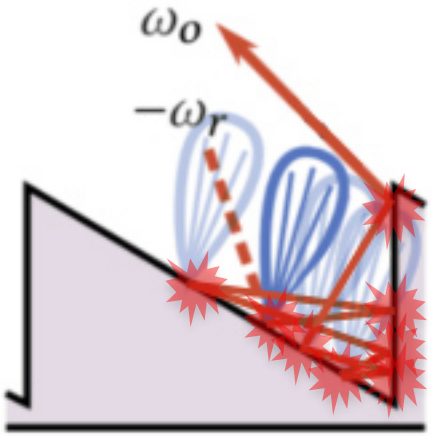
Study 3. Multiple Scattering

KAIST

Motivation:

How do we get the BRDF value at a fixed angles,

i.e. how do we evaluate the value of $f(\omega_i=\textit{fixed}, \omega_o=\textit{fixed})$?



But our model has strong assumptions:

1. **Microgeometry is super simple.**
2. **Tangent facets always do perfect mirror reflection.**
3. **Masking-shadowing function is separable.**

→ They do NOT compute "explicit ray-facet intersection"(e.g. depth) during random walk.

→ Instead, they simply use the average intersection probability. Evidence of simplicity.

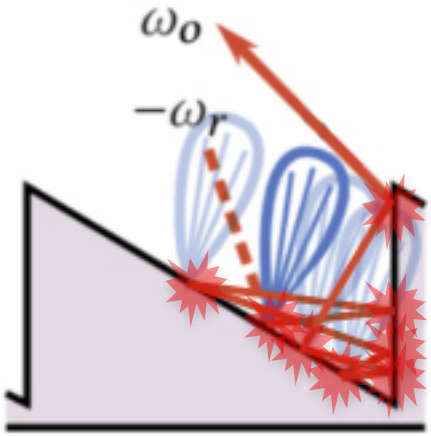
Study 3. Multiple Scattering

KAIST

Motivation:

How do we get the BRDF value at a fixed angles,

i.e. how do we evaluate the value of $f(\omega_i=\textit{fixed}, \omega_o=\textit{fixed})$?



But our model has strong assumptions:

1. **Microgeometry is super simple.**
2. **Tangent facets always do perfect mirror reflection.**
3. **Masking-shadowing function is separable.**

MAYBE, those strong assumptions can give some kind of ***"practical & (more optimistically,) analytic relationship*** *between microBRDF and random-walked macroBRDF"*.

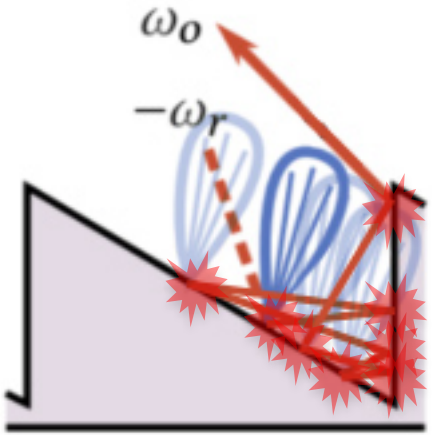
Study 3. Multiple Scattering

KAIST

Motivation:

How do we get the BRDF value at a fixed angles,

i.e. how do we evaluate the value of $f(\omega_i=\textit{fixed}, \omega_o=\textit{fixed})$?



"practical & (more optimistically,) analytic relationship between microBRDF and random-walked macroBRDF".

Devil: "Hey dreamer, it's impossible, no doubts!"

Angel: "But the assumptions may make things easier!"

If it exist, and if we can derive it for this specific problem, we can reduce time for random walk and will be happy ☺

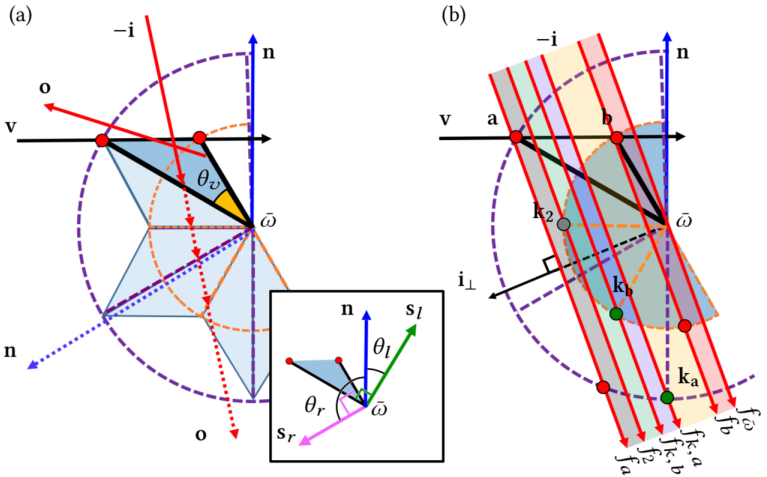
Study 3. Multiple Scattering

Survey:

There was recent SIGGRAPH paper in similar perspective:

<Practical Multiple Scattering for Rough Surfaces>

<Multiple Scattering from Distributions of Specular V-Grooves>



They derived analytic formula to calculate higher-order scattering in the V-groove microfacets.

*(**with enormous maths**)*

(What we exactly dreamed)

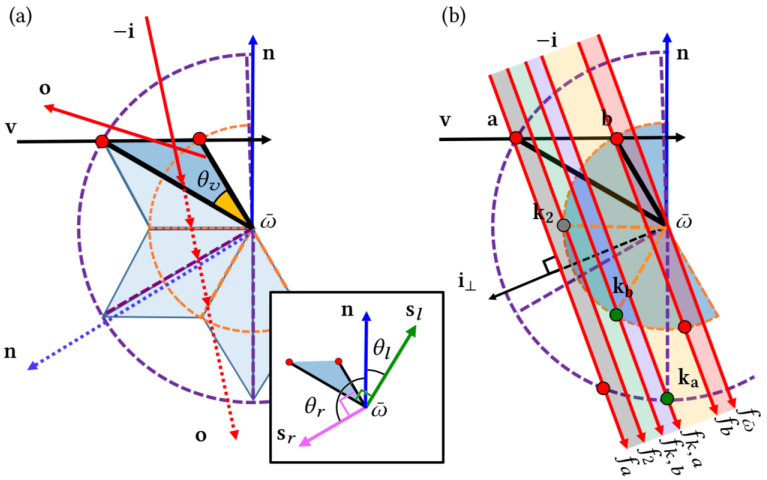
Study 3. Multiple Scattering

Survey:

There was recent SIGGRAPH paper in similar perspective:

<Practical Multiple Scattering for Rough Surfaces>

<Multiple Scattering from Distributions of Specular V-Grooves>



They derived analytic formula to calculate higher-order scattering in the V-groove microfacets.

*(**with enormous maths**)*

But with different assumptions, mainly that **all facets are mirror**.

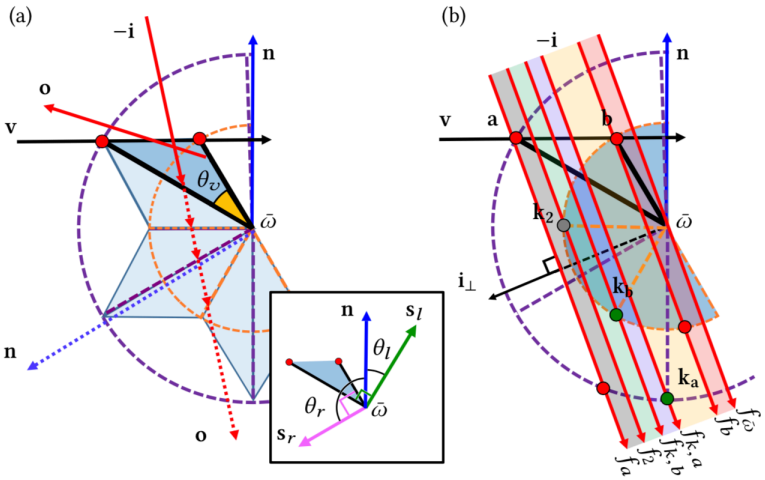
Study 3. Multiple Scattering

Survey:

There was recent SIGGRAPH paper in similar perspective:

<Practical Multiple Scattering for Rough Surfaces>

<Multiple Scattering from Distributions of Specular V-Grooves>



With even stronger assumption (**all facets are mirror**), their derivation is already full of math.

Sadly, we concluded that we are hopeless. However we feel this experience of hypothesizing was valuable to us.

Summary

1. **Support transmittance term**
   - We succeeded to change implementation of the original paper
     .. with massive struggling against Mitsuba framework.

2. **Change microgeometry**
   - There was a reason why the author decided to do so ☺
   - Original microgeometry is the best.

3. **Explore analytic solution for multiple scattering**
   - We dreamed but concluded that there is no hope.

Role

KAIST

Gaspard

- Huge contribution on exploring various theories & ideas
- Experiments on changing micro-geometry

Dahyun

- Manage environments for experiment & co-working
- Study on multiple scattering

Hakyeong

- Main implementation of transmission stuffs
- Support us with a firm theoretical foundation of micro-facets

Implementation is available at https://github.com/313usually/cs580_team4