
unity
Physically-Based Material, where are we?
Sébastien Lagarde
30 JULY – 3 AUGUST Los Angeles
SIGGRAPH2017
Open problems in Real-Time Rendering Siggraph 2017

Hello,

Introduction

Introduction

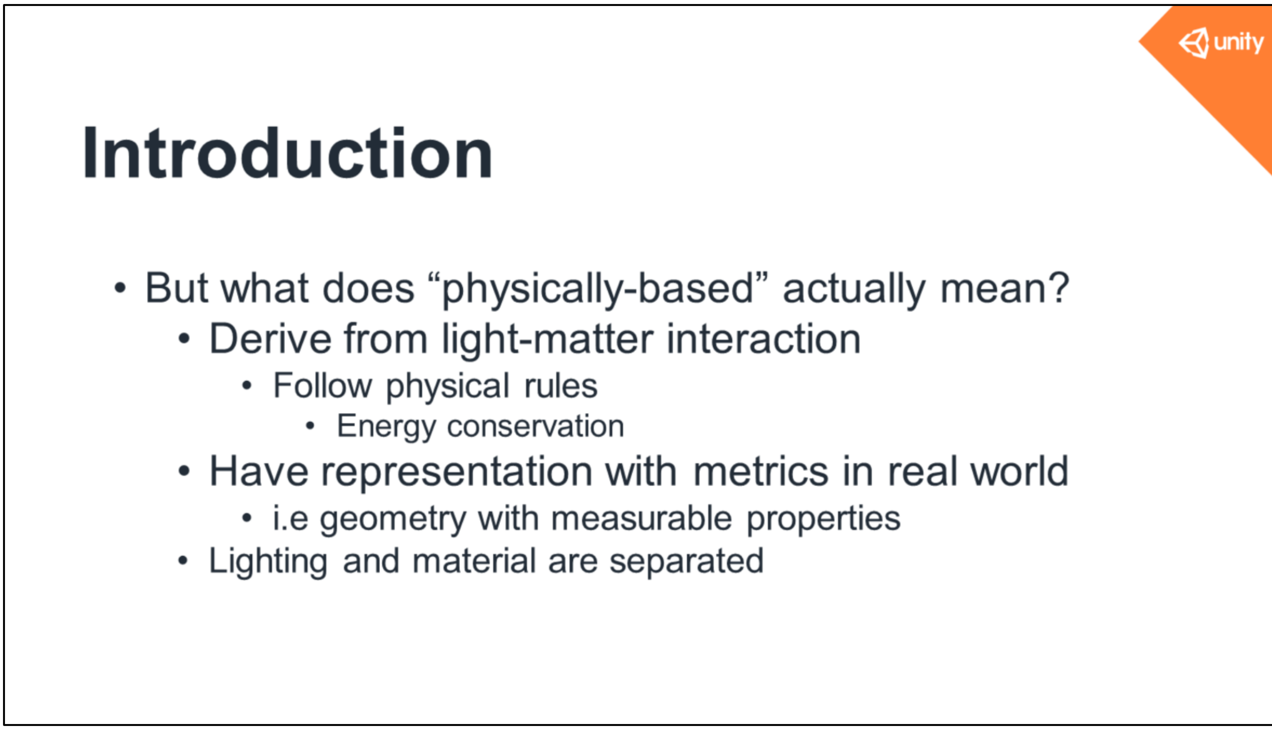
- Nowadays everything is PBR
  - PBR engines
  - PBR materials
  - PBR textures
  - PBR lighting
  - PBR beers

Pabst blue ribbon (PBR) beer

Introduction

- Everything is PBR nowaday
  - PBR engines
  - PBR materials
  - PBR textures
  - PBR lighting
  - PBR beers
- What is a physically-based material?

Within this PBR zoo, let's try to understand what a physically based material actually means



Rendering engineers tend to talk about physically *inspired* materials rather than physically *based*

Introduction

- Everything is PBR nowaday
  - PBR engines
  - PBR materials
  - PBR textures
  - PBR lighting
  - PBR beers
- What is a physically based texture?

Introduction

- No definition, but will make one for this talk
- Physically-based textures are inputs for the physically-based material
- Physically-based textures can contain measured (e.g. scanned) real-world data

Introduction

- Today's talk
  - Definition of a physically based material
    - Restrict to bidirectional reflectance distribution function (BRDF)
      - Diffuse / Specular
    - No time to cover transparency, volume, non-local subsurface
  - Current and future real-time material model
  - Current physically based texture workflow

Rendering of transparency is covered by [McGuire16], M. McGuire, Peering Through a Glass, Darkly at the Future of Real-Time Transparency, Siggraph 2016

Physically based material?

unity

Physically based material?

- How do we know we are physically based?
  - Compare to the real world?

Physically based material?

- How do we know we are physically based?
  - Compare to the real world?
    - Real world <u>IS</u> the reference
    - But not practical
      - No parameter control
      - Many sources of noise
      - Hard to make a comparison

Of course, the real world is right image, the purpose of this comparison is to show that the real world is hard to compare to: different lighting conditions, multiple material layers, layout, point of view, camera response curve, etc..

Physically based material?

- Compare to controlled real world?
  - Measurements provide good reference
  - But be aware of device limitation
    - Often not discussed in publication
    - Example: Incorrect acquisition near grazing angles / backscattering
      - [MERL06]
    - Still good for highlight shape

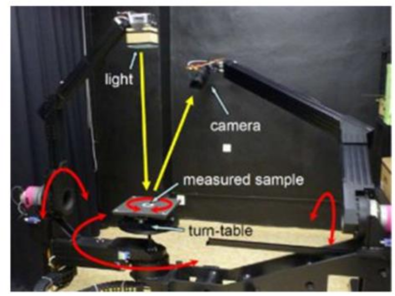


Image from "BRDF Slices: Accurate Adaptive Anisotropic Appearance Acquisition", J. Filip 2013

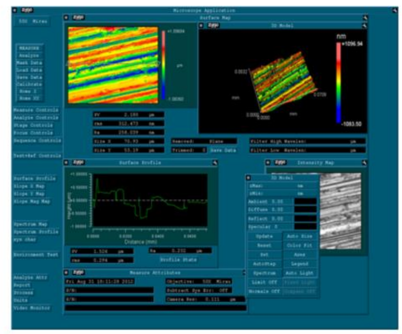
There is plenty of measurement devices available with more or less complex apparatus.
Some can be really accurate like the new reflectometer of Wenzel Jakob but most suffer from limitations of optics.

Issues with BRDF measurement:
Light sources: angular size, brightness, stability, speckle
Detector: Angular size, sensitivity, noise, resolution
positioning: accuracy, drift, hysteresis

For example the MERL database apparatus does not correctly capture the reflectance at grazing angles.
[MERL06] https://www.merl.com/brdf/
Still measurements like MERL are useful to compare highlight shape

Physically based material?

- Measurements are not just about RGB reflectance
  - Complex devices provide advanced information
    - Spectral measurement
    - Microsurface measurement
  - Needs more widespread adoption



Having Microsurface measurement could help for several material validation

Physically based material?

- But BRDF fit to measurements != Physically-based
  - Fitting process is similar to compression
  - Fitted values can be unintuitive
  - BRDF can rely on non physical assumption
    - Shifted Gamma Microfacet [Bagher12] add an inverse Fresnel term to get better fitting

[Bagher12] M. Bagher, “Accurate fitting of measured reflectances using a Shifted Gamma micro-facet distribution”, 2012

Something important to note is that the fitting process is pure mathematics. It can be see as compressing the data.
When fitting a brdf, you can get a good fit but with very unintuitive parameters.
Also the BRDF can be created with non physical assumption.
For example the (SGD) Shifted Gamma Microfacet distribution have added a negative Fresnel term to match MERL database.
https://hal.inria.fr/hal-00702304/file/paper.pdf

Physically based material?

- Simulated data?
  - Based on physical/optics rules
  - Free of noise
  - Parameters controls
  - Easier to compare with
  - Can isolate terms
    - Single bounce / multiple bounce
    - ...

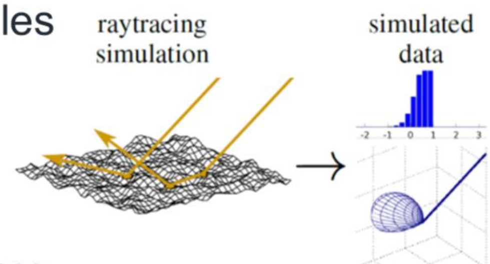


Image from "Multiple-Scattering Microfacet BSDFs with the Smith Model, E. Heitz Siggraph 2016

Physically based material?

- How do we know we are physically based?
  - The key is to combine measurements and simulation
    - Validate simulated data using measurements
    - Measure microsurface data and do simulation based it
  - Based on such approach researchers have established various theories to represent material
    - Let's explore one of them: Microfacet theory

Microfacet BRDF

- Start with this BRDF equation

$$\frac{F(\omega_o, \omega_h)\, G_2(\omega_o, \omega_i, \omega_h)\, D(\omega_h)}{4\,|\omega_g \cdot \omega_o|\,|\omega_g \cdot \omega_i|}$$

Cook-Torrance BRDF

Microfacet BRDF

- Bigger one please!

$$\rho(\omega_o, \omega_i) = \frac{1}{|\omega_g \cdot \omega_o| \, |\omega_g \cdot \omega_i|} \int_\Omega \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_2(\omega_o, \omega_i, \omega_m) D(\omega_m) \, d\omega_m$$

Microfacet BRDF

- Not a math talk but want to explain certain concepts
- [Heitz14] provides a framework for microfacet BRDFs
  - A set of rules to follow to be physically based
- Below is the general microfacet equation

$$\rho(\omega_o, \omega_i) = \frac{1}{|\omega_g \cdot \omega_o| \, |\omega_g \cdot \omega_i|} \int_\Omega \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_2(\omega_o, \omega_i, \omega_m) D(\omega_m) \, d\omega_m$$

[Heitz14] E. Heitz, Understanding the Masking-Shadowing Function in Microfacet-Based BRDFs, JCGT

Microfacet BRDF

- Pm: Facet BRDF
  - Pure Lambertian or pure specular
- D: Normal distribution function (NDF)
- G2: Shadowing-Masking term derived from D
- This defines a BRDF

$$\rho(\omega_o,\omega_i) = \frac{1}{|\omega_g \cdot \omega_o|\,|\omega_g \cdot \omega_i|} \int_\Omega \rho_{\mathcal{M}}(\omega_o,\omega_i,\omega_m)\,\langle\omega_o,\omega_m\rangle\,\langle\omega_i,\omega_m\rangle\, G_2(\omega_o,\omega_i,\omega_m)\, D(\omega_m)\, d\omega_m$$

We now construct the BRDF upon the distribution of visible normals.
The radiance L(ωo,ωm) of each microfacet can be expressed in terms of the facet BRDF ρM (ωo,ωi ,ωm) associated with each microfacet and integrated with the incident radiance L(ωi) over the domain of the incident directions Ωi (we reserve Ω for the space of the normals):

Facet BRDF

- Pm: Facet BRDF - pure specular

$$\rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) = \left\| \frac{\partial \omega_h}{\partial \omega_i} \right\| \frac{F(\omega_o, \omega_h)\, \delta_{\omega_h}(\omega_m)}{|\omega_i \cdot \omega_h|}$$

$$\rho(\omega_o, \omega_i) = \frac{1}{|\omega_g \cdot \omega_o|\,|\omega_g \cdot \omega_i|} \int_\Omega \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_2(\omega_o, \omega_i, \omega_m) D(\omega_m)\, d\omega_m$$

$$= \frac{F(\omega_o, \omega_h)\, G_2(\omega_o, \omega_i, \omega_h)\, D(\omega_h)}{4\,|\omega_g \cdot \omega_o|\,|\omega_g \cdot \omega_i|}$$

With a pure specular we get back familiar Cook-Torrance BRDF

Facet BRDF

- Pm: Facet BRDF - pure Lambertian

$$\rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) = \frac{1}{\pi}$$

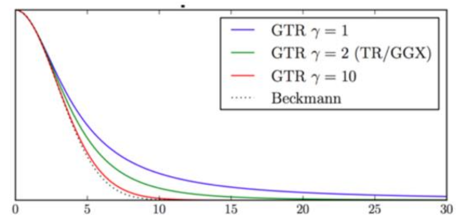
$$\rho(\omega_o, \omega_i) = \frac{1}{|\omega_g \cdot \omega_o|\,|\omega_g \cdot \omega_i|} \int_\Omega \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m)\,\langle \omega_o, \omega_m \rangle \,\langle \omega_i, \omega_m \rangle \, G_2(\omega_o, \omega_i, \omega_m)\, D(\omega_m)\, d\omega_m$$

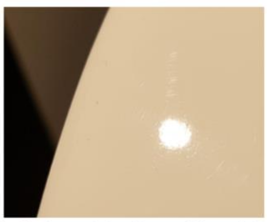
$$= \frac{1}{\pi} \frac{1}{|\omega_g \cdot \omega_o|\,|\omega_g \cdot \omega_i|} \int_\Omega \langle \omega_o, \omega_m \rangle \,\langle \omega_i, \omega_m \rangle \, G_2(\omega_o, \omega_i, \omega_m)\, D(\omega_m)\, d\omega_m$$

With pure Lambertian, we still have complex integral… This is a physically based diffuse term.

D: NDF

- Isotropic or anisotropic
- Like a curve driving the highlight shape





Images from “Physically based shading at Disney”, B. Burley Siggraph 2012

D: NDF shape

- Phong, Beckmann, Trowbridge-Reitz (GGX), etc…
  - GGX is proven to better match measurements
    - But several materials have a narrower peak and a longer tail
    - What could be a good shape ?

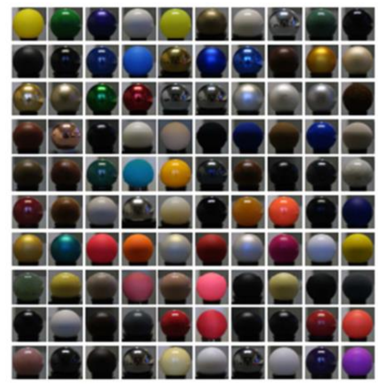


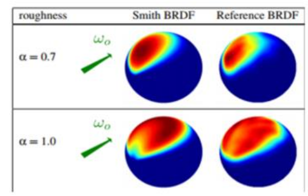
Image from "The MERL BRDF database", MERL 2007

GGX have been proven to be a better match, but it doesn't match all measurement [MERL06] https://www.merl.com/brdf/

G2: Shadowing-masking

- Smith's height-correlated function
  - Microfacets with higher elevation are more visible
    - Unlike uncorrelated Smith
  - Proved by simulation to be more accurate
    - From Gaussian Heightfield...

$$\frac{\chi^+(\omega_o \cdot \omega_m)\,\chi^+(\omega_i \cdot \omega_m)}{1+\Lambda(\omega_o)+\Lambda(\omega_i)}$$



Images from "Understanding the Masking-Shadowing Function in Microfacet-Based BRDFs", E. Heitz, jcgt

A more accurate form of the masking-shadowing function models the correlation between masking and shadowing due to the height of the microsurface. Intuitively, the more a microfacet is elevated within the microsurface, the more the probabilities of being visible for the outgoing direction (unmasked) and for the incident direction (unshadowed) increase at the same time. Thus, masking and shadowing are correlated through the elevation of the microfacets.

Note: Accuracy of height-correlated has been validated by brute force simulation on gaussian surface in
[Heitz14] E. Heitz, Understanding the Masking-Shadowing Function in Microfacet-Based BRDFs, JCGT

G2: Shadowing-masking

- End of story?
  - Real world observation not handled by Smith's height-correlated term
    - Hotspot/opposition effect
      - View and light are aligned
      - No more occlusion

G2: Shadowing-masking

- [Heitz14] Provide a better candidate term
  - Height-Direction-Correlated Masking and Shadowing
  - Open problem is to find **λ** in this equation

$$G_2(\omega_o, \omega_i, \omega_m) = \frac{\chi^+(\omega_o \cdot \omega_m)\,\chi^+(\omega_i \cdot \omega_m)}{1 + \max\left(\Lambda(\omega_o), \Lambda(\omega_i)\right) + \lambda(\omega_o, \omega_i)\,\min\left(\Lambda(\omega_o), \Lambda(\omega_i)\right)}$$

[Heitz14] E. Heitz, Understanding the Masking-Shadowing Function in Microfacet-Based BRDFs, JCGT

Masking and shadowing are also strongly correlated when the outgoing and incident directions are close to one another. Typically, when $\omega o = \omega i$, masking and shadowing are perfectly correlated because microfacets visible from direction ωo are also visible from direction ωi . In this case, the shadowing should be removed from the BRDF because shadowed microfacets are not visible from direction ωi , and thus they are also not visible from ωo. This is known as the “hotspot effect”: when the view and light directions are parallel, shadows disappear. Since the BRDF models the radiance measured along the outgoing direction, if shadowing exists on the surface but is not visible then it should not be part of the BRDF

Here, masking and shadowing are fully correlated when the outgoing and incident directions are parallel and $\lambda = 0$

The derivation of practical forms for λ and generalization non-Gaussian distributions are open problems

Shadowing-masking

- Other problem: view-dependent roughness
  - Rough surfaces appear smoother at grazing angles
    - Fresnel is not enough to explain this behavior
  - Question: All surfaces don't follow Smith's hypothesis?
    - Need new shadowing/masking?
  - Diffraction at grazing angles?
  - Microsurface measurement could help

Image from "Hideo Kojima GDC 2013 Panel - MGS5 & Fox Engine"

View dependent roughness term have been introduced at GDC 2013 in the panel about MGS5 and the fox engine. But there is not really further research on the topic. Still the phenomena exist, at grazing angles some surfaces are mirror like and is not due to Fresnel but rather due to visible normal.
Note from Naty Hoffman: It raise other questions "Don't follow smith hypothesis" which is all that's needed for the distribution of visible normals to change with view direction
(the Smith assumption - that visibility and orientation are uncorrelated - is in effect assuming that the distribution of visible normals does not change)

Note: Artists handle it in shader graph with roughness and view vector. Not physically based, no reciprocity

Beyond microfacet BRDFs?

- [Holzschuch17] Introduce Diffraction term in addition to Specular term
  - Terms fits well with a corrected MERL database
    - Results show cases where Diffraction part seems to match plausible physical phenomena behavior
  - Should we update the BRDF definition then?

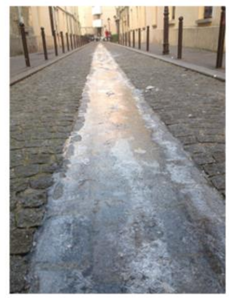
Image from "A Two-Scale Microfacet Reflectance Model Combining Reflection and Diffraction", N. Holzschuch, Siggraph 2017

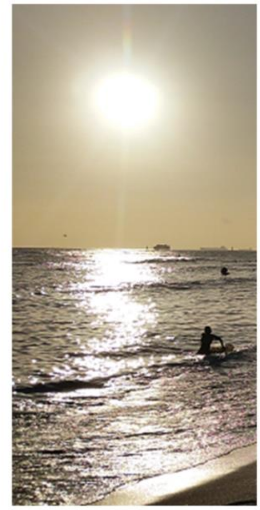
[Holzschuch17] N. Holzschuch, A Two-Scale Microfacet Reflectance Model Combining Reflection and Diffraction, Siggraph 17

This paper show that diffraction effects in the micro-geometry provide a plausible explanation to observe discrepancy of reflectance with microfacet model prediction. It introduce a two-scale reflectance model, separating between geometry details much larger than wavelength and those of size comparable to wavelength.
The former model results in the standard Cook-Torrance model. The latter model is responsible for diffraction effects.

Multiscale Representation

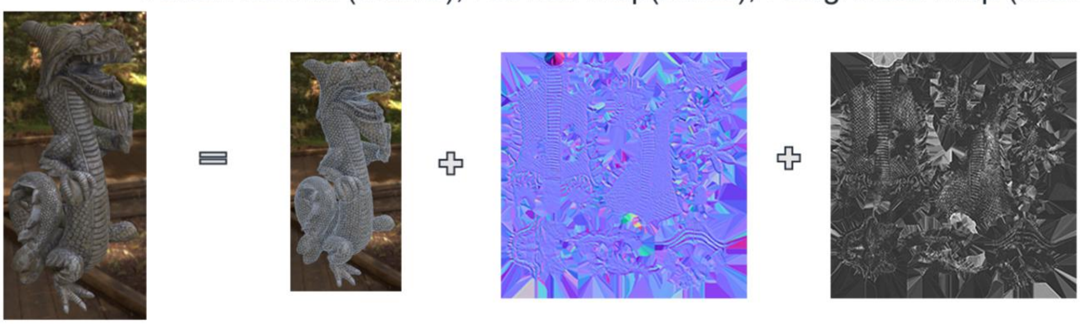
- A physically based materials conserve its appearance at all scales
  - Material stay rough, anisotropic, glinty... with distance
  - Related to normal distribution on surfaces



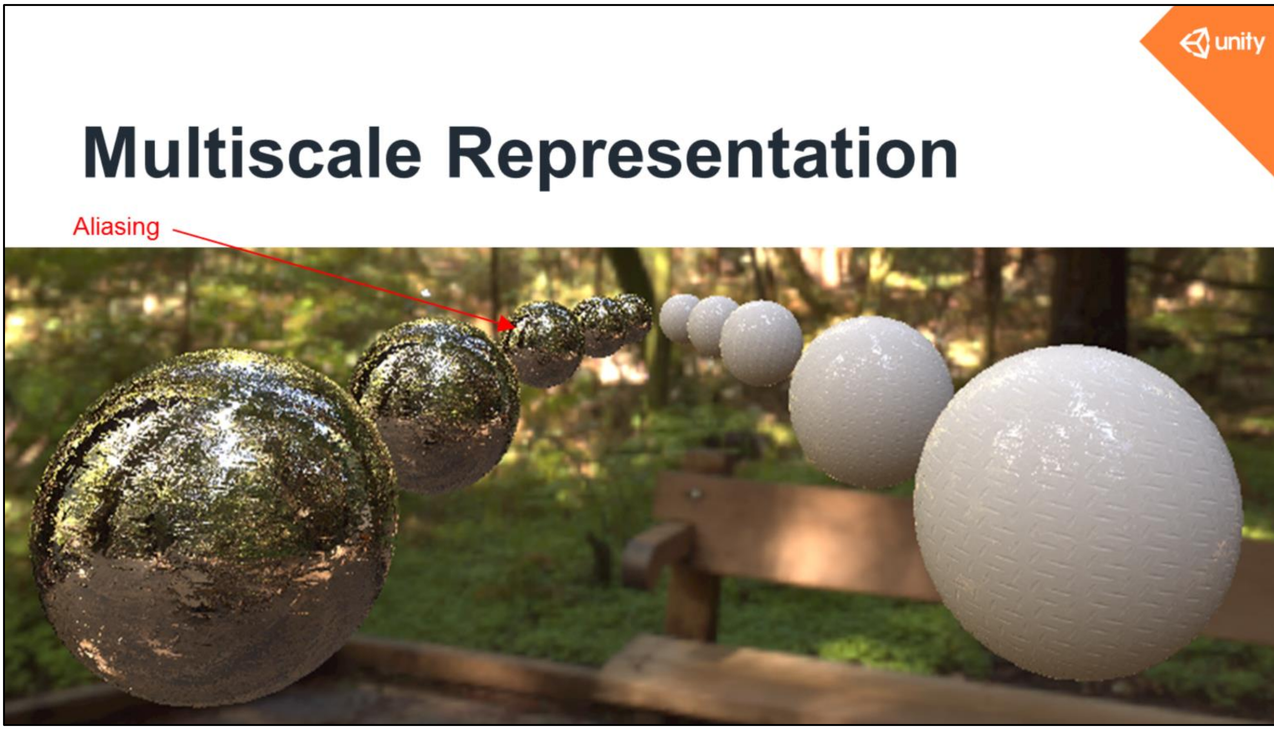


Multiscale Representation

- In computer graphics (CG) normal distribution are represented at various scale by
  - Vertex normal (Macro), Normal map(Meso), Roughness map (Micro - D)



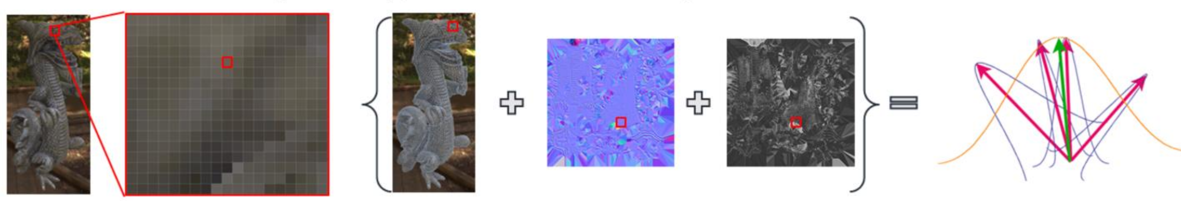
These representations have filtering issues. For example, when using normal map to add details
Normal are averaged by mipmap, a very well know problem. With distance the object become smooth.



Some artists try to disable mipmaps to solve it but then introduce aliasing

Multiscale Representation

- Similar issue happens with normal geometry filtering
- How to conserve appearance at all scale in CG?
  - Fold all pixel's footprint surface information to an NDF (i.e. new D)
    - Transfer information from one scale to the other. But how ?
    - The problem applies to both diffuse and specular BRDF



Images from "How to design your assets for physically based rendering", Y. Gotanda, Cedec 2012

Note that the multiscale problem don't apply only to specular but also to diffuse. Both derive from the same NDF.

Energy conservation

- Energy conservation?
  - We hear about it all the time with PBR
  - No energy loss in ideal condition with no absorption
    - NDFs must be normalized
    - BRDFs must pass the white furnace test
      - White environment lighting a sphere with Fresnel term = 1 result in white sphere

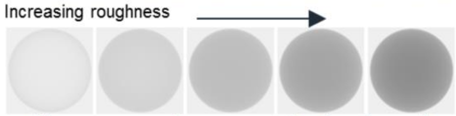
Image from “Extending the Disney BRDF to a BSDF with Integrated Subsurface Scattering”, B. Burley, Siggraph 2015

A white furnace test is like putting a white sphere with a BRDF where Fresnel term is set to 1 in a white environment, the sphere should be white if it passes the test, i.e is energy conserving

Energy conservation

- Almost all current BRDF fail the white furnace test
  - Rougher materials are darker due to energy loss

Increasing roughness →



  - Because they only model single scattering
- [Heitz16] introduce a multiple scattering framework
  - A physically based material is multiscattering

Images from "Multiple-Scattering Microfacet BSDFs with the Smith Model", E.Heitz, Siggraph 2016

[Heitz16] E. Heitz, Multiple-Scattering Microfacet BSDFs with the Smith Model, Siggraph 2016

Energy conservation

- We just discussed energy conservation of a BRDF
- What about energy conservation between BRDFs?
  - This requires defining a physical representation
  - Important for layered BRDFs
  - And this is where the Fresnel term comes into play

$$r_{\perp} = \frac{n_i \cos\theta_i - n_t \cos\theta_t}{n_i \cos\theta_i + n_t \cos\theta_t}$$

$$r_{\parallel} = \frac{n_t \cos\theta_i - n_i \cos\theta_t}{n_t \cos\theta_i + n_i \cos\theta_t}$$

Energy conservation

- We can consider different physical representations
  - Specular BRDF on top of diffuse BRDF
  - Specular + Diffuse as a BRDF
    - Subsurface scattering distance is small as compared to the size of the microstructure
      - Don't fit most materials but doesn't mean the resulting visual is not good
  - Others…

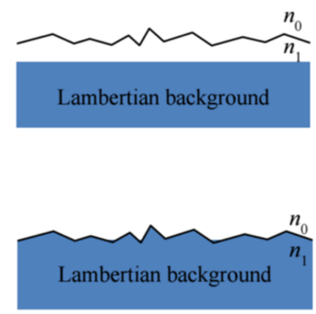


Image from "Rendering Rough Opaque Materials with Interfaced Lambertian Microfacets, D. Meneveaux, 2017

Sum up

- What is a physically based material?
  - Based on a physical representation
    - Energy conservation
      - Multi scattering, layered Fresnel interfaces
  - NDF Shape that match measurement
    - Isotropic and anisotropic
  - Diffuse and specular term derived from normalized NDF
  - Multiscale representation
- Let’s see what's available out there

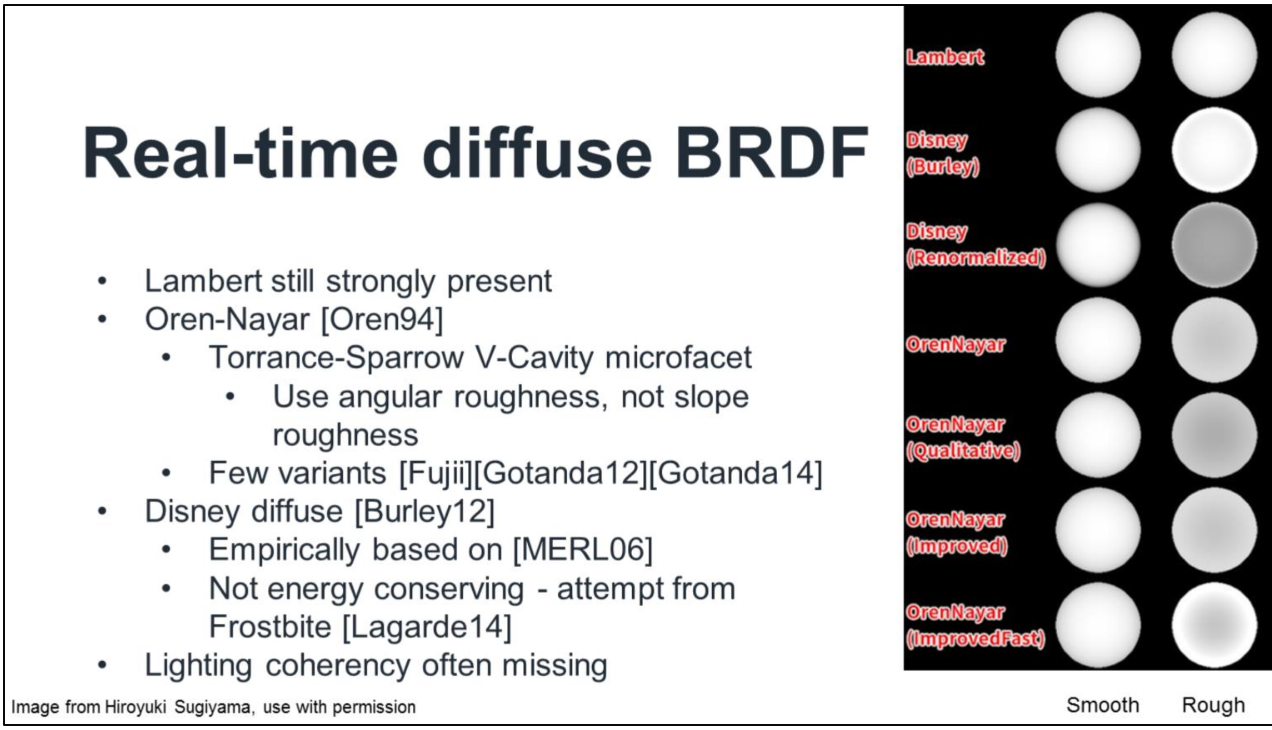
Energy conservation inside and between BRDF

Where are we
(in real-time rendering)?

unity

Sadly global illumination (indirect diffuse only, abuse term by game engine, i.e. in this case from lightmap/lightprobe), area and environment lights integration are often not considered when adding new material in an engine. It is important that a material interact correctly with a giving light type.
Lighting and material are decoupled on artists side but are often coupled on the code side for performance reasons (only for cubemap, for GI and area it is currently only a pre-integratoin of the BRDF on a given solid angle, unless the area light is textured)

Coherency for both diffuse and specular terms
The problem with pre-integration is dimensionality explosion.



[Oren94], M. Oren, S. Nayar, Generalization of Lambert's Reflectance Model, Siggraph 1994
This paper also propose the Qualitative mode
Oran-Nayar derive equations using numerical integration, but rely on the Torrance-Sparrow V-Cavity model. A model from the "old time" ☺
[Fujii] Y. Fujii, A A tiny improvement of Oren-Nayar reflectance model
This paper introduce Improved, ImproveFast
[Gotanda12] Y. Gotanda, Beyond a Simple Physically Based Blinn-Phong Model in Real-Time, Siggraph 2012
This paper introduce optimization
[Gotanda14] U. Gotanda, Designing Reflectance Models for New Consoles, Siggraph 2014
This paper introduce energy conservation with a Fresnel term

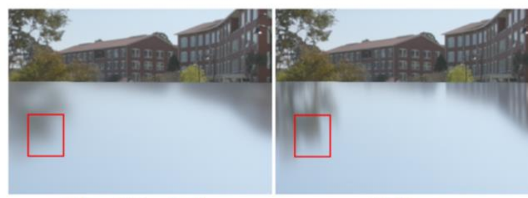
[Burley12] B. Burley, Physically based shading at Disney, Siggraph 2012
Derive from observation of MERL database
Disney: Darkening at incoming/outgoing grazing angle for smooth surface, brightening for rough (Backscatter)
Note that this behavior is due to their choice of term separation

Note: the Disney BRDF has a Sheen term to compensate for the energy loss due to the lack of multiple scattering
It is not energy conserving, Lagarde and de Rousiers attempted to make it energy

conserving but doesn’t look it is the correct way to do it.
Lighting coherency is often missing in game engine. If disney or Oren-Nayar is use, it must be integrated with lightmap/lightprobe too.
And Lightmaper often work with Lambert only, relying on a post-step approximation to make it coherent. Same thing for the area lights.

Real-time specular BRDF

- Trowbridge-Reitz (GGX)
  - With Smith's Height-correlated Masking-Shadowing function
  - Lighting coherency is often supported
    - Environment pre-integration
      - With simplifications [Karis13]
        - L = V
    - Area lights pre-calculation
      - Linear cosine transform [Heitz16b]



(a) Isotropic lobe assumption. (b) Reference.

Image from "Moving Frostbite to PBR", S. Lagarde, Siggraph 2014

[Karis13] B. Karis, "Real Shading in Unreal Engine 4", Siggraph 2013
[Heitz16b] E. Heitz, "Real-time polygonal-light shading with linearly transformed cosines", Siggraph 2016

Environment map integration in game rely on split integral approximation popularized by Karis (But Gotanda have presented it in 2010 and Drobot also in a presenation of KZ4 before him).
The lighting and the NDF is coupled in this case.
The pre-integration of cubemap is simplified to reduce dimensionality explosion with L = V in pre-integration step, resulting in missing grazing angle stretching effect in practice

In the case of area lights, there is a pre-calculation of a matrix transformation to convert from specular lobe to diffuse lobe. With textured area lights, the textures also need to be pre-integrated with the NDF

Real-time BRDF

- Going further?
  - More physically-accurate diffuse BRDF
  - NDF shape
    - Anisotropy, shape control
  - Multiscale representations
  - Multiple scattering
  - Iridescence / Thin-film

Real-time BRDF

- Going further?
  - More physically-accurate diffuse BRDF
  - NDF shape

$$\rho(\omega_o, \omega_i) = \frac{1}{|\omega_g \cdot \omega_o| \, |\omega_g \cdot \omega_i|} \int_\Omega \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_2(\omega_o, \omega_i, \omega_m) D(\omega_m) \, d\omega_m$$

  - Multiple scattering
  - Iridescence / Thin-film

More physical diffuse BRDF

- More physically-accurate diffuse
  - Inherits properties of the chosen NDF
    - Example: Anisotropic GGX diffuse BRDF
  - Need an efficient approximation (a la Oren-Nayar)
    - [Gotanda15] and [Hammon17] provide an approximation for isotropic Height-Correlated GGX diffuse
      - Use Fresnel Schlick
  - [Gotanda15] use Facet BRDF: $\rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) = \frac{1}{\pi}(1 - F(\omega_i, \omega_m))$
    - Expensive approximation

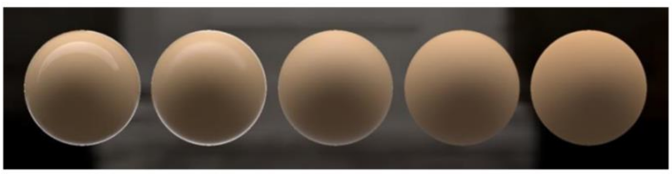
Image from "Designing Reflectance Models for New Consoles", Y. Gotanda, Siggraph 2014

[Gotanda15] Y. Gotanda, Designing Physically Based Microfacet Models for Next Generation, Cedec 2015
[Hammon17] E. Hammon, PBR Diffuse Lighting for GGX+Smith Microsurfaces, GDC 2017

Gotanda have provide an expensive approximation with a Facet BRDF that include Schlick Fresnel term to consider energy conservation. But this term is not physical as it is not reciprocal

More physical diffuse BRDF

- [Hammon17] use Facet BRDF: $\rho = F\,\rho_{spec} + (1-F)\frac{1.05}{\pi}(1-(1-N\cdot V)^5)$
  - Used in *Titanfall 2*
  - Derived energy conservation term is the same than [Shirley95]
    - Include reciprocity
- How to achieve lighting coherency?



- $facing = 0.5 + 0.5\,L\cdot V$
- $rough = facing(0.9 - 0.4 facing)\left(\frac{0.5+N\cdot H}{N\cdot H}\right)$
- $smooth = 1.05(1-(1-N\cdot L)^5)(1-(1-N\cdot V)^5)$
- $single = \frac{1}{\pi} lerp(smooth, rough, \alpha)$
- $multi = 0.1159\alpha$
- $diffuse = albedo * (single + albedo * multi)$

Image from “PBR Diffuse Lighting for GGX+Smith Microsurfaces”, E. Hammon, GDC 2017

[Shirley57] P. Shirley, A Practitioners’ Assessment of Light Reflection Models, 1995 Equation (5)
[Hammon17] E. Hammon, PBR Diffuse Lighting for GGX+Smith Microsurfaces, GDC 2017

The facet BRDF term is more complete than the one from Gotanda as it is reciprocal. It also match the derivation from Shirley for energy conservation between specular and diffuse. But this derivation is based on Fresnel Schlick, not the real Fresnel term.
Note 1.05 is 21 / 20 as in Shirley and Gotanda attempt
new consensus ?
Hammon exhibit darkening at smooth edge and backscattering, similar properties as Disney difffuse. But Hammon have use his model only for direct lighting, not indirect lighting.
For lighting coherency we may pre-integrate single and multi in two different term to apply on albedo and albedo square.
What about LTC for area Light ?

More physical diffuse BRDF

- [Meneveaux17] uses Facet BRDF: $f_b^\mu(\mathbf{i},\mathbf{o},\mathbf{m}) = \frac{1}{\pi n_i^2} T(\mathbf{i},\mathbf{m}) T(\mathbf{o},\mathbf{m}) \frac{K_d}{(1 - K_d r_i)}$
  - Exact Fresnel and interface multiple reflection
  - Not for real-time

$$n_i^2(1 - r_i) = 1 - r_e$$

$$r_e = \frac{1}{2} - \frac{2n_i^3(n_i^2+2n_i-1)}{(n_i^2+1)(n_i^4-1)} + \frac{(n_i-1)(3n_i+1)}{6(n_i+1)^2} + \frac{8n_i^4(n_i^4+1)}{(n_i^2+1)(n_i^4-1)^2}\ln(n_i) + \frac{n_i^2(n_i^2-1)^2}{(n_i^2+1)^3}\ln\left(\frac{n_i-1}{n_i+1}\right)$$

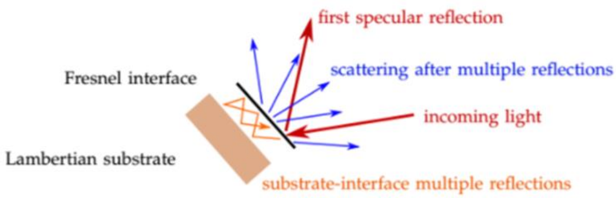


Image from “Rendering Rough Opaque Materials with Interfaced Lambertian Microfacets, D. Meneveaux, 2017

[Meneveaux17] D. Meneveaux, Rendering Rough Opaque Materials with Interfaced Lambertian Microfacets, 2017

Previous BRDF use Shlick Fresnel, but to do it correctly it should use exact Fresnel Term and take into account the multiple reflection within the Fresnel interface.

More physical diffuse BRDF

- [Meneveaux17] non real-time approximation provided for anisotropic Beckmann with uncorrelated smith diffuse

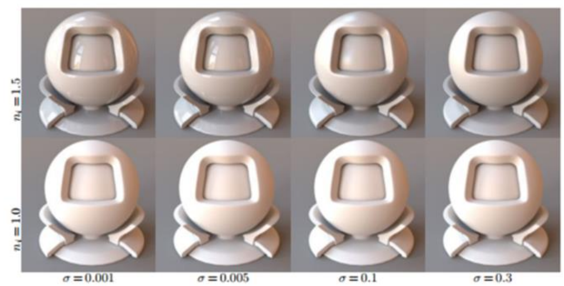


Image from “Rendering Rough Opaque Materials with Interfaced Lambertian Microfacets, D. Meneveaux, 2017

[Meneveaux17] D. Meneveaux, Rendering Rough Opaque Materials with Interfaced Lambertian Microfacets, 2017
There is darkening at smooth edge and backscatter behavior like with Hammon.

More physical diffuse BRDF

- Previous approaches use Specular + Diffuse as a BRDF for physical representation
  - Specular and diffuse use same roughness
    - Convenient for games since it has fewer parameters
    - Don't match most of real world surface representation

Image from "Rendering Rough Opaque Materials with Interfaced Lambertian Microfacets, D. Meneveaux, 2017

Previous approach follow a physical representation of specular+diffuse BRDF that I describe in previous section, it mean that diffuse and specular have the same properties, like same roughness.
This is convenient but this representation doesn't match most of real world surface.
Maybe a better way will be to handle diffuse BRDF and specular BRDF as layer instead,

More physical diffuse BRDF

- Specular BRDF on top of diffuse BRDF instead ?
  - Fresnel term is not in Facet BRDF but between the BRDF layers
    - Separate roughness
      - More control for the artists
    - Common in VFX industry
  - Find efficient approximation for anisotropic Height-Correlated GGX diffuse BRDF with correct Fresnel interface with Specular BRDF ?

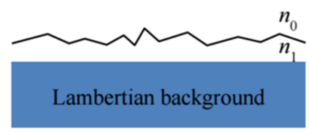


Image from “Rendering Rough Opaque Materials with Interfaced Lambertian Microfacets, D. Meneveaux, 2017

Roughness for diffuse will be different from roughness of specular. This provide more control by the artists and is already a well established practice in VFX industry.

Then the problem is to find an approximation for a simple Facet BRDF of 1/PI. Should be easier than what have been done by previous work.
Most difficult part is to find a correct approximation from a Rough Specular, Rough diffuse Fresnel interface with multiple reflection.

Still it require one extra roughness parameters (2 for anisotropy), thus why game it is not use in game due to gbuffer storage.

Real-time BRDF

- Going further?
  - More physically-accurate diffuse BRDF
  - NDF shape
    - Anisotropy, shape control
  - Multiscale
  - Multiple scattering
  - Iridescence / Thin-film

NDF shape - anisotropy

- Few game engines use the anisotropic GGX NDF
  - Requires tangent + extra roughness parameters
    - Roughness often remapped from anisotropy factor [Burley12]
  - Visibility term must be derived from anisotropic NDF

$$D(\omega_m) = \frac{\chi^+(\omega_m \cdot \omega_g)}{\pi \alpha_x \alpha_y \cos^4 \theta_m \left(1 + \tan^2 \theta_m \left(\frac{\cos^2 \phi_m}{\alpha_x^2} + \frac{\sin^2 \phi_m}{\alpha_y^2}\right)\right)^2}$$

$$\Lambda(\omega_o) = \frac{-1 + \sqrt{1 + \frac{1}{a^2}}}{2}$$

$$a = \frac{1}{\alpha_o \tan \theta_o} \qquad \alpha_o = \sqrt{\cos^2 \phi_o \alpha_x^2 + \sin^2 \phi_o \alpha_y^2}$$

[Burley12] B. Burley, Physically based shading at Disney, Siggraph 2012

Anisotropy is not something that is common in game engine due to its cost and extra complexity
Something often missed is shadowing and masking term derive from NDF, and thus if the NDF is anisotropic, the visibility term is.

NDF shape - anisotropy

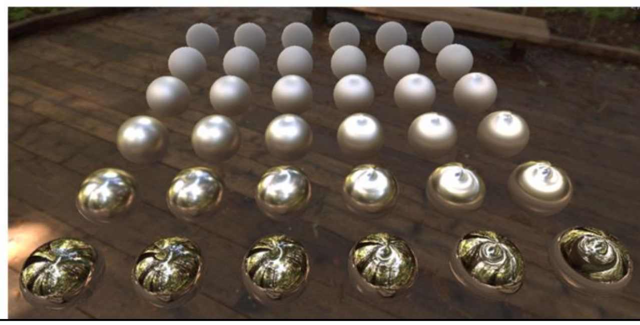
After simplification (Height correlated visibility term: G pre divide by (4.0 * NdotL * NdotV)) [McAuley15]

```
// roughnessT -> roughness in tangent direction
// roughnessB -> roughness in bitangent direction
float D_GGXAnisoNoPI(float TdotH, float BdotH, float NdotH, float roughnessT, float roughnessB)
{
    float f = TdotH * TdotH / (roughnessT * roughnessT) + BdotH * BdotH / (roughnessB * roughnessB) + NdotH * NdotH;
    return 1.0 / (PI * roughnessT * roughnessB * f * f);
}

float V_SmithJointGGXAniso(float TdotV, float BdotV, float NdotV, float TdotL, float BdotL, float NdotL, float roughnessT, float roughnessB)
{
    float aT = roughnessT;
    float aT2 = aT * aT;
    float aB = roughnessB;
    float aB2 = aB * aB;

    float lambdaV = NdotL * sqrt(aT2 * TdotV * TdotV + aB2 * BdotV * BdotV + NdotV * NdotV);
    float lambdaL = NdotV * sqrt(aT2 * TdotL * TdotL + aB2 * BdotL * BdotL + NdotL * NdotL);

    return 0.5 / (lambdaV + lambdaL);
}
```

[McAuley15] S. McAuley, The rendering of far cry 4, Cedec 2015

NDF shape - anisotropy

- Lighting coherency is very hard: Dimensionality
  - Games use a normal vector hack [Revie11][McAuley15]

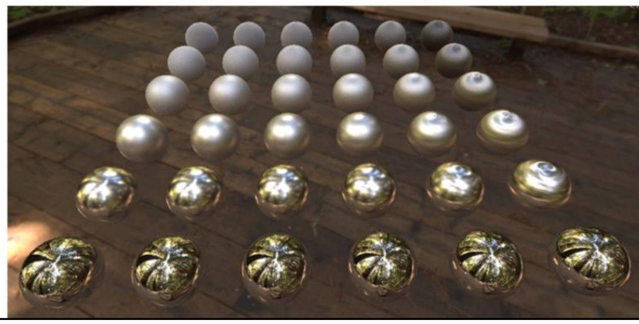


[Pesce15] A. Pesce, M. Iwanicki, Approximate Models For Physically Based Rendering, Siggraph 2015 - no hardware anisotropic filtering for cubemap
[Revie11] D. Revie, Implementing Fur Using Deferred Shading, GPU Pro 2
[McAuley15] S. McAuley, The rendering of far cry 4, Cedec 2015

Supporting pre-integration with cubemap is not possible due to dimmensionnality.
Game rely on a normal vector hack introduce by Revie and use in Far car 4 and The order 1886.
The hack simply bend the normal vector based on anisotropy and view direction. It still perform a single fetch. Which is convenient.
It may be visually pleasant but I want to show how far it is from the referene.
Note that when moving the camera, the highlights also move (inherent to the hack use), whereas the highlight must be stable.

NDF shape - anisotropy

- But it is far from the reference
  - Need a better hack
    - And with the current hack highlights move with the camera



Noise is due to undersampling.
Reference done in engine with importance sampling of anisotropic GGX.
See how the smooth case is totally wrong. When you are smooth, there is no anisotropy.
We need a better hack that handle this case and stretch the highlight more closely. (Unless someone found a good accurate way to do it ☺).

Specular anisotropy - Hack

```
// The grain direction (e.g. hair or brush direction) is assumed to be orthogonal to the normal.
// The returned normal is NOT normalized.
float3 ComputeGrainNormal(float3 grainDir, float3 V)
{
    float3 B = cross(-V, grainDir);
    return cross(B, grainDir);
}

// Fake anisotropic by distorting the normal.
// The grain direction (e.g. hair or brush direction) is assumed to be orthogonal to N.
// Anisotropic ratio (0->no isotropic; 1->full anisotropy in tangent direction)
float3 GetAnisotropicModifiedNormal(float3 grainDir, float3 N, float3 V, float anisotropy)
{
    float3 grainNormal = ComputeGrainNormal(grainDir, V);
    return normalize(lerp(N, grainNormal, anisotropy));
}

// Tangent = highlight stretch (anisotropy) direction. Bitangent = grain (brush) direction.
float3 anisoIblNormalWS = GetAnisotropicModifiedNormal(bitangentWS, normalWS, V, anisotropy);
iblR = reflect(-V, anisoIblNormalWS);
```

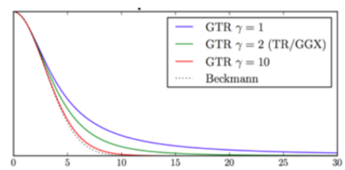
Specular anisotropy - Ref

```
void SampleAnisoGGXDir(float2 u, float3 V, float3 N,
                       float3 tangentX, float3 tangentY,
                       float  roughnessT, float roughnessB,
                       out float3 H, out float3 L)
{
    // AnisoGGX NDF sampling
    H = sqrt(u.x / (1.0 - u.x)) *
        (roughnessT * cos(TWO_PI * u.y) * tangentX + roughnessB * sin(TWO_PI * u.y) * tangentY) + N;
    H = normalize(H);

    // Convert sample from half angle to incident angle
    L = 2.0 * saturate(dot(V, H)) * H - V;
}

void ImportanceSampleAnisoGGX(float2 u, ...)
{

    float3 H;
    SampleAnisoGGXDir(u, V, N, tangentX, tangentY, roughnessT, roughnessB, H, L);

    float NdotH = saturate(dot(N, H));
    // Note: since L and V are symmetric around H, LdotH == VdotH
    VdotH = saturate(dot(V, H)); NdotL = saturate(dot(N, L));
    // For anisotropy we must not saturate these values
    float TdotV = dot(tangentX, V); float BdotV = dot(tangentY, V);
    float TdotL = dot(tangentX, L); float BdotL = dot(tangentY, L);

    float Vis = V_SmithJointGGXAniso(TdotV, BdotV, NdotV, TdotL, BdotL, NdotL, roughnessT, roughnessB);
    weightOverPdf = 4.0 * Vis * NdotL * VdotH / NdotH;
}
```

NDF shape - more control

- Generalized Trowbridge & Reitz (GTR) [Burley12]
  - **γ** [0..10], match GGX for **γ** = 2, singularity at **γ** = 1
  - Not shape-invariant
  - Shadowing-Masking function is hard to derive
    - Discrete value of **γ** in (0, 4] spline interpolated [Dimov15] for height-correlated visibility term

$$D_{\mathrm{GTR}}(\theta_h) = \frac{(\gamma - 1)(\alpha^2 - 1)}{\pi(1 - (\alpha^2)^{1-\gamma})} \frac{1}{(1 + (\alpha^2 - 1)\cos^2\theta_h)^\gamma}$$



Images from “Physically based shading at Disney”, B. Burley Siggraph 2012

[Burley12] B. Burley, Physically based shading at Disney, Siggraph 2012
[Dimov15] R.Dimov, Deriving the Smith shadowing function G1 for γ in (0, 4], Chaos group documentation 2015

Shape-invariant property is important as it allow to more easily derive a shadowing and masking term and perform other operation. Here GTR is not invariant, so it is hard to find an analytic term for shadowing and masking. Rossen Dimov from Chaos Group has derived a Smith shadowing function for discrete GTR values, and uses interpolated values in-between: Deriving the Smith shadowing function G1 for γ in (0, 4]

NDF shape - more control

- GGX's shadowing-masking term for everything

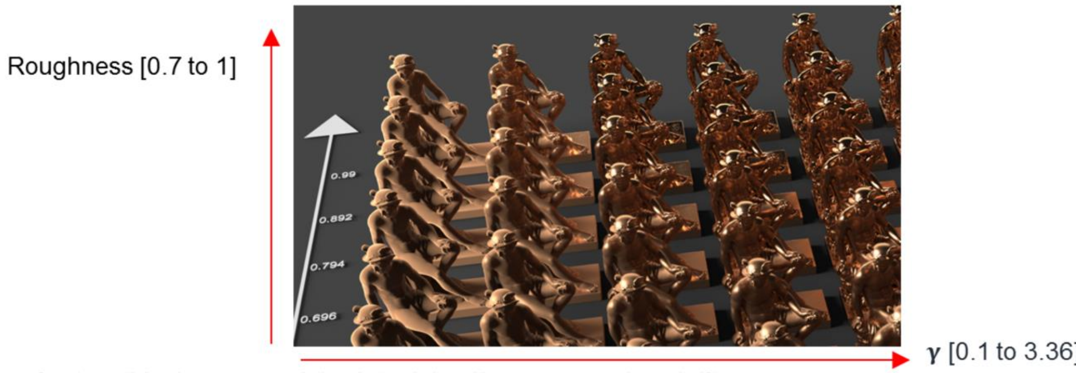


Image from https://labs.chaosgroup.com/index.php/rendering-rd/improvements-to-the-gtr-brdf/

Left to right is increasing gamma value from 0.1 to 3.36, bottom to top is roughness from 0.7 to 1.
GGX shadowing-masking term is use for everything

NDF shape - more control

- Correct shadowing-masking term is important for rough objects

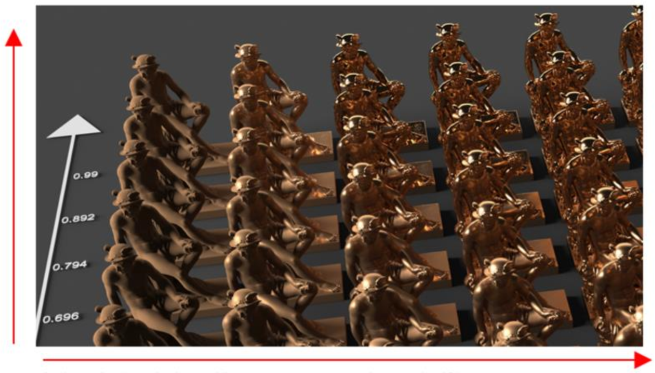


Image from https://labs.chaosgroup.com/index.php/rendering-rd/improvements-to-the-gtr-brdf/

Left to right is increasing gamma value from 0.1 to 3.36, bottom to top is roughness from 0.7 to 1.
Approximated GTR shadowing-masking, this show how it is important to use the correct term.

NDF shape - more control

- Bivariate Student-T distribution (STD) [Ribardiere17]
  - $\boldsymbol{\gamma}$ [1.5, 40], Include GGX ($\boldsymbol{\gamma}$=2), Beckmann ($\boldsymbol{\gamma}$ = 40), singularity at $\boldsymbol{\gamma}$ = 1.5
  - Shape-invariant
  - Analytic G2 term
  - Cons
    - Not real-time
    - Low interest for Beckmann shape vs GGX shape
    - Peak/Longer tail control less interesting than GTR

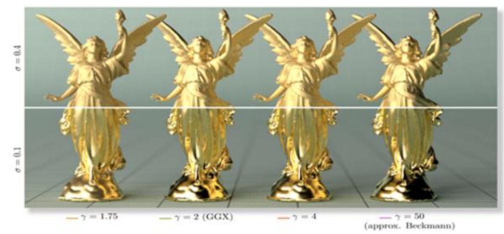


Image from “STD: Student’s t-Distribution of Slopes for Microfacet Based BSDFs”, M. Ribardière, eurographics 2017

[Ribardiere17] STD: Student’s t-Distribution of Slopes for Microfacet Based BSDFs, eurographics 2017

This paper try to fix the issue with GTR and provide a shape-invariant version.
Shape enables relatively straightforward derivation of an anisotropic form, Smith G, distribution of visible normals, etc.
Their term however is really heavy, the paper also provide an approximation but it is expensive.
Also the shape control is less interesting than GTR.

NDF shape - more control

- Want GTR shadowing-masking efficient approximation?
  - Lighting coherency?
  - γ control blurriness like roughness
- Want a different shape control?
  - Haze/Halo perception control [Vangorp17][Kulla17]

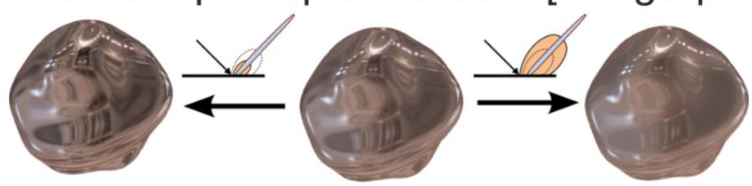


Image from "The perception of hazy gloss", P. Vangorp, Journal of vision 2017

Even if we adopt GTR we need an efficient approximation of the shadowing-masking term.
Also shape control increase size dimensionality of pre-integration, need one extra parameters, not compatible with cubemap...
Lastly, the gamma and roughness parameter both control the blurriness aspect of the lobe, making it difficult to know which one to chose to control the shape.

[Vangorp17] P. Vangorp, "the perception of Hazy gloss",
They show that haziness is not only a readily perceivable material quality, but that it is distinct from the blur quality
"we find that certain aspects of our data can be explained by a nonphysical decomposition into a central reflection peak flanked by a halo component. We suggest that it is the presence of the halo component that is responsible for the perception of hazy gloss. "

Two lobes, narrow and wide to simulate hazy effect.
This will be compatible with current approach, just need to run all the lighting code twice, like for layering and lerp the lobe.

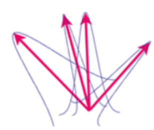
It is use in production at Imagework. [Kulla17] C. Kulla, "Revisiting Physically Based Shading at Imageworks", Siggraph 2017

Real-time BRDF

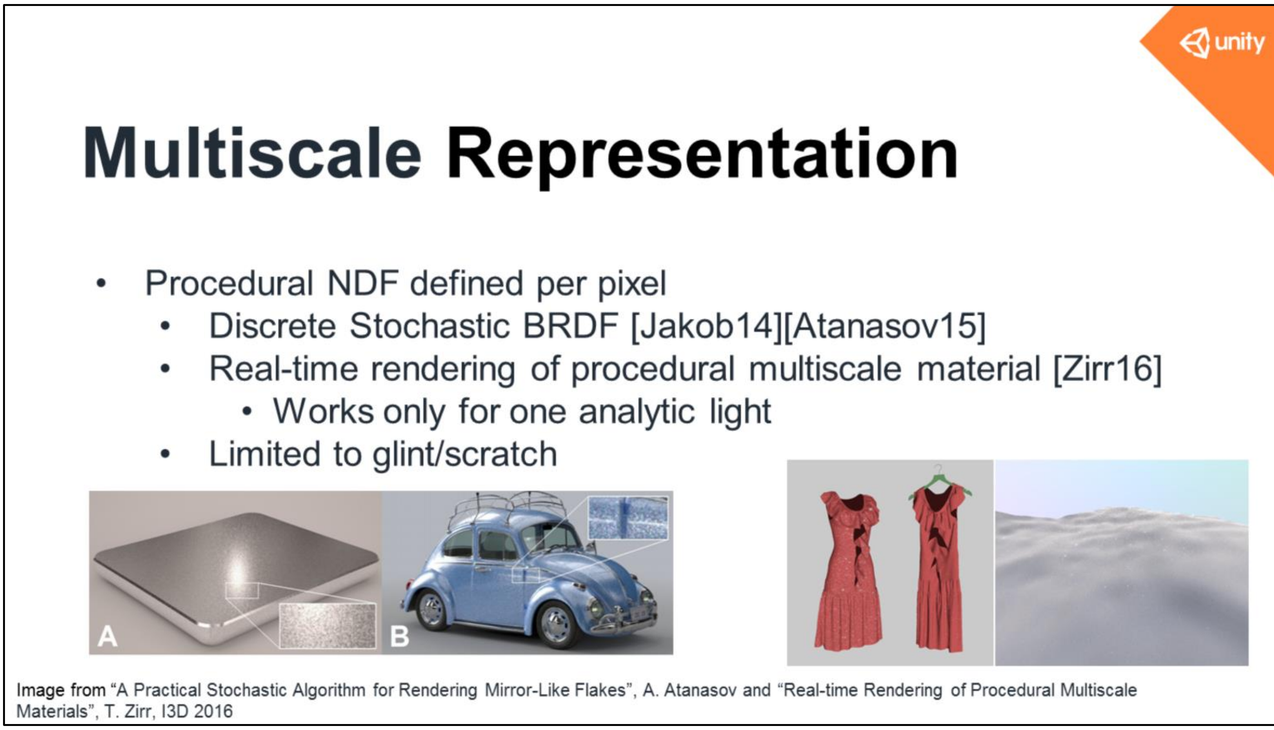
- Going further?
  - More physically-accurate diffuse BRDF
  - NDF shape
    - Anisotropy, shape control
  - Multiscale Representation
  - Multiple scattering
  - Iridescence / Thin-film

Multiscale Representation

- Related to NDF and CG surface representation
- What if we rely solely on an NDF instead ?
  - Need multiscale glinty/spiky NDF
    - Keep high frequency information
    - Query over finite areas and solid angles
      - i.e estimates pixel’s footprint NDF at any scale



Remember: want to conserve appearance with distance
Queried not at a single points and directions but over finite areas and solid angles



[Jakob14] W. Jakob, Discrete Stochastic Microfacet Models, Siggraph 2014
Jakob et al. stochastically model the number of discrete slopes covered by the pixel footprint and can be quickly evaluated by a hierarchical subdivision of the latter in the microfacet domain.
[Atanasov15] A. Atanasov, A Practical Stochastic Algorithm for Rendering Mirror-Like Flakes, Chaos group documentation 2015
https://www.fxguide.com/quicktakes/v-rays-practical-stochastic-rendering-of-spec-y-things/
The method works by assuming that there are N flakes that are uniformly distributed in a unit of texture space and their normals follow a micro-facet distribution on that unit's rendering or lighting equation hemisphere. The multiscale BRDF is defined as the microfacet BRDF, averaged over a finite surface area. In the implementation, the algorithm works with a patch or parallelogram approximation of the pixel footprint. Into this mix goes more complex techniques such as caching, overlapping sampling of the parallelograms and optimal Importance Sampling. A key advantage of the algorithm is that the flakes are not stored in memory, but their counts in this patch are reproduced by a deterministically seeded stochastic process.

[Zirr] T. Zirr, Real-time Rendering of Procedural Multiscale Materials, I3D 2016
Zirr paper method: Get a random number to turn on a pixels. It is spatially and angulary stable (inside a cone we will trigger the same number).
Note: the approach is not physically based, we can see 2 glint in two different light direction at the same time (but should not be visible in practice). This is the limitation

of not having a true NDF.
Define a macro and a micro NDF and compose them: In practice it is just a box filter that allow to mimic a Beckmann shape (as we do in rendering for gaussian postprocessing)

Multiscale Representation

- Instead of starting from procedural NDF, use high resolution normal map to defines the pixel NDF?
  - Better artist-friendly control
    - Position-Normal distribution [Yan 2016]
  - Similar goal as normal map filtering algorithm

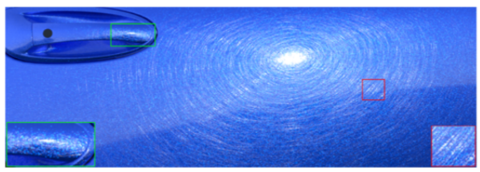


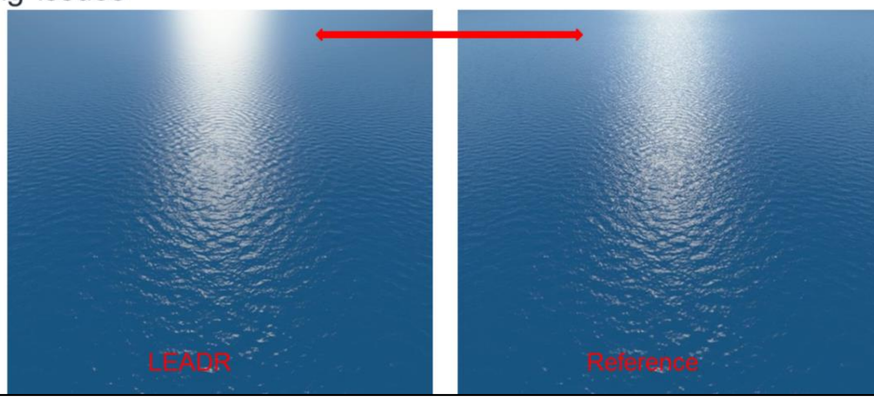
Image from "Position-Normal Distributions for Efficient Rendering of Specular Microstructure", L. Yan, Siggraph 2016

[Yan16], L. Yan, Position-Normal Distributions for Efficient Rendering of Specular Microstructure, Siggraph 2016
Yan proposed another method for filtering spatially varying microstructure. They handle a high-resolution normal map texture by resolving a mesoscale NDF defined on a pixel footprint by hierarchically pruning irrelevant normal map texels.
The paper represent high resolution normal with a mixture of Gaussians. Combine a macro-level standard normal map and a micro-level normal map. NDF solely defined on normal map
Defining microstructure pattern with textures is attractive for artists. Not real time

Skipped by lack of time:
Multi-Scale Rendering of Scratched Materials using a Structured SV-BRDF Model [Raymond 16]
When getting farther away, reflections from individual scratches may still remain visible even though they are much smaller than a pixel: they lead to glint lines At a distance, the distribution of scratches still has a visible impact on appeareance: it modulates highlight silhouettes and smears environment reflections. Scratched materials thus require a BRDF model that is not only spatially-varying, but also multi-scale

Multiscale Representation

- But Toksvig[Tokvis04] / LEAN [Olano10] / LEADR [Dupuy13] have filtering issues



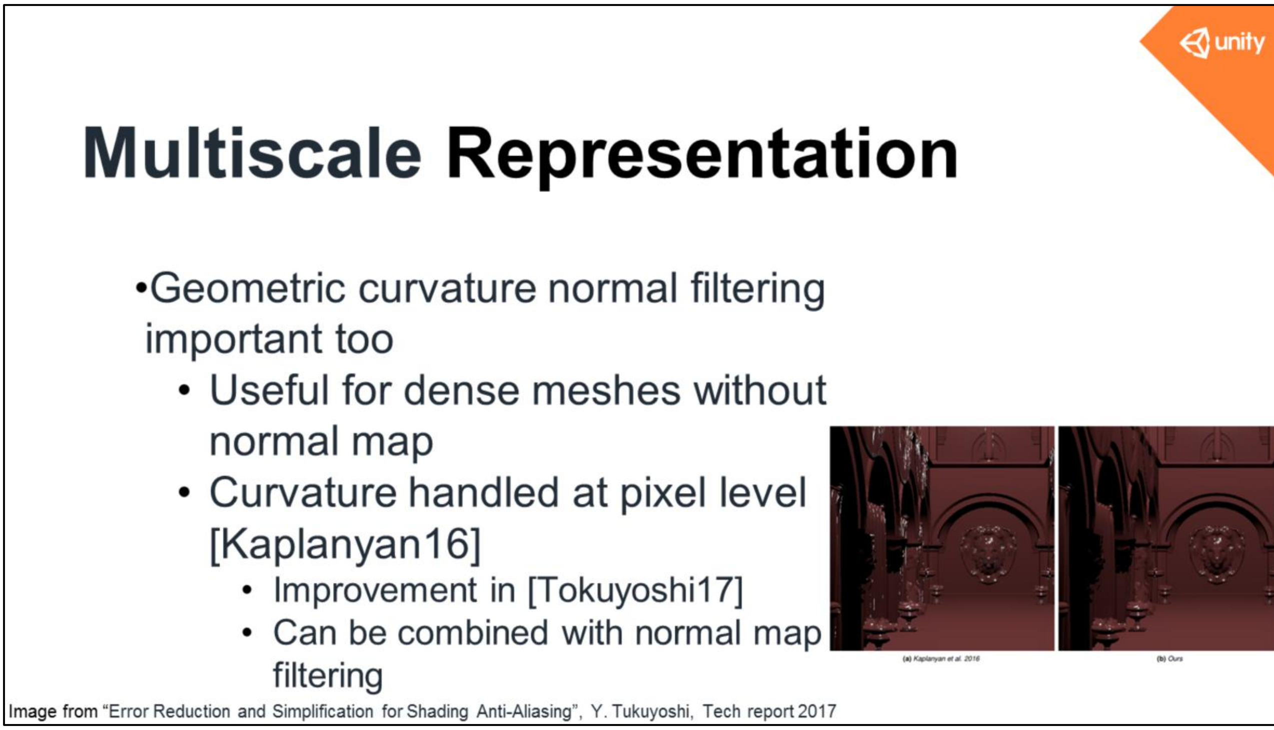
Except well-known normal map filtering algorithm don’t solve the problem fully for very far distance
Also note that none of these algorithm work with GGX.
There is also the Bum to Roughness method from Pixar
None of this algorithm rely on GGX (but Pixar claim that it is ok in their case).

Multiscale Representation

- These algorithm can't reproduce sharp features as normal filtering are based on smooth NDF (Beckman for LEADR)
  - Need to transfer information to spiky NDF
  - But want to keep original NDF
    - Real-Time Linear BRDF Mip-Mapping [Xu17]
      - BRDF and normal map are convolved in texture space then are mipmapped
      - Promising but expensive

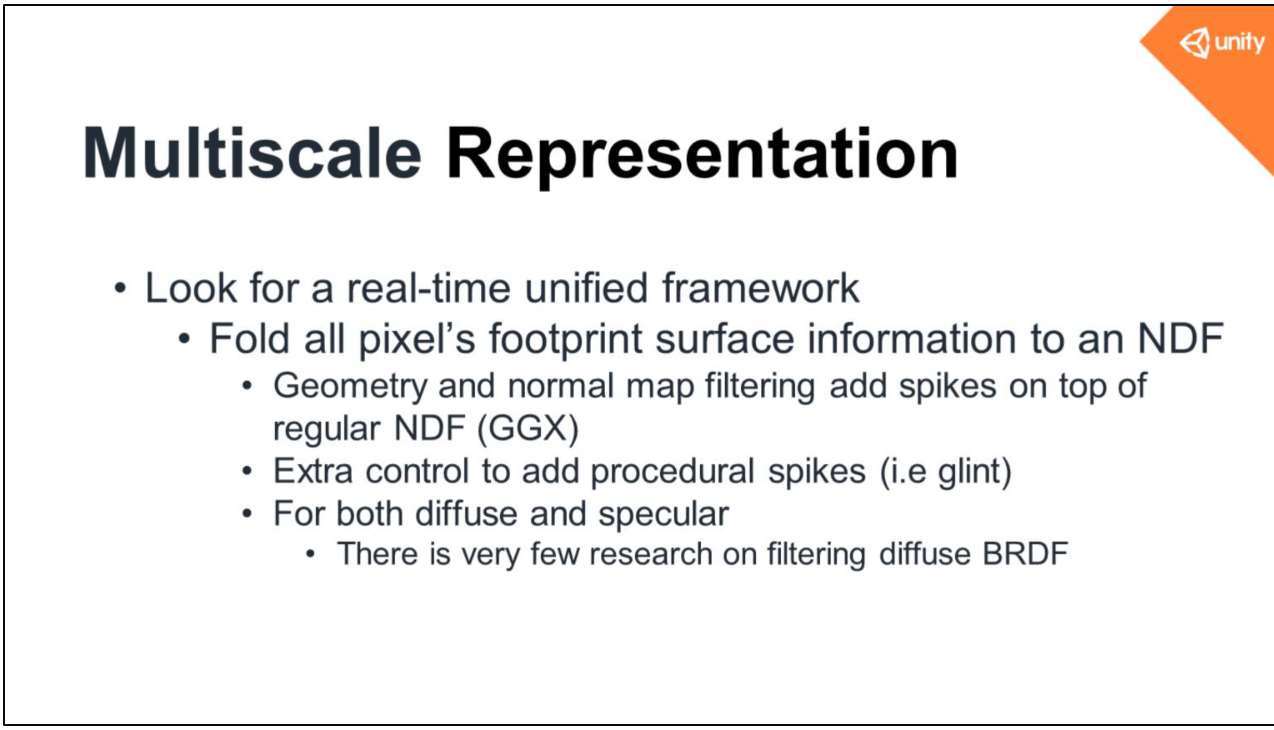
[Xu17] C. Xu, Real-Time Linear BRDF MIP-Mapping, Eurographic 2017
Not really real-time yet, but the approach of unifying BRDF and normal is good

[Becker93] B. Becker, Smooth Transitions between Bump Rendering Algorithms, Siggraph 1993
Smooth Transitions between Bump Rendering Algorithms
Already attempted Siggraph 93 ?
https://pdfs.semanticscholar.org/ab20/338c4297248cca5f32322fce6352461c2915.pdf
using a hierarchy of multiple BSDF frequency levels as well as a modification to bump mapping.



[Kaplanyan16] A. Kaplanyan, Filtering Distributions of Normals for Shading Antialiasing, HPG 2016
[Tokuyoshi17] Y. tokuyoshi, Error Reduction and Simplification for Shading Anti-Aliasing, Tech report 2017 (http://www.jp.square-enix.com/tech/library/pdf/Error%20Reduction%20and%20Simplification%20for%20Shading%20Anti-Aliasing.pdf)

Anton: We provide a practical solution applicable for real-time rendering by employing recent advances in light transport for estimating the filtering region from various effects (such as pixel footprint) directly in the parallel-plane half-vector domain (also known as the slope domain), followed by filtering the NDF over this region



Almost not research on diffuse filtering, only LEADR speak about it

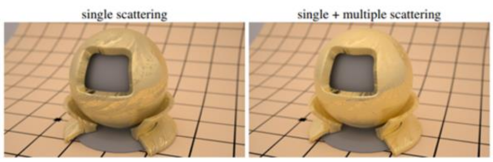
It is important to conserve the original shape of the NDF, previous paper tend to replace it by multiple Beckmann

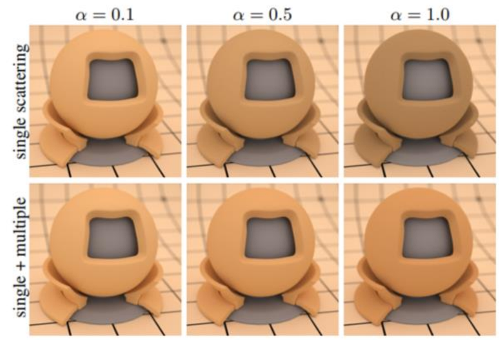
Real-time BRDF

- Going further?
  - More physically-accurate diffuse BRDF
  - NDF shape
    - Anisotropy, shape control
  - Multiscale Representation
  - Multiple scattering
  - Iridescence / Thin-film

Multi-scattering

- [Heitz16] Provide multi scattering GGX BRDF
  - Rougher is more saturated
  - For both diffuse and specular
  - Not real-time





Images from "Multiple-Scattering Microfacet BSDFs with the Smith Model", E.Heitz, Siggraph 2016

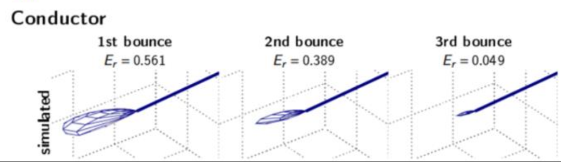
[Heitz16] E. Heitz, Multiple-Scattering Microfacet BSDFs with the Smith Model, Siggraph 2016
Will be a break changer for the artists, may require to re-author albedo.

Multi-scattering

- Real-time approximation possible ?
- [Kulla17] provides approximation of multiscattering with tabulated values
- Idea from [Heitz14]
    - Second BRDF to compensate for the lost energy

$$\int_{\Omega_i} \rho(\omega_o, \omega_i)\, |\omega_g \cdot \omega_i|\, d\omega_i = \int_{\Omega_i} \rho_1(\omega_o, \omega_i)\, |\omega_g \cdot \omega_i|\, d\omega_i + \int_{\Omega} \rho_{2^+}(\omega_o, \omega_i)\, |\omega_g \cdot \omega_i|\, d\omega_i$$

- Simulations show that bounce lobes are similar



Images from "Multiple-Scattering Microfacet BSDFs with the Smith Model", E.Heitz

[Heitz14] E. Heitz, Understanding the Masking-Shadowing Function in Microfacet-Based BRDFs, JCGT

[Kulla17] C. Kulla, "Revisiting Physically Based Shading at Imageworks", Siggraph 2017

Kulla approximation is similar approach than Kelemen et al in 2001.

Eric Heitz mentionned an approach that could be possible in real time. The bounce could be approximate by a second BRDF.
From simulation it appear the bounce lobe looks like a scaled version of the 1st bounce.
A cheap approximation could be to try to fit this scale factor (taking also into account Fresnel term) and apply it at the end of the lighting calculation (as lobe are identical).
Attempt have been done in non published work and are promising.

Real-time BRDF

- Going further?
  - More physically-accurate diffuse BRDF
  - NDF shape
    - Anisotropy, shape control
  - Multiscale Representation
  - Multiple scattering
  - Iridescence / Thin-film

Iridescence / Thin-film

- Iridescence / Thin-film
  - Chromatic Fresnel effect
    - Chromatic offset Fresnel curve [Drobot17]
    - Replace Fresnel term by Airy reflectance [Belcour17]
      - Almost practical, multiscale, lighting coherency
      - Allows a similar visual to iridescent flakes
      - Needs a simpler parametrization
  - End of story?
    - Single thin-layer, not multi-layer (like butterfly)

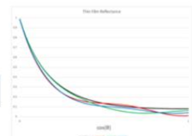


Image from “Practical multi layered rendering”, M. Drobot, Siggraph 2017

[Drobot17] M. drobot, Practical multi layered rendering, Siggraph 2017
Provide a hacked iridescence term based on shifting the Fresnel curve efficient for real time

[Belcour17] L. Belcour, A Practical Extension to Microfacet Theory for the Modeling of Varying Iridescence, Siggraph 2017
More physically based approach, need some optimization and simpler parametrization but very promising.

Layered, where are we (in real-time rendering)?

Layered BRDF

- A layered BRDF is a stack of BRDFs
  - Games use 2 specular BRDF on top of a Diffuse BRDF
  - Hardcoded roughness and Fresnel interface
    - Mimic ClearCoat
    - Simplified
- Need better rough Fresnel interface interaction

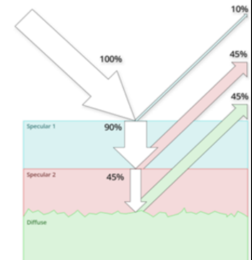


Image from “Physically Based Shader Design in Arnold”, A. Langlands, Siggraph

Layered BRDF

- Two specular layers provide lot of combination for real times. Is that enough ?

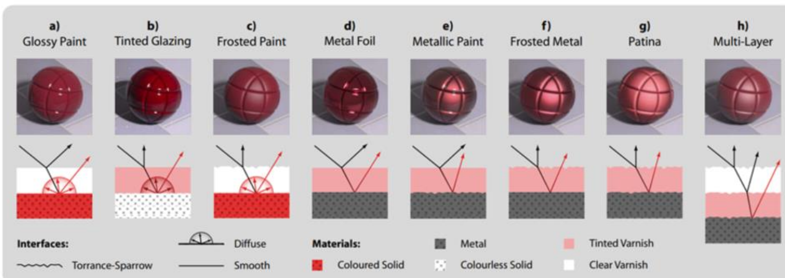


Image from: “Arbitrarily Layered Micro-Facet Surfaces”, A. Weidlich, Graphite 2007

Layered BRDF

- A hard problem with the Layered BRDF: Light transmission between layer
  - Multiple interface reflection in down/up directions
- [Jakob14] Introduced tabulated reflectance functions in a Fourier basis for offline
  - Easy to chain
  - Produce accurate results for an arbitrary number of layers

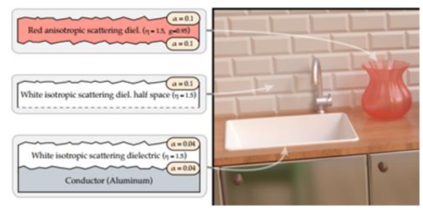
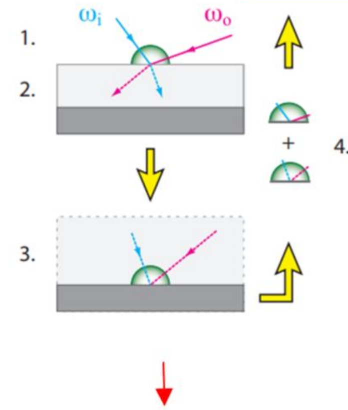


Image from "A Comprehensive Framework for Rendering Layered Materials", w. Jakob, Siggraph 2014

[Jakob14], w. Jakob, "A Comprehensive Framework for Rendering Layered Materials", Siggraph 2014

One solution suggested by Jakob et al. iis to redistribute the lost energy in a near diffuse distribution in order to not lose energy. While it seems to be a coarse approximation, their results look plausible.

Layered BRDF



- [Weidlich07] defines arbitrarily layered micro-facet surfaces
  - Specular layers only
    - Assumes single scattering thin layer
    - Approximate light transmission between layers (up and down)
      - Total internal reflection, absorption
    - No energy conservation

$$f_r = f_{r_1}(\theta_i, \theta_r) + T_{12} \cdot f_{r_2}(\theta_{i'}, \theta_{r'}) \cdot a \cdot t$$

$$a = e^{-\alpha d \cdot (\frac{1}{\theta_{i'}} + \frac{1}{\theta_{r'}})}.$$

$$t = (1 - G) + T_{21} \cdot G$$

Image from “Arbitrarily Layered Micro-Facet Surfaces”, A. Weidlich, Graphite 2007

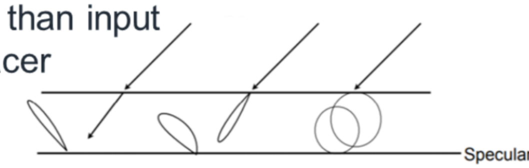
[Weidlich07], A. Weidlich, “Arbitrarily Layered Micro-Facet Surfaces”, Graphite 2007

no energy conservation, single scattering

1. The BRDF of the topmost level fr1 is evaluated for the two given, arbitrary incoming directions ωi , and ωo. This yields a reflection component, and, except at the lowest layer, two refraction directions.
2. Any energy that is refracted into the next level T12 follows the two refraction directions associated with the initial incident directions, and is partly absorbed a by the medium.
3. These two refraction directions are assumed to meet at a single point on the next layer fr2 , and the process is repeated from step 1 until an opaque layer without a refraction component is encountered.
4. On returning from the recursion, the individual BRDF components are attenuated by the Fresnel transmission coefficients T21 for the level above them, and added to the total BRDF.

Layered BRDF

- An important consideration for real-time layering
  - The top BRDF acts as a blurring filter on the bottom BRDF
    - Mean result can't be sharper than input
    - Naturally handled by path tracer



    - Real-time needs to fake it
      - Should we consider roughness of top to modify the base roughness ?

Image from "Reflection Model Design for Wall-E and Up" B. Smiths, Siggraph 2012

Layered BRDF

- Real-time layering base on [Weidlich07]
- Layered Materials in Real-Time Rendering [Elek10]
  - Simulate path tracer blurring effect in layer
    - Use min of top and base roughness
- Practical multi-layered rendering [Drobot17]
  - Improvement framework for “modern” PBR
    - Simulate path tracer blurring effect in layer
      - Use a mix of roughness, scattering and thickness

[Elek10], O. Elek, “Layered Materials in Real-Time Rendering”, 2010
[Weidlich07], A. Weidlich, “Arbitrarily Layered Micro-Facet Surfaces”, Graphite 2007
[Drobot17] M. drobot, Practical multi layered rendering, Siggraph 2017

Layered BRDF

- Previous approach approximates specular BRDF / specular BRDF interface
- What about specular BRDF / diffuse BRDF interface ?
  - Fallback to previous section question
    - [Meneveaux17] multiple reflection Fresnel term ?
      - But smooth specular only

[Meneveaux17] D. Meneveaux, Rendering Rough Opaque Materials with Interfaced Lambertian Microfacets, 2017

Remember introduction about dependency of BRDF lobes between layer

Layered BRDF

- Which parametrization for IOR?
  - Games use specular (color)
    - Deduce IOR from D65 light and air-matter interface
      - Wrong to use it for underwater for example
    - Invertible for dielectric, not for conductor
    - Many engine use fixed 1.5 value for dielectric IOR
  - Layering needs the IOR ratio
    - Problematic with conductor base layer
    - Alternate parametrization for metals exists [Gulbrandsen14]

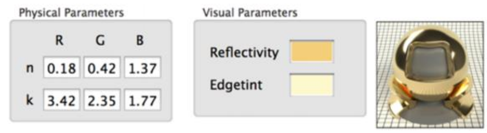


Image from "Artist Friendly Metallic Fresnel", O.Gulbrandsen, JCGT 2014

[Gulbrandsen14], O.Gulbrandsen, "Artist Friendly Metallic Fresnel", JCGT 2014

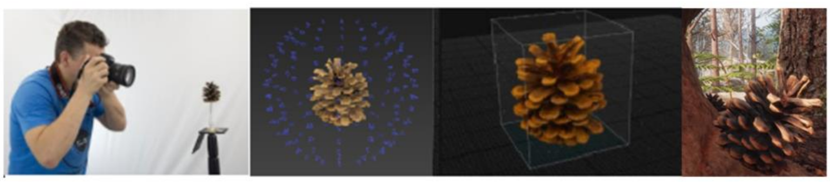
IOR: index of refraction
We can determine the IOR at an extra cost. Perhaps it would be better to store a different encoding for a dieletric IOR since currently there is no standard storage representation. Most engines simply ignore it.
Switching to Gulbrandsen solution mean we should move to forward rendering. Gbuffer storage is prohibitive.

Physically-based textures

unity

Physically-based textures

- How to acquire physically-based textures?
  - Scan the real world with photogrammetry
    - Most practical and widely adopted solution
  - Scans also capture the lighting
    - Lighting is not a part of physically-based textures



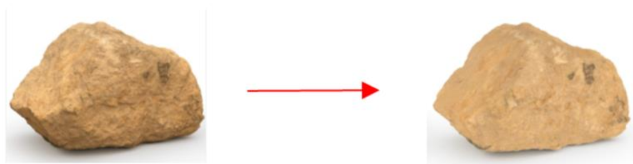
Physically-based textures

- How to extract BRDF parameters from a scan of a lit material?
  - Want individual BRDF parameters
    - Unlit diffuse albedo
    - Roughness
  - Open problem

Reconstruct unlit diffuse albedo ?

- Goal: retrieve object's illumination at the time of capture

- Use this information to remove lighting from the texture



Removing illumination is simply dividing the lit material by the recovered light. We assume Lambertian surface.

Capture Object's illumination?

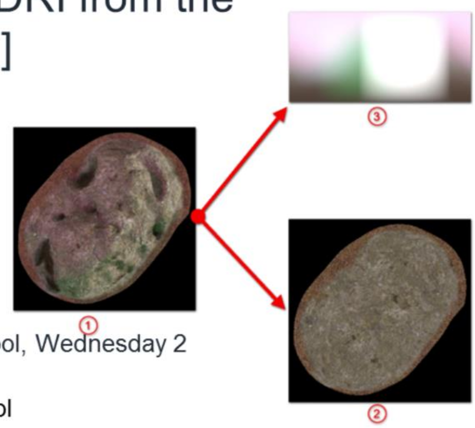
- Use an HDRI from capture location [Antoine15]
  - HDRI position matters for big objects
    - Take multiple HDRI?
  - Outdoor lighting condition changes quickly
  - Time consuming

Image from "The Tech & art behind Epic's UE4 Open World Demo", F. Antoine, GDC 2015

[Antoine15] F. Antoine, "The Tech & art behind Epic's UE4 Open World Demo", GDC 2015

Capture Object's illumination?

- Reconstruct a pre-convolved HDRI from the reconstructed lit mesh [Jover17]
  - Fast, no extra input
  - Works with large objects
  - Looks promising!



Photogrammetry Workflow and the tech behind the delighting tool, Wednesday 2 August - Unity Central Room - 503 - 11:00am

https://labs.unity.com/article/experimental-feature-de-lighting-tool

Please test our Unity De-lighting tool and tell us what you think :)

Capture Object's illumination?

- Throwing ideas inspire by [Duchene15]
  - Use photogrammetry pictures
    - Reconstruct HDRI from a set of pictures?
    - Remove lighting from pictures before projecting onto the reconstructed mesh?

(a) Input image (c) Reflectance (d) Shading

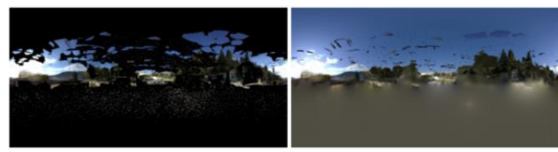


Image from "Multi-View Intrinsic Images of Outdoors Scenes with an Application to Relighting", S. Duchene, 2015

[Duchene15] S. Duchene, "Multi-View Intrinsic Images of Outdoors Scenes with an Application to Relighting", 2015
They split lighting and reflectance using a multiple view picture set

Retrieve roughness?

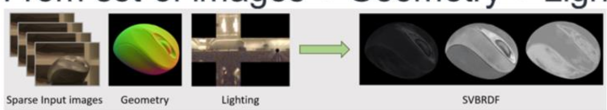
- Unity experiment
  - Take samples at varying elevation angles with aligned flash & camera
    - Based on a modified version of [Dupuy15]
  - Seems to work, but not practical
  - 24H processing time for 4K textures…

Smoothness map

[Dupuy15] J. Dupuy, Extracting Microfacet-based BRDF Parameters from Arbitrary Materials with Power Iterations, EGSR 15
(L == V, mean backscatter)
Requires a normal map
Direct evaluation (no optimization problem)
Brute force processing that says which roughness match a particular backscatter intensity
For a pixel in the smoothness map, we will take all the angles, check into a table, use the algorithm from the paper, then extract roughness
It use a set of images at different angle. For grazing angle, our artists have manually align the pixles

Retrieve roughness?

- Promising research available
  - Sparse-as-Possible SVBRDF Acquisition[Zhou16]
    - From set of images + Geometry + Lighting



  - Two-Shot SVBRDF Capture for Materials [Aittala15]
    - Could be practical
- Key point: Should be easy for artists to experiment

Image from "Sparse-as-Possible SVBRDF Acquisition", Z. Zhou, Siggraph Asia 2016

[Aittala15] M. Aittala, "Two-Shot SVBRDF Capture for Stationary Materials", Siggraph 2015
This paper take set of images, geometry and HDRI as input (i.e what you have with photogrammetry) and extract the BRDF parameters
[Zhou16] Z. Zhou, "Sparse-as-Possible SVBRDF Acquisition", Siggraph Asia 2016
This paper use two inputs: with and without flash pictures of a material and provide the material BRDF parameters.

The problem with this kind of papers is that there is no way for an artists to experiment it. Implementing a paper like this is extremely complex and even where there is matlab source (like for Aittala), there is no way an artists can make anything with that. It require at least a command line program.

Machine learning?

- Generating virtual data is not always simple
  - Generate virtual lit/unlit data: OK. Roughness?
    - Need to compute normal distribution of the mesh and fit it to an NDF to recover roughness parameters ?
      - Mean very dense mesh
  - Getting representative assets is very challenging
- Few research available
  - Reflectance Modeling by Neural Texture Synthesis [Aittala16]

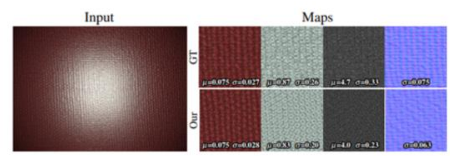


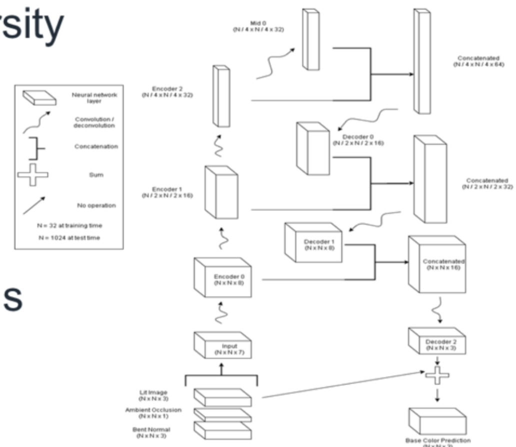
Image from “Reflectance Modeling by Neural Texture Synthesis”, M. Aittala, siggraph 2016

Obvious trend: Try deep learning? When you don’t know what to do.
The problem with deep learning is first to find the data. And finding a large amount of “correct” data, is often not possible. Better to rely on generation. But then, how to generate representative roughness map ?

[Aittala16], Reflectance Modeling by Neural Texture Synthesis, Siggraph 2016
Use a single image with Flash to recover texture

Machine learning?

- Research effort at Unity for unlit case
- Partnership with Berkeley University
  - Based only on texture data
    - Inputs: Lit diffuse / bent normal map / ambient occlusion
    - Outputs: Unlit diffuse
  - Try various input combinations
    - Works best for lit diffuse only...



Project Managers: Michael Zhang, Zhongxia “Zee” Yan
Members: Murtaza Dalal, Quinn Tran, Varsha Ramakrishnan, Tracy Lou, Gefen Kohavi

Training time: 32x32 slice
Test time: 1024x1024

Machine learning?

- Results: blurry, but promising. Further research is needed.

lighted delighted predicted

Project Managers: Michael Zhang, Zhongxia “Zee” Yan
Members: Murtaza Dalal, Quinn Tran, Varsha Ramakrishnan, Tracy Lou, Gefen Kohavi

Future may be a mix of artists tools and deep learning

Conclusion

Conclusion

- What could be a future general physically-based material for real time?
  - Layered BRDF: 2 specular BRDF + Diffuse BRDF
    - All derives from the same anisotropic NDF
    - Energy conserving: MultiScattering, Fresnel interfaces
  - Option to switch to Airy reflectance Fresnel
  - Shape-invariant “matching measure” NDF
  - Multiscale Diffuse and Specular representation
- Require to move to forward+ with all parameters ?

Conclusion

- And need to capture physically-based textures for such a material based on a photogrammetry workflow
- Physically-based materials are hard!
  - And we've only scratched the surface with BRDFs
    - Subsurface scattering, volume, transparency even more complex
    - Diffraction ?
  - Not the only part of the equation
    - Physically-based lighting even more important

Acknowledgements

- Aaron Lefohn, Natalya Tatarchuk
- Naty Hoffman, Eric Heitz, Jonathan Dupuy, Laurent Belcour, Evgenii Golubev, Charles de Rousiers, Brian Karis, Sébastien Hillaire, Emmanuel Turquin, Cyril Jover, Sébastien Lachambre: Discussion and help


Unity is hiring
careers.unity.com

Questions ?

Lagardese@hotmail.fr
Twitter: @seblagarde

References

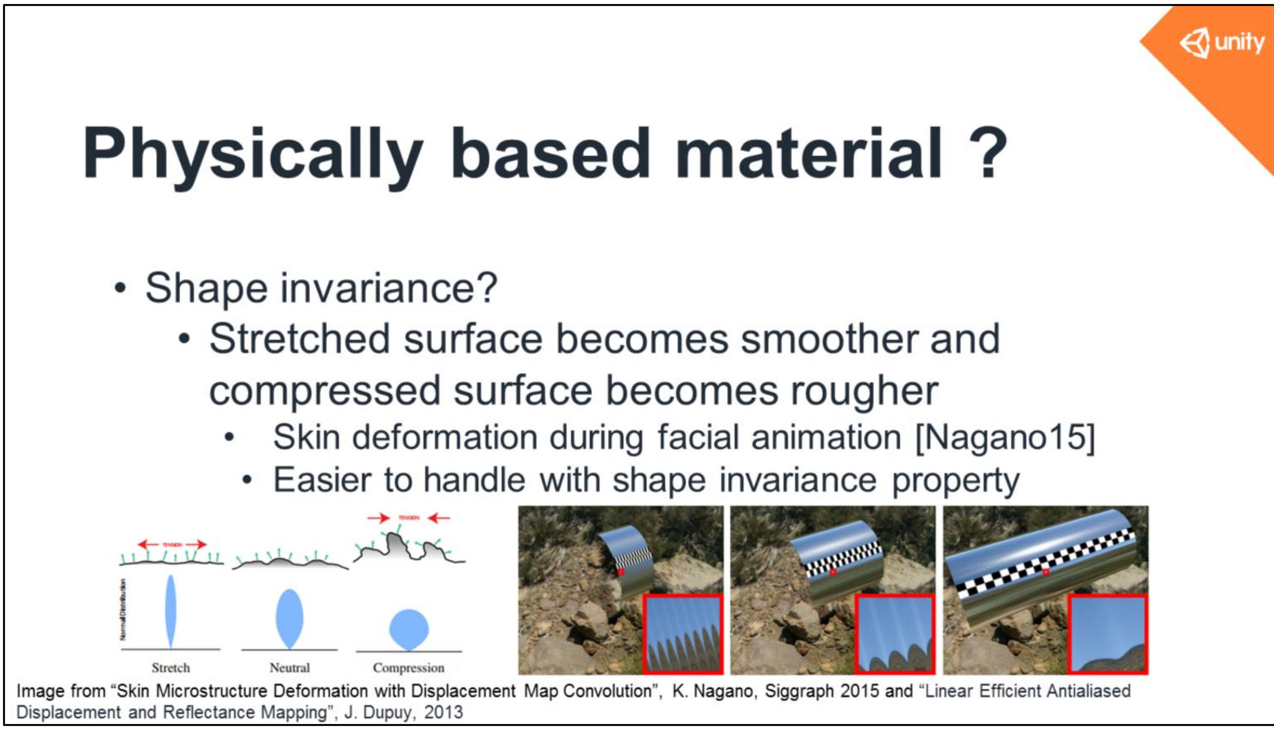
[Aittala15] M. Aittala, "Two-Shot SVBRDF Capture for Stationary Materials", Siggraph 2015
[Aittala16], Reflectance Modeling by Neural Texture Synthesis, Siggraph 2016
[Antoine15] F. Antoine, "The Tech & art behind Epic's UE4 Open World Demo", GDC 2015
[Atanasov15] A. Atanasov, A Practical Stochastic Algorithm for Rendering Mirror-Like Flakes, Chaos group documentation 2015
[Bagher12] M. Bagher, "Accurate fitting of measured reflectances using a Shifted Gamma micro-facet distribution", 2012
[Becker93] B. Becker, Smooth Transitions between Bump Rendering Algorithms, Siggraph 1993
[Belcour17] L. Belcour, A Practical Extension to Microfacet Theory for the Modeling of Varying Iridescence, Siggraph 2017
[Burley12] B. Burley, Physically based shading at Disney, Siggraph 2012
[Dimov15] R.Dimov, Deriving the Smith shadowing function G1 for γ in (0, 4], Chaos group documentation 2015
[Dupuy13] J. Dupuy, E. Heitz, Linear Efficient Antialiased Displacement and Reflectance Mapping, 2013
[Dupuy15] J. Dupuy, Extracting Microfacet-based BRDF Parameters from Arbitrary Materials with Power Iterations, EGSR 2015
[Drobot17] M. drobot, Practical multi layered rendering, Siggraph 2017
[Elek10], O. Elek, "Layered Materials in Real-Time Rendering", 2010
[Fujii] Y. Fujii, A A tiny improvement of Oren-Nayar reflectance model
[Gotanda12] Y. Gotanda, Beyond a Simple Physically Based Blinn-Phong Model in Real-Time, Siggraph 2012
[Gotanda14] U. Gotanda, Designing Reflectance Models for New Consoles, Siggraph 2014
[Gotanda15] Y. Gotanda, Designing Physically Based Microfacet Models for Next Generation, Cedec 2015
[Gulbrandsen14], O.Gulbrandsen, "Artist Friendly Metallic Fresnel", JCGT 2014
[Hammon17] E. Hammon, PBR Diffuse Lighting for GGX+Smith Microsurfaces, GDC 2017
[Heitz14] E. Heitz, Understanding the Masking-Shadowing Function in Microfacet-Based BRDFs, JCGT
[Heitz15] E. Heitz, Implementing a Simple Anisotropic Rough Diffuse Material with Stochastic Evaluation, 2015
[Heitz16] E. Heitz, Multiple-Scattering Microfacet BSDFs with the Smith Model, Siggraph 2016
[Heitz16b] E. Heitz, "Real-time polygonal-light shading with linearly transformed cosines", Siggraph 2016
[Holzschuch17] N. Holzschuch, A Two-Scale Microfacet Reflectance Model Combining Reflection and Diffraction, Siggraph 17

References

[Jakob14] W. Jakob, Discrete Stochastic Microfacet Models, Siggraph 2014
[Jakob14b], W. Jakob, "A Comprehensive Framework for Rendering Layered Materials", Siggraph 2014
[Jover17] C. Jover, "The Tech behind the Unity De-Lighting tool", 2017
[Kaplanyan16] A. Kaplanyan, Filtering Distributions of Normals for Shading Antialiasing, HPG 2016
[Karis13] B. Karis, "Real Shading in Unreal Engine 4", Siggraph 2013
[Lagarde14] S. Lagarde, C. De Rousiers, Moving Frostbite to Physically based Rendering, Siggraph 2014
[McAuley15] S. McAuley, The rendering of far cry 4, Cedec 2015
[McGuire16], M. McGuire, Peering Through a Glass, Darkly at the Future of Real-Time Transparency, Siggraph 2016
[Meneveaux17] D. Meneveaux, Rendering Rough Opaque Materials with Interfaced Lambertian Microfacets, 2017
[MERL06] https://www.merl.com/brdf/
[Nagano15] K. Nagano, Skin Microstructure Deformation with Displacement Map Convolution, Siggraph 2015
[Olano10] M.Olano, D.Baker, Lean Mapping, I3D 2010
[Oren94], M. Oren, S. Nayar, Generalization of Lambert's Reflectance Model, Siggraph 1994
[Pesce15] A. Pesce, M. Iwanicki, Approximate Models For Physically Based Rendering, Siggraph 2015
[Raymond16], B. Raymond, Multi-Scale Rendering of Scratched Materials using a Structured SV-BRDF Model, Siggraph 16
[Revie11] D. Revie, Implementing Fur Using Deferred Shading, GPU Pro 2
[Ribardiere17] STD: Student's t-Distribution of Slopes for Microfacet Based BSDFs, eurographics 2017
[Shirley97] P. Shirley, A Practitioners' Assessment of Light Reflection Models, 1995 Equation (5)
[Toksvig04] M. toksvig, Mipmapping Normal Maps, 2014
[Tokuyoshi17] Y. tokuyoshi, Error Reduction and Simplification for Shading Anti-Aliasing, Tech report 2017
[Xu17] C. Xu, Real-Time Linear BRDF MIP-Mapping, Eurographic 2017
[Yan16], L. Yan, Position-Normal Distributions for Efficient Rendering of Specular Microstructure, Siggraph 2016
[Zirr] T. Zirr, Real-time Rendering of Procedural Multiscale Materials, I3D 2016
[Zhou16] Z. Zhou, "Sparse-as-Possible SVBRDF Acquisition", Siggraph Asia 2016

Bonus slides

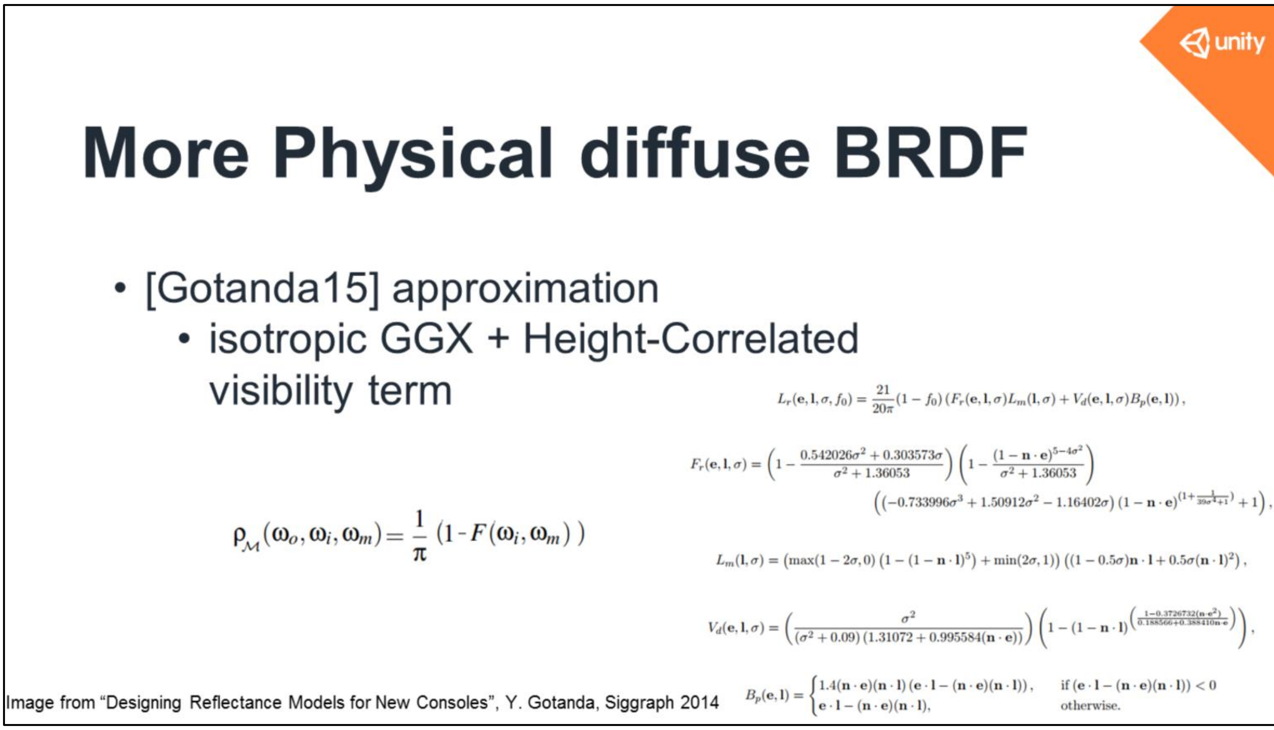
Lagardese@hotmail.fr
Twitter: @seblagarde



[Nagano15] K. Nagano, Skin Microstructure Deformation with Displacement Map Convolution, Siggraph 2015

Since the skin surface is relatively stiff, it develops a rough microstructure to effectively store a reserve of surface area to prevent rupturing when extended. Thus, parts of the skin which stretch and compress significantly (such as the forehead and around the eyes) are typically rougher than parts which are mostly static, such as the tip of the nose or the top of the head. When skin stretches, the microstructure flattens out and the surface appears less rough as the reserves of tissue are called into action. Under compression, the microstructure bunches up, creating micro-furrows which exhibit anisotropic roughness. Often, stretching in one dimension is accompanied by compression in the perpendicular direction to maintain the area of the surface or the volume of tissues below. A balloon provides a clear example of roughness changes under deformation: the surface is diffuse at first, and becomes shiny when inflated

https://spiral.imperial.ac.uk/bitstream/10044/1/23615/2/skinstretch-final-small.pdf



[Gotanda15] Y. Gotanda, Designing Physically Based Microfacet Models for Next Generation, Cedec 2015

Note the energy conserving term used (Left), the resulting 21/20 PI (1 - fo) seems a convenient approximation

Energy conservation

- Energy conservation between diffuse and specular?
  - Several conceptual approach exist
    - Mean different results
    - Let's see two of them
  - Multi-layered and interfaced Lambertian microfacets

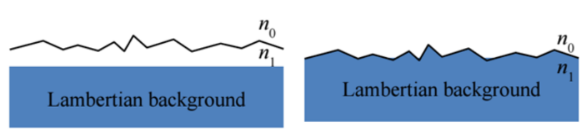
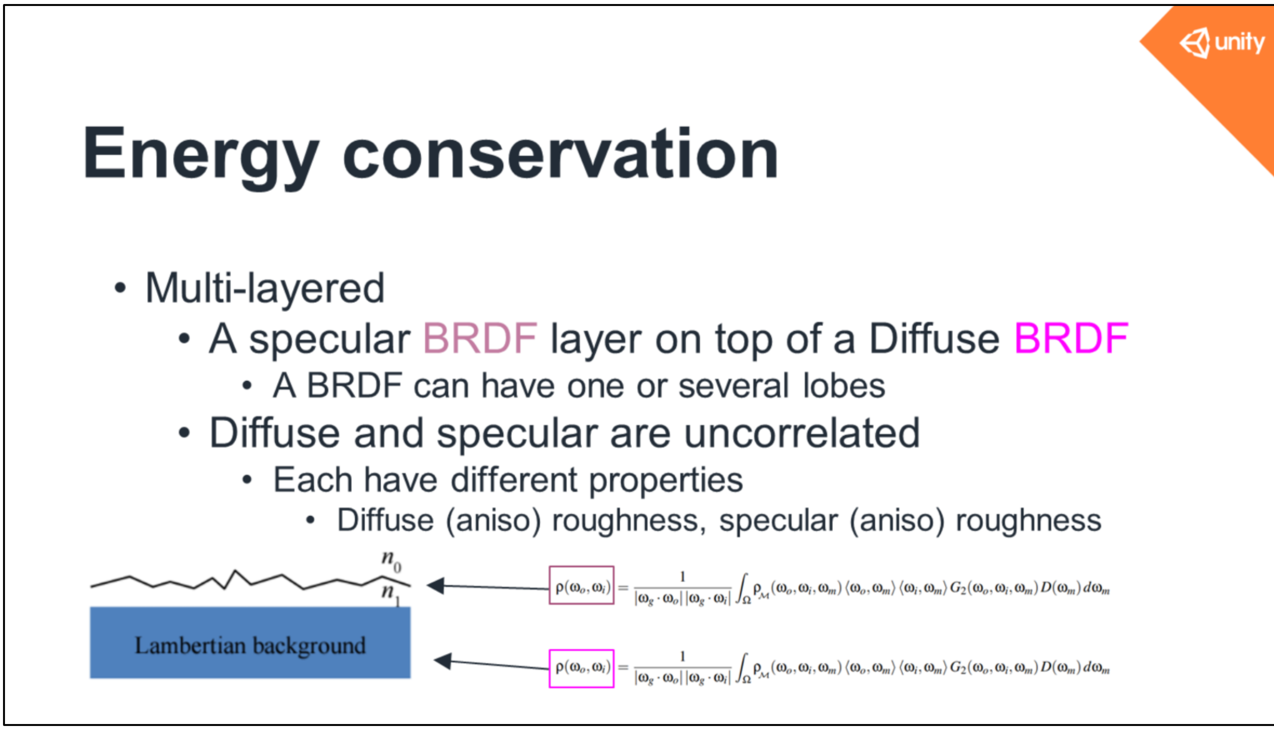


Image from "Rendering Rough Opaque Materials with Interfaced Lambertian Microfacets, D. Meneveaux, 2017

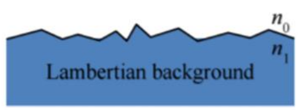


Diffuse and specular can be perceived as an aggregation of BRDF

For example, Jakob et al. suggest to reintroduce the energy loss as a diffuse radiation in reflection and transmission so that the energy is conserved. W

Energy conservation

- Interfaced Lambertian microfacets
  - Uber BRDF: Diffuse + Specular
  - Diffuse and specular are correlated
    - Share properties
      - One (anisotropic) roughness



$$\rho(\omega_o, \omega_i) = \frac{1}{|\omega_g \cdot \omega_o| \, |\omega_g \cdot \omega_i|} \int_\Omega \rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) \langle \omega_o, \omega_m \rangle \langle \omega_i, \omega_m \rangle G_2(\omega_o, \omega_i, \omega_m) D(\omega_m) \, d\omega_m$$

Energy conservation

- What is the difference ?
  - Comparison done by [Meneveux17]
    - As roughness increases
      - Interfaced BRDF exhibits backscattering (flat look)
      - Layered BRDF has a Lambertian behavior

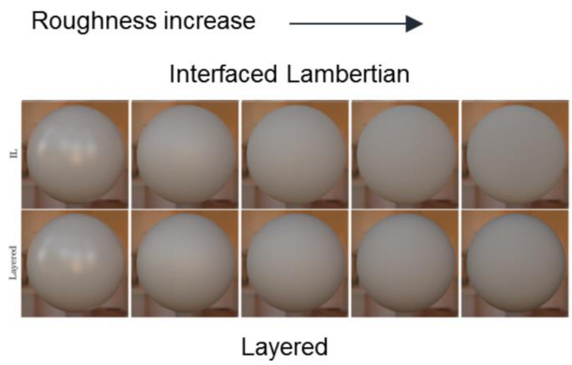


Image from “Rendering Rough Opaque Materials with Interfaced Lambertian Microfacets, D. Meneveaux, 2017

[Meneveaux17] D. Meneveaux, Rendering Rough Opaque Materials with Interfaced Lambertian Microfacets, 2017 (in supplemental)

Energy conservation

- Diffuse BRDF - Facet BRDF $\rho_{\mathcal{M}}(\omega_o, \omega_i, \omega_m) = \frac{1}{\pi}$

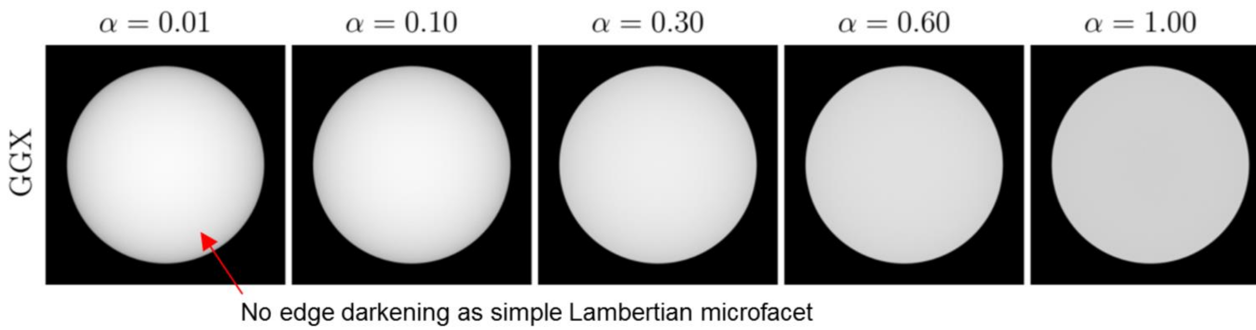


Image from "Implementing a Simple Anisotropic Rough Diffuse Material with Stochastic Evaluation", E. Heitz, 2015

As you can see, when defining a diffuse BRDF uncorrelated to specular BRDF it doesn't include darkening at edge. The energy conservation will come from layer interface.

Physically based material ?

- So far, we have discussed vertical layering only
- But what about horizontal layering (i.e. mixing BRDFs)?
  - With distance BRDFs are mixed in pixel's footprint
  - Simple: evaluate all BRDFs and weight the results
    - Blending parameters is wrong
    - Very hard in practice in real time
      - Even more complex than multiscale BRDFs
        - Fresnel can change as well
      - Could be approximated by adding constraints?

Multiscale

- Non smooth NDF ?
  - Glint/Spiky BRDF
  - Some materials exhibit a bright sparkling or glittering surface
    - Sand, snow or frost (ice crystals)
    - Rock (mica-flakes), metallic paints (mirror-flakes)
    - Brushed and scratched metal…

Images from “How to design your assets for physically based rendering”, Y. Gotanda, Cedec 2012

Due to bumps and flakes

Layered BRDF

- Weidlich and Wilkie's approach is compatible with the general microfacet equation
  - Just add one layer [Gotanda15]

$$f_r(\boldsymbol{x},\boldsymbol{l},\boldsymbol{e}) = \frac{1}{4(\boldsymbol{l}\cdot\boldsymbol{n})(\boldsymbol{e}\cdot\boldsymbol{n})}\int_{\Omega} F_m(\boldsymbol{e},\boldsymbol{m},\eta_0)\delta(\boldsymbol{l},\boldsymbol{e},\boldsymbol{m})\,G(\boldsymbol{e},\boldsymbol{m})D(\boldsymbol{m})d\boldsymbol{m}$$

$$+f_{trans}(\boldsymbol{l},\boldsymbol{e})\left\{\frac{1}{\boldsymbol{4}(\boldsymbol{l}\cdot\boldsymbol{n})(\boldsymbol{e}\cdot\boldsymbol{n})}\int_{\Omega} F_m\left(\boldsymbol{e}',\boldsymbol{m},\frac{\eta_1}{\eta_0}\right)\delta(\boldsymbol{l}',\boldsymbol{e}',\boldsymbol{m})\,G(\boldsymbol{e}',\boldsymbol{m})D(\boldsymbol{m})d\boldsymbol{m} + \frac{1}{\pi}\right\}$$

Image from "Designing Physically Based Microfacet Models for Next Generation", Y. Gotanda, Cedec 2015

[Gotanda15] Y. Gotanda, Designing Physically Based Microfacet Models for Next Generation, Cedec 2015

Note: Gotanda here don't use GGX diffuse but just 1 / PI