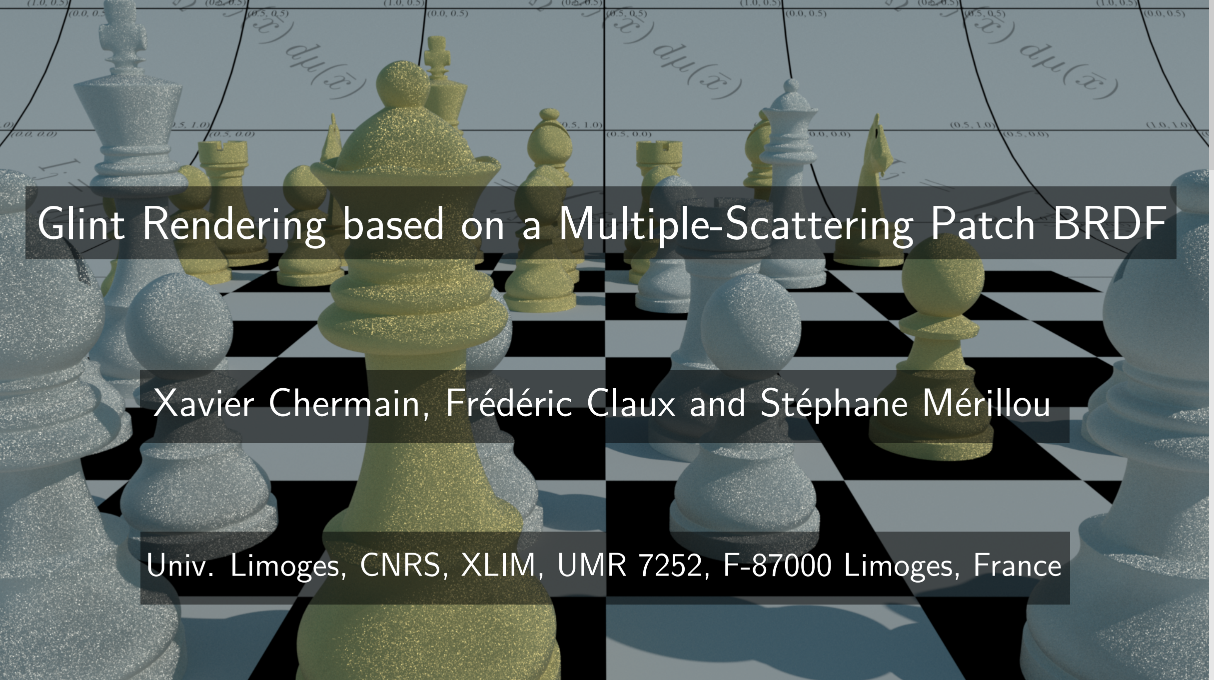
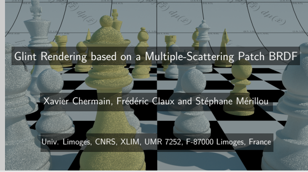




Good afternoon, my name is Xavier Chermain and I will present Glint Rendering based on a Multiple-Scattering Patch BRDF.

Outline

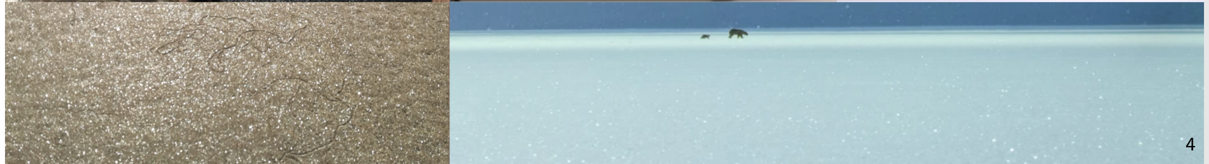
Outline

1 Introduction
2 Related Work
3 Local Multiple-Scattering BRDF for Glint Rendering
4 Multiple-Scattering Patch BRDF for Glint Rendering
5 Results
6 Conclusion and future works

- In the introduction, I will explain what glint rendering is and what it takes to do it right.

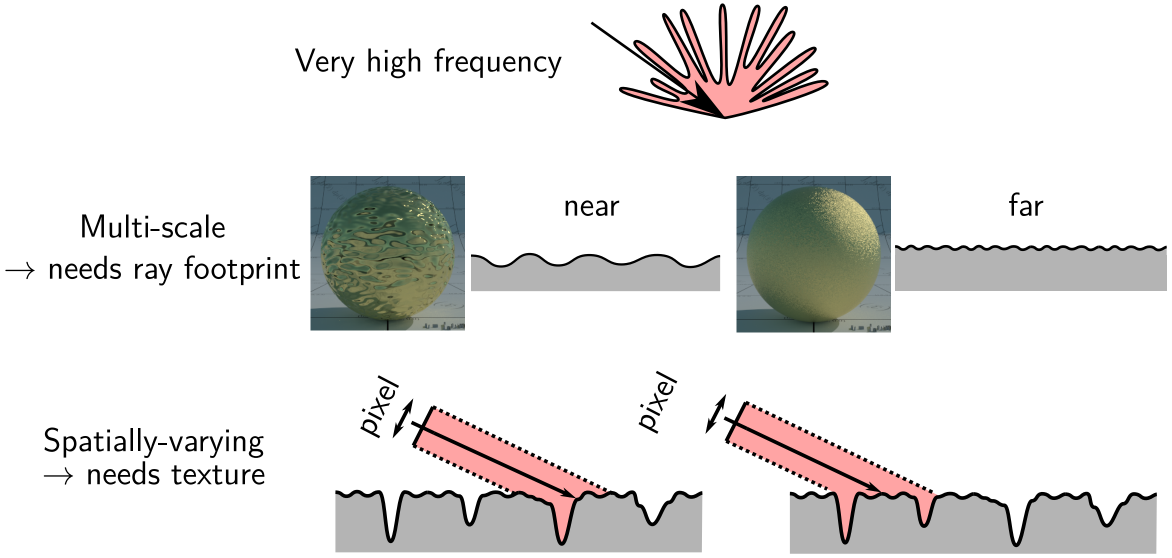
Glint rendering consists in generating images with many sub-pixel micro-mirrors. These specular micro-details are only visible under powerful and sharp lighting, otherwise the surface appears relatively smooth.

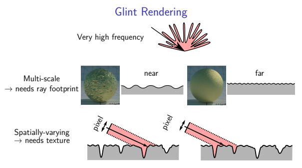
In real life



In real life, we can find many examples of surfaces with micro-mirrors, such as rough plastics, glitter materials, 3D fabrics, the ocean, sand, snow.

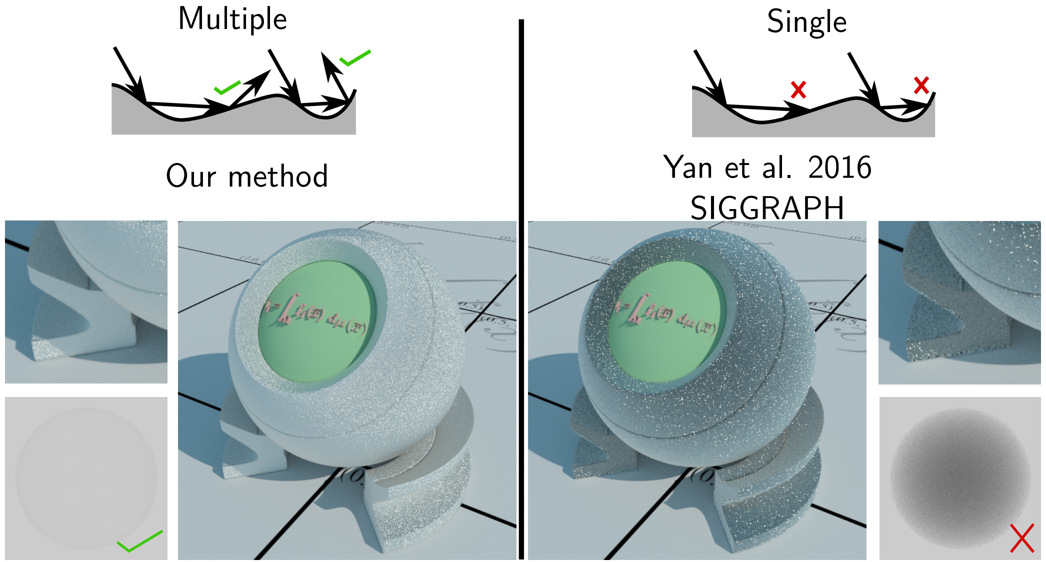
Glint Rendering

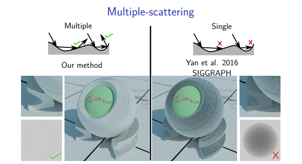




- These materials have BRDFs with hundreds of lobes, making it hard to filter the appearance accurately.
- In addition, the appearance changes according to the zoom level. To handle this, we must use a ray footprint.
- Finally, the BRDF is spatially varying, because the micro-surface is different from one position to another.

Multiple-scattering





To make things worse, modeling single scattering is not sufficient to faithfully replicate the appearance. To check energy conservation, we use white furnace tests. A BRDF passes the white furnace test only when the sphere has the same color as the background. Modeling multiple-scattering is important in order not to have overall black appearance.

We contribute to this field in our work.

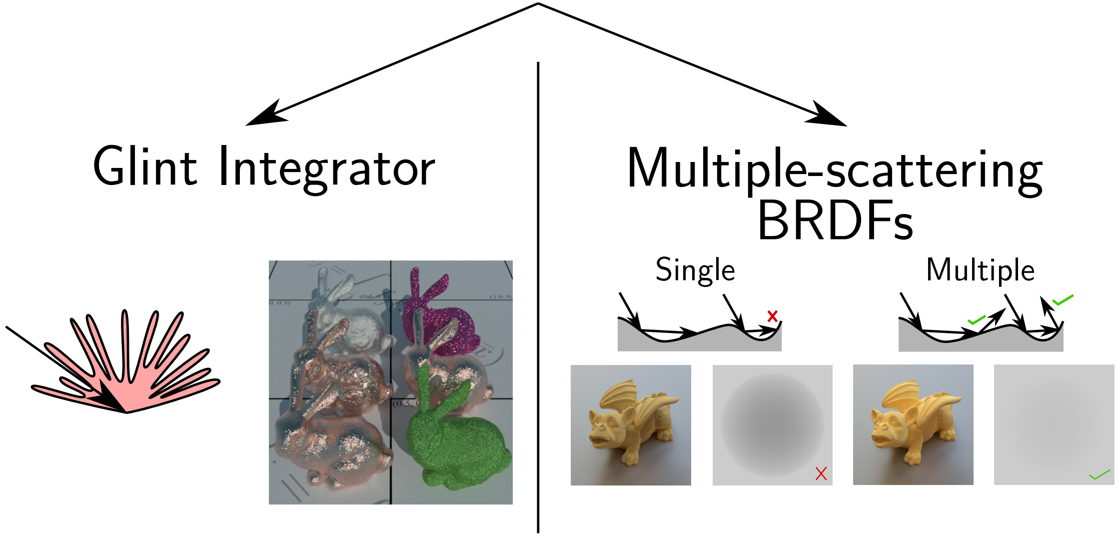
Outline

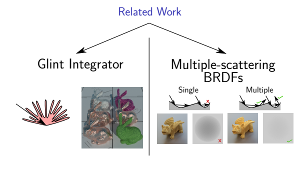
Outline

1 Introduction
2 Related Work
3 Local Multiple-Scattering BRDF for Glint Rendering
4 Multiple-Scattering Patch BRDF for Glint Rendering
5 Results
6 Conclusion and future works

Now, let's review related works.

Related Work





Our method relates to two different areas in computer graphics:

- Glint integrators, to render sub-pixel micro-mirrors.
- And Multiple-Scattering BRDFs, to avoid the overall dark appearance, caused by energy leaks.

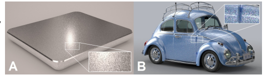
Glint integrators

Single-scattering only

Stochastic surfaces

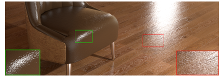
Jakob et al.
SIGGRAPH 2014

Atanasov & Koylazov
SIGGRAPH 2016

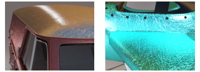


Normal mapped surfaces

Yan et al.
SIGGRAPH 2016



Chermain et al.
Vis. Computer 2018



Gamboa et al.
SIG. ASIA 2018

Multiple-Scattering, but only for scratched surfaces

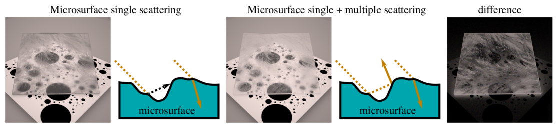
Raymond et al. 2016

Glint integrators determine very quickly which reflections contribute to the lighting. The surface representation is either stochastic or normal mapped.
Almost all methods here have a problem with the BRDF normalization term. Almost no method focuses on multiple-scattering.
Only the method of Raymond and colleagues is correctly normalized. It handles multiple-scattering, but it only deals with scratched surfaces.
In our work we will use normal mapped surfaces, because this representation handles a large range of materials.

Multiple-Scattering BRDFs

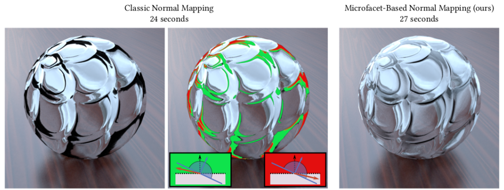
How to *quickly* model multiple-scattering
(we have a lot of sparkles to integrate...)

Heitz et al.
SIGGRAPH 2016



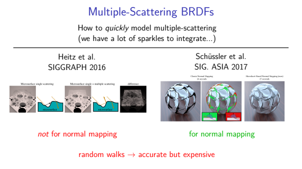
not for normal mapping

Schüssler et al.
SIG. ASIA 2017



for normal mapping

random walks → accurate but expensive



There's been work on multiple-scattering for smooth surfaces and macroscopic normal mapping.

The method of Heitz uses random walks for microfacet-based BSDFs using the Smith model, but it is not specifically designed to handle normal maps. The method of Schüssler is, and we were inspired by their work. However, we do not use the same micro-surface and random walks.

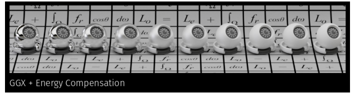
Multiple-Scattering BRDFs

How to *quickly* model multiple-scattering
(we have a lot of sparkles to integrate...)

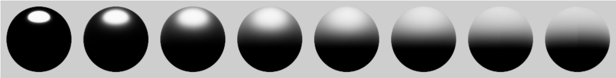
energy compensation schemes

not available for normal mapping

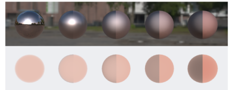
Conty and Kulla
SIGGRAPH Course 2017



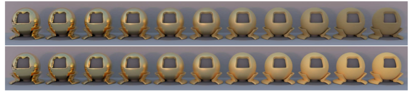
Stephan Hill
Self Shadow blog post (2018)

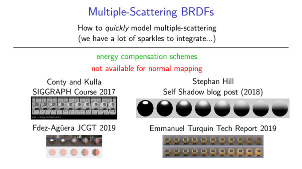


Fdez-Agüera JCGT 2019



Emmanuel Turquin Tech Report 2019





To efficiently model multiple-scattering, severals works directly re-integrate the energy lost by a single-scattering formulation. Some works focus on reciprocity. Turquin focuses on efficient sampling. We also do, but do it in the context of normal mapping for glint rendering.

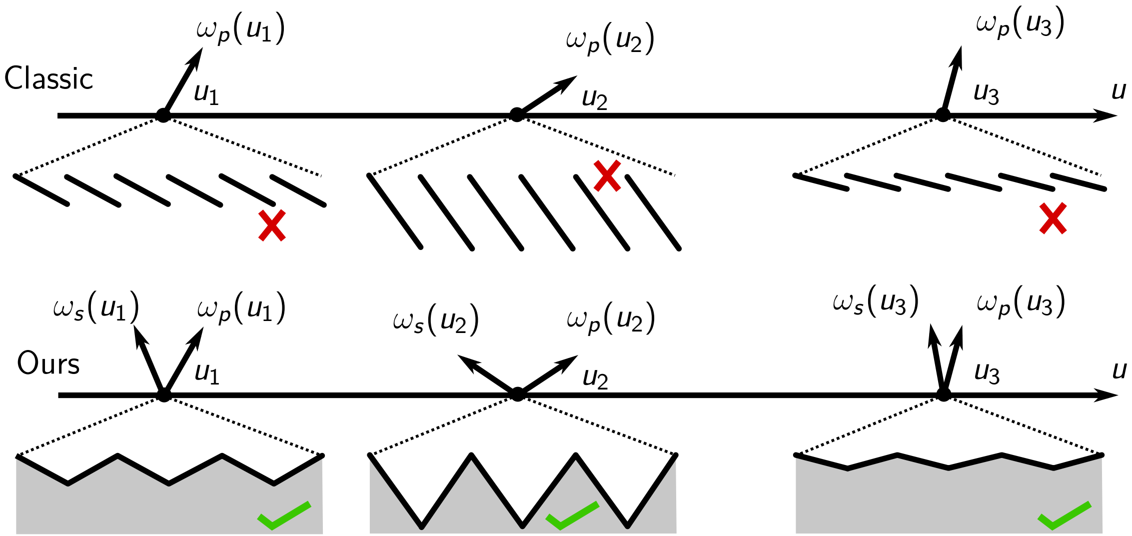
Outline

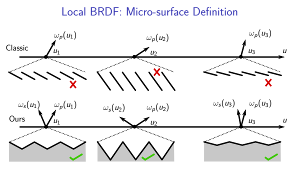
Outline

1 Introduction
2 Related Work
3 Local Multiple-Scattering BRDF for Glint Rendering
4 Multiple-Scattering Patch BRDF for Glint Rendering
5 Results
6 Conclusion and future works

Our model is built around the definition of a local and a patch-wide BRDFs. We start with the formulation of the local BRDF.

Local BRDF: Micro-surface Definition

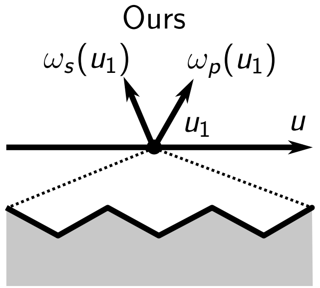




Our micro-surface model is based on a continuous normal map. For each position u over the surface, classic normal mapping defines one normal called perturbed normal. If we zoom on a point on the surface, we will see unrelated and individual discrete facets. With this configuration, we cannot model multiple-scattering.

To solve this issue, we seal the micro-surface using symmetric normals, making it really continuous. Locally, at any position u over the surface, the perturbed normal and its symmetric form V-shaped cavities.

Local BRDF: Overview

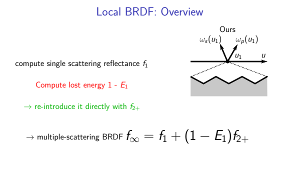


compute single scattering reflectance f_1

Compute lost energy 1 - E_1

$\rightarrow$ re-introduce it directly with f_{2+}

$\rightarrow$ multiple-scattering BRDF $f_\infty = f_1 + (1 - E_1) f_{2+}$



Now that we have defined the micro-surface, let's compute its BRDF.
First step: we compute the single scattering reflectance of a V-cavity.
Second step: we compute the percentage of lost energy.

Last step: we re-introduce the lost energy, in combination with an energy compensation BRDF, that models the second and more scattering events. Its choice is arbitrary, as long as it conserves energy.

Local BRDF: Single-Scattering Formulation f_1

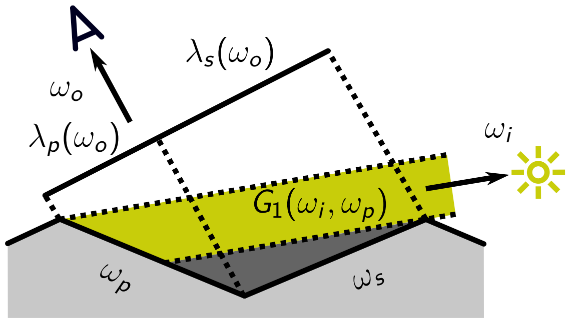
micro-BRDF f_m: specular for glint rendering

Normalize it with:

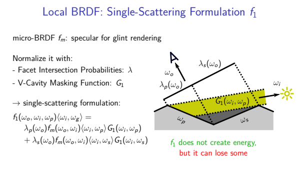
- Facet Intersection Probabilities: λ
- V-Cavity Masking Function: G_1

$\rightarrow$ single-scattering formulation:

$$f_1(\omega_o, \omega_i, \omega_p)\langle\omega_i, \omega_g\rangle = \lambda_p(\omega_o) f_m(\omega_o, \omega_i)\langle\omega_i, \omega_p\rangle G_1(\omega_i, \omega_p) + \lambda_s(\omega_o) f_m(\omega_o, \omega_i)\langle\omega_i, \omega_s\rangle G_1(\omega_i, \omega_s)$$



f_1 does not create energy,
but it can lose some



Let's derive our single-scattering BRDF for one near-perfectly specular V-cavity.
To correctly normalize the micro-BRDF, we use facet intersection probabilities and the V-Cavity masking function. Our single-scattering formulation is just a weighted sum of 2 micro-BRDFs. The derivation is similar to the one of Schüssler. f_1 does not create energy, but it can lose some.

Local BRDF: Lost Energy $1 - E_1$ and Multiple-Scattering f_∞

Energy Term E_1

$$E_1(\omega_o, \omega_p) = \int_\Omega f_1(\omega_o, \omega_i, \omega_p)\langle \omega_i, \omega_g \rangle \mathrm{d}\omega_i$$

Analytical solution if f_m is perfectly specular

Glints are microscopic mirrors ✔

Multiple-scattering f_∞:

$$f_\infty(\omega_o, \omega_i, \omega_p) = f_1(\omega_o, \omega_i, \omega_p) + (1 - E_1(\omega_o, \omega_p))f_{2+}(\omega_o, \omega_i)$$

- efficient to integrate all micro-mirrors

- 100% energy conservation

Local BRDF: Lost Energy $1 - E_1$ and Multiple-Scattering f_∞

Energy Term E_1

$E_1(\omega_o, \omega_p) = \int_\Omega f_1(\omega_o, \omega_i, \omega_p)\langle \omega_i, \omega_g \rangle \mathrm{d}\omega_i$

Analytical solution if f_m is perfectly specular

Glints are microscopic mirrors ✔

Multiple-scattering f_∞:

$f_\infty(\omega_o, \omega_i, \omega_p) = f_1(\omega_o, \omega_i, \omega_p) + (1 - E_1(\omega_o, \omega_p))f_{2+}(\omega_o, \omega_i)$

- efficient to integrate all micro-mirrors

- 100% energy conservation

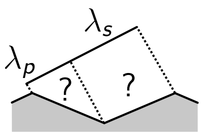
To handle multiple-scattering, we use the energy term which gives the single-scattering reflection. It implies an integral for which we have an analytic solution if the micro-BRDF is perfectly specular, which it is, as our glints are microscopic mirrors.
Now, we have our multiple-scattering BRDF which is efficient enought to integrate all micro mirrors. It conserves 100 % of the incomming energy.

Note that our multiple-scattering formulation is non-symmetric because of the energy term.

It is only a deal breaker when using bidirectionnal path tracing.

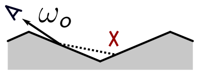
Local BRDF: Importance Sampling $f_\infty = f_1 + (1 - E_1) f_{2+}$

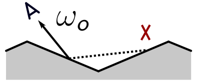
simple procedure, weights ≤ 1, no lost sample

Pick a facet 

then pick ω_i 

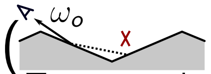
Transmitted or Shadowed

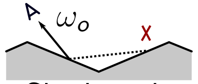
If 



Then Sample f_{2+}

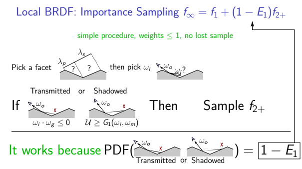
$\omega_i \cdot \omega_g \leq 0$ $\quad$ $\mathcal{U} \geq G_1(\omega_i, \omega_m)$

It works because PDF(



) = $\boxed{1 - E_1}$

Transmitted or Shadowed



Importance sampling is relatively simple with our multiple-scattering formulation. Our algorithm gives weights less than or equal to 1, and no samples are lost.
First, a facet is chosen. Then, the micro-BRDF is sampled.
If the sampling is successful, we keep this sample.
Otherwise, if the picked direction is transmitted or shadowed, we sample f_{2+}.
It works because the PDF of having an invalid sample from f_1 is equal to the percentage of lost energy.

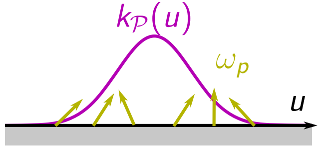
Outline

Outline

1 Introduction
2 Related Work
3 Local Multiple-Scattering BRDF for Glint Rendering
4 Multiple-Scattering Patch BRDF for Glint Rendering
5 Results
6 Conclusion and future works

Now, I'm going to talk about the ray footprint or patch BRDF.

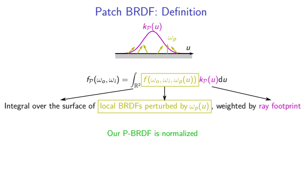
Patch BRDF: Definition



$$f_{\mathcal{P}}(\omega_o, \omega_i) = \int_{\mathbb{R}^2} \boxed{f(\omega_o, \omega_i, \omega_p(u))}\, k_{\mathcal{P}}(u)\,\mathrm{d}u$$

Integral over the surface of $\boxed{\text{local BRDFs perturbed by } \omega_p(u)}$, weighted by ray footprint

Our P-BRDF is normalized



To formally define a patch BRDF, we have to introduce the ray footprint, right here in purple.
It is just a normalized low-pass filter, which in our case is a Gaussian.
For us, a P-BRDF is just an integral over the surface of Gaussian weighted local BRDFs.

Our P-BRDF is normalized.

Patch BRDF: Evaluation

$$f_{\mathcal{P}}(\omega_o, \omega_i) = \int_{\mathbb{R}^2} f_\infty(\omega_o, \omega_i, \omega_p(u)) k_{\mathcal{P}}(u) \mathrm{d}u$$

$$= \boxed{f_1(\omega_o, \omega_i, \mathcal{P})} + \boxed{E_{2+}(\omega_o, \mathcal{P})} f_{2+}(\omega_o, \omega_i)$$

Single-Scattering Patch BRDF

Energy Lost in the Patch

Patch BRDF: Evaluation

$$f_{\mathcal{P}}(\omega_o, \omega_i) = \int_{\mathbb{R}^2} f_\infty(\omega_o, \omega_i, \omega_p(u)) k_{\mathcal{P}}(u) \mathrm{d}u = \boxed{f_1(\omega_o, \omega_i, \mathcal{P})} + \boxed{E_{2+}(\omega_o, \mathcal{P})} f_{2+}(\omega_o, \omega_i)$$

Single-Scattering Patch BRDF

Energy Lost in the Patch

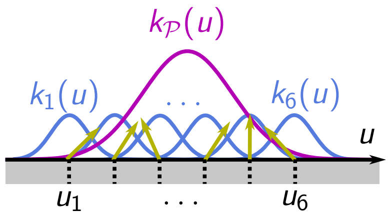
By substituting our multiple-scattering local BRDF into the previous equation, we can isolate two terms that depend on the patch: the single-scattering Patch BRDF and the energy lost in the Patch.

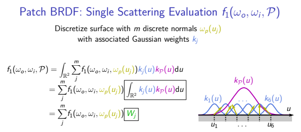
Let me explain now how to efficiently evaluate these terms.

Patch BRDF: Single Scattering Evaluation $f_1(\omega_o, \omega_i, \mathcal{P})$

Discretize surface with m discrete normals $\omega_p(u_j)$
with associated Gaussian weights k_j

$$
\begin{aligned}
f_1(\omega_o, \omega_i, \mathcal{P}) &= \int_{\mathbb{R}^2} \sum_j^m f_1(\omega_o, \omega_i, \omega_p(u_j)) k_j(u) k_{\mathcal{P}}(u) \mathrm{d}u \\
&= \sum_j^m f_1(\omega_o, \omega_i, \omega_p(u_j)) \boxed{\int_{\mathbb{R}^2} k_j(u) k_{\mathcal{P}}(u) \mathrm{d}u} \\
&= \sum_j^m f_1(\omega_o, \omega_i, \omega_p(u_j)) \boxed{W_j}
\end{aligned}
$$





Previous equations assume an infinite set of normals. We discretize the surface with discrete normals. They have an associated Gaussian weight which gives us a closed form to weight the discrete local BRDFs.

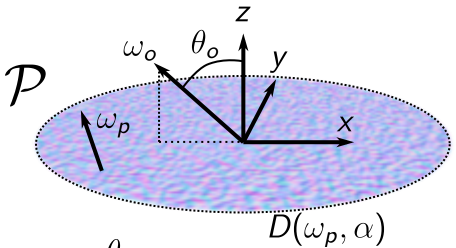
Normals contributing to the BRDF are quickly isolated using a BVH query, much like in previous works.

Patch BRDF: Energy Lost in the Patch $E_{2+}(\omega_o, \mathcal{P})$

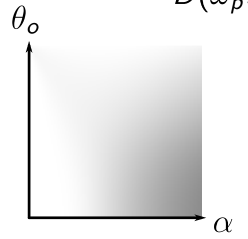
too many discrete normals ω_p

Assume Gaussian Distribution
$D(\omega_p, \alpha_x(\mathcal{P}), \alpha_y(\mathcal{P}))$ in ray footprint

LEADR gives them
Dupuy et al. 2013

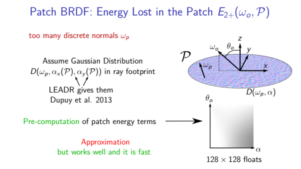


Pre-computation of patch energy terms



128×128 floats

Approximation
but works well and it is fast



Now, we have to know the percentage of energy lost in the ray footprint. To know it, we have to go through all the discrete normals, but they are too many.
To improve performance, we assume a Gaussian distribution of micro-normals in the ray footprint. The LEADR method gives us rapidly the roughness in the patch.
We numerically pre-compute patch energy terms by varying the roughness in the patch and the polar angle of the observation direction.

It's an approximation, but it works well and it is fast.

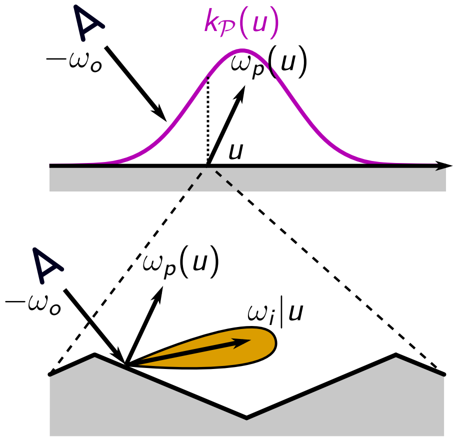
Patch BRDF: Importance Sampling

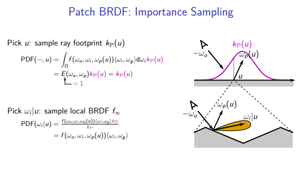
Pick u: sample ray footprint $k_{\mathcal{P}}(u)$

$$\begin{aligned}\mathrm{PDF}(-,u) &= \int_\Omega f(\omega_o,\omega_i,\omega_p(u))\langle\omega_i,\omega_g\rangle \mathrm{d}\omega_i\, k_{\mathcal{P}}(u) \\ &= \underbrace{E(\omega_o,\omega_p)}_{=1} k_{\mathcal{P}}(u) = k_{\mathcal{P}}(u)\end{aligned}$$

Pick $\omega_i|u$: sample local BRDF f_∞

$$\begin{aligned}\mathrm{PDF}(\omega_i|u) &= \frac{f(\omega_o,\omega_i,\omega_p(u))\langle\omega_i,\omega_g\rangle k_{\mathcal{P}}}{k_{\mathcal{P}}} \\ &= f(\omega_o,\omega_i,\omega_p(u))\langle\omega_i,\omega_g\rangle\end{aligned}$$





Importance sampling is used to efficiently solve the patch rendering equation.
We first sample a position on the surface, then a direction.
The first step implies a marginalized PDF, which in our case simplifies to the ray footprint.
The second step uses the local BRDF sampling procedure described earlier. This algorithm is only valid thanks to our multiple-scattering formulation.

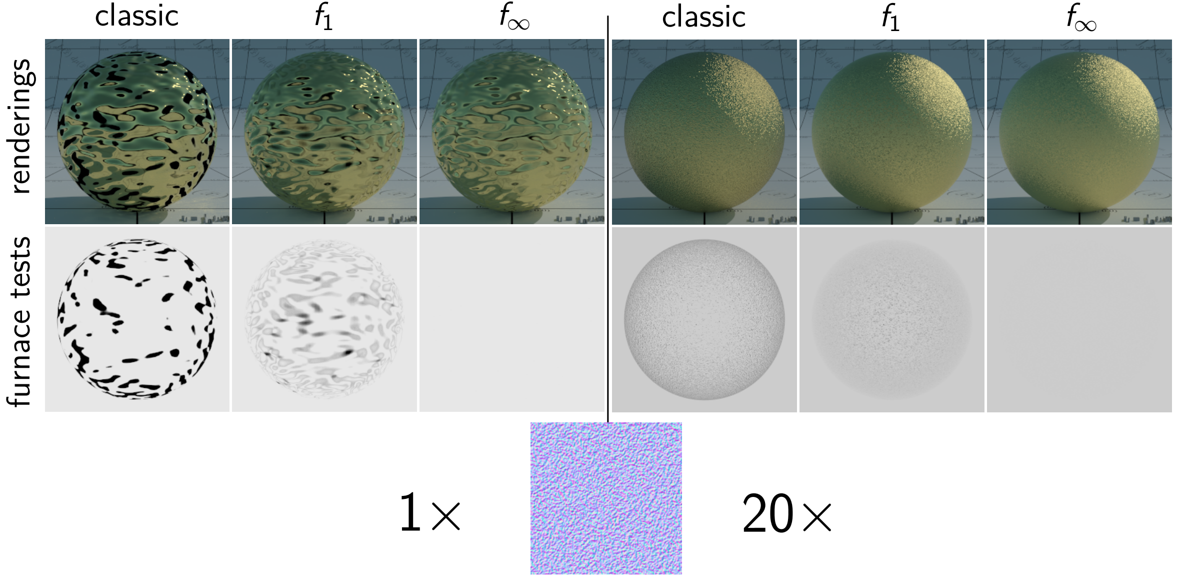
Outline

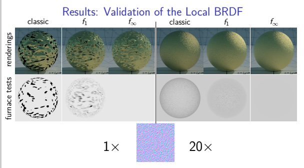
Outline

1 Introduction
2 Related Work
3 Local Multiple-Scattering BRDF for Glint Rendering
4 Multiple-Scattering Patch BRDF for Glint Rendering
5 Results
6 Conclusion and future works

Now, we will see the results.

Results: Validation of the Local BRDF

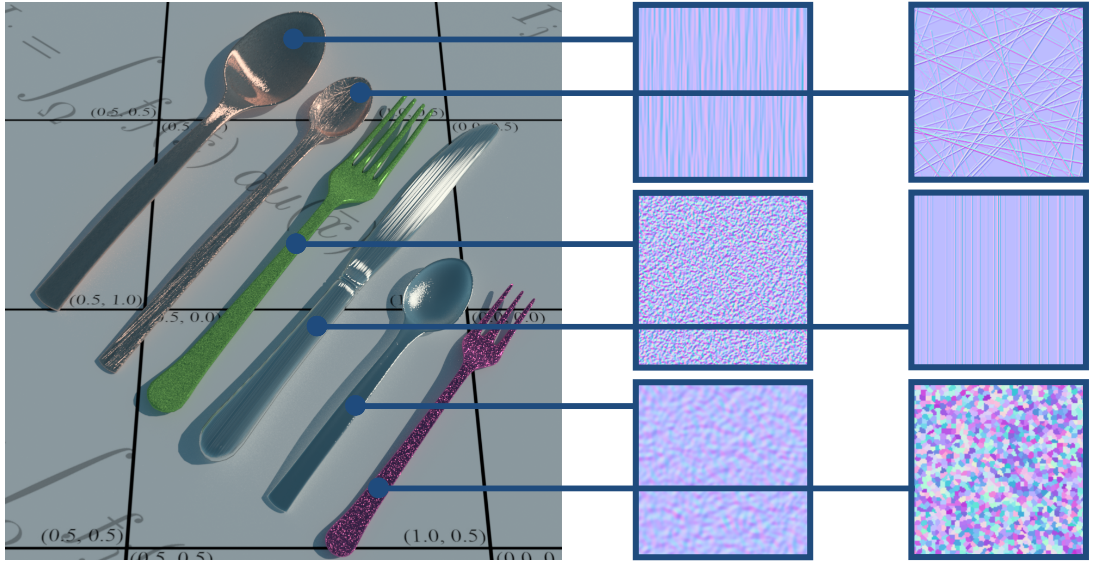




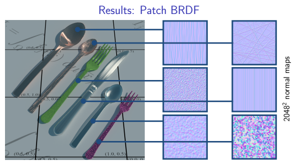
First, we validate our multiple-scattering local BRDF for different level of details.
Classic normal mapping has black fringes whearas our multiple-scattering model passes the white furnace test.

Note that when the texture repetition is twentyfold, classic normal mapping gives an overall dark appareance.

Results: Patch BRDF



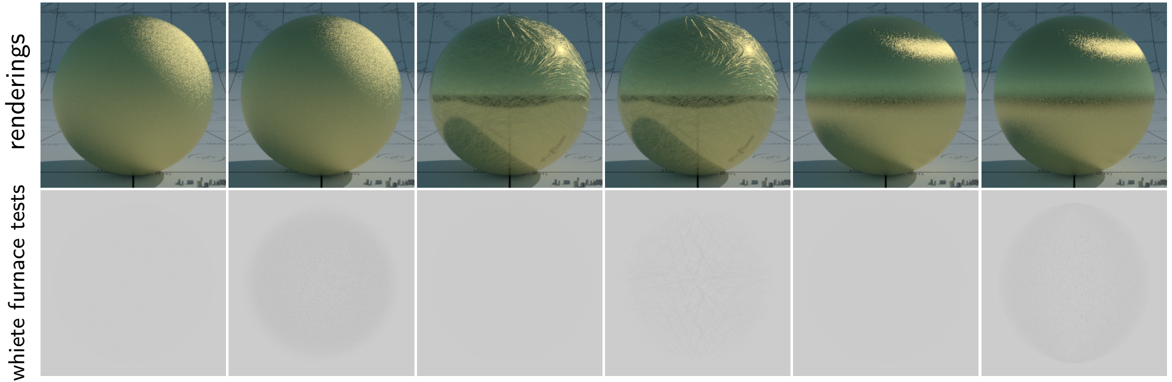
2048^2 normal maps



Now, you can see results using our Patch-BRDF, based on ray footprints. Here, each piece of cutlery has a different micro-surface, modeled by a normal map. We use 2048^2 normal maps.

Results: Impact of the Patch-Energy Approximation

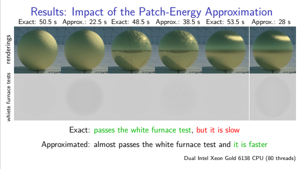
Exact: 50.5 s | Approx.: 22.5 s | Exact: 48.5 s | Approx.: 38.5 s | Exact: 53.5 s | Approx.: 28 s



Exact: passes the white furnace test, but it is slow

Approximated: almost passes the white furnace test and it is faster

Dual Intel Xeon Gold 6138 CPU (80 threads)

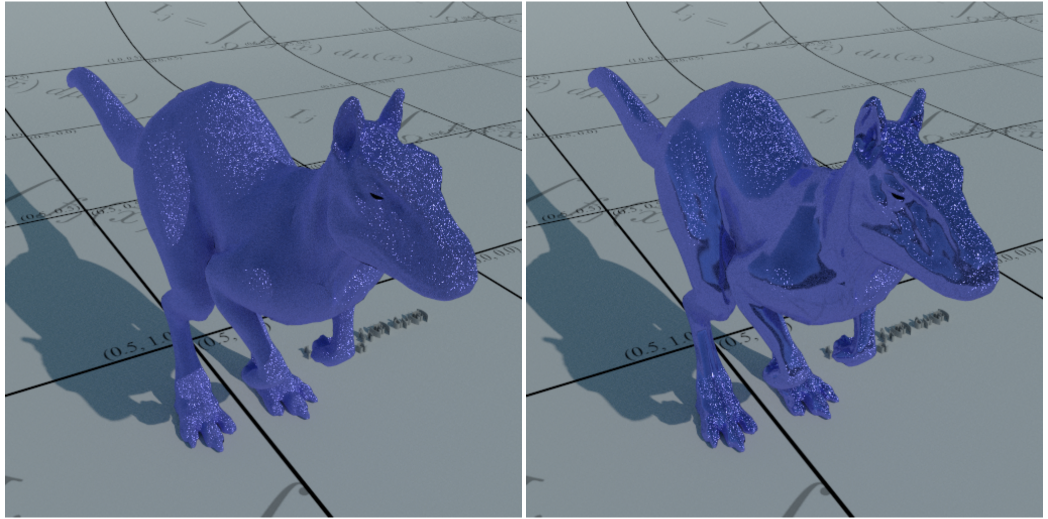


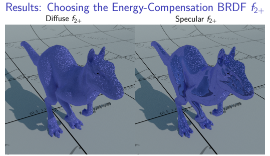
As I said before, we use an approximation to quickly determine the energy lost in the patch.
Here we test this approximation and we also validate our exact Patch-BRDF.
The exact formulation passes the white furnace test, but it is slow.

Our approximated model almost passes the white furnace test and it is faster. We use it in all our results.

Results: Choosing the Energy-Compensation BRDF f_{2+}

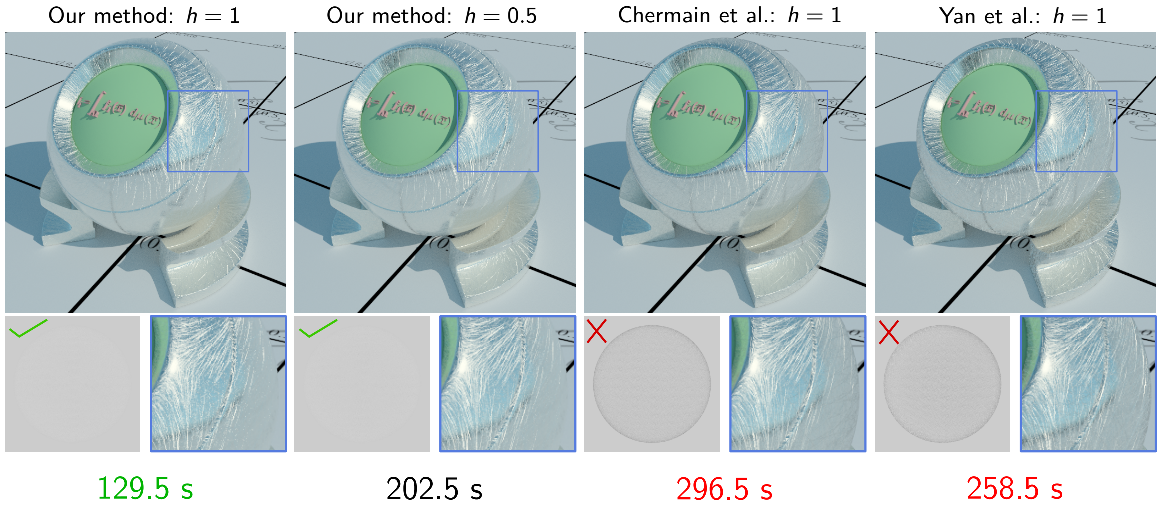
Diffuse f_{2+} Specular f_{2+}

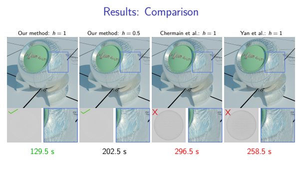




We tested our model with two different energy compensation BRDFs. On the left side, a diffuse f_{2+}, leading to a rough appearance. On the right side, a near zero roughness BRDF, leading to a coated surface, such as a mettalic paint.

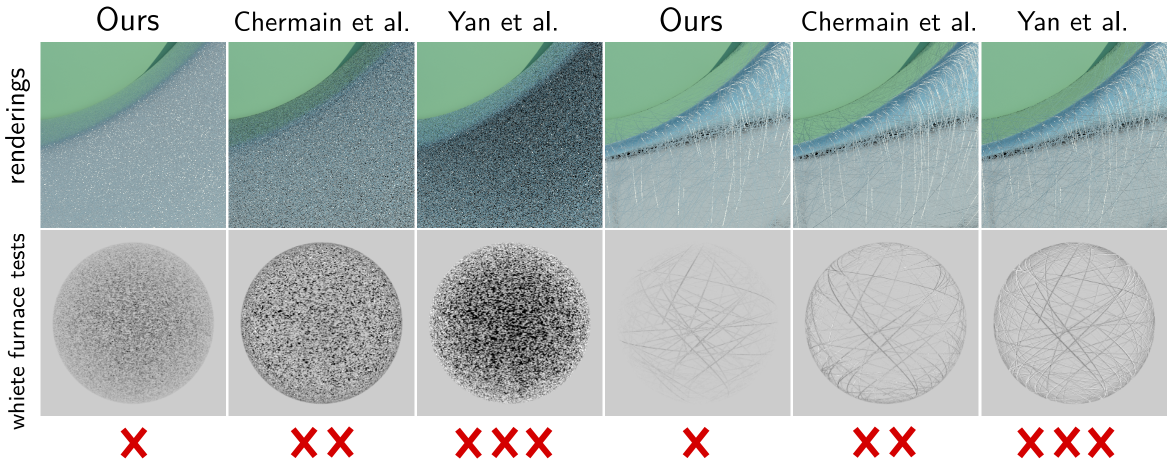
Results: Comparison





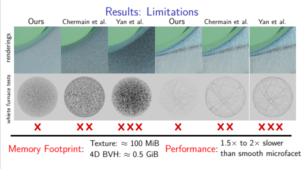
Previous methods try to reconstruct a continuous surface using NDFs with roughness parameters derived from local surface curvature. Performance is lower with this approach as high roughness badly affects the performance. To avoid this, we only use tiny roughness. We prefer to reduce the discretization step to better fit surface curvature.

Results: Limitations



Memory Footprint: Texture: $\approx$ 100 MiB
4D BVH: $\approx$ 0.5 GiB

Performance: $1.5\times$ to $2\times$ slower than smooth microfacet



Our method assumes a Gaussian distribution of normals within the ray footprint for the patch energy term. At high zoom levels, the Gaussian approximation no longer holds and we have energy leaks. But compared to previous methods, we still are better.

Our method also relies on textures and accelaration structures, leading to an important memory footprint. Also, compared to a smooth BRDF, we still have an important rendering time overhead, varying between 50 % and 100%.

Outline

Outline
1 Introduction
2 Related Work
3 Local Multiple-Scattering BRDF for Glint Rendering
4 Multiple-Scattering Patch BRDF for Glint Rendering
5 Results
6 Conclusion and future works

Now, I will summarize our work and discuss future works.

Conclusion

- First multiple-scattering glint integrator.
- No more BRDF sample wasted and fireflies.

Conclusion

- First multiple-scattering glint integrator.
- No more BRDF sample wasted and fireflies.

We have proposed the first multiple-scattering glint integrator based on a normal map. Compared to previous methods, we no longer waste samples.

Future works

- Dielectrics.
- No more textures and BVH. Only procedural with several parameters.

Future works

- Dielectrics.
- No more textures and BVH. Only procedural with several parameters.

We still have things to do.
Glints from dielectric materials have never been modelled and there's probably something to do here.

Ideally, we would like to get rid of the texture and BVH dependencies. Our normal maps are generated procedurally. Perhaps there is a way to directly use these random processes to compute the reflectance.

If you are interested in our method, we have published the source code.
Thank you for your attention. Do you have any questions?