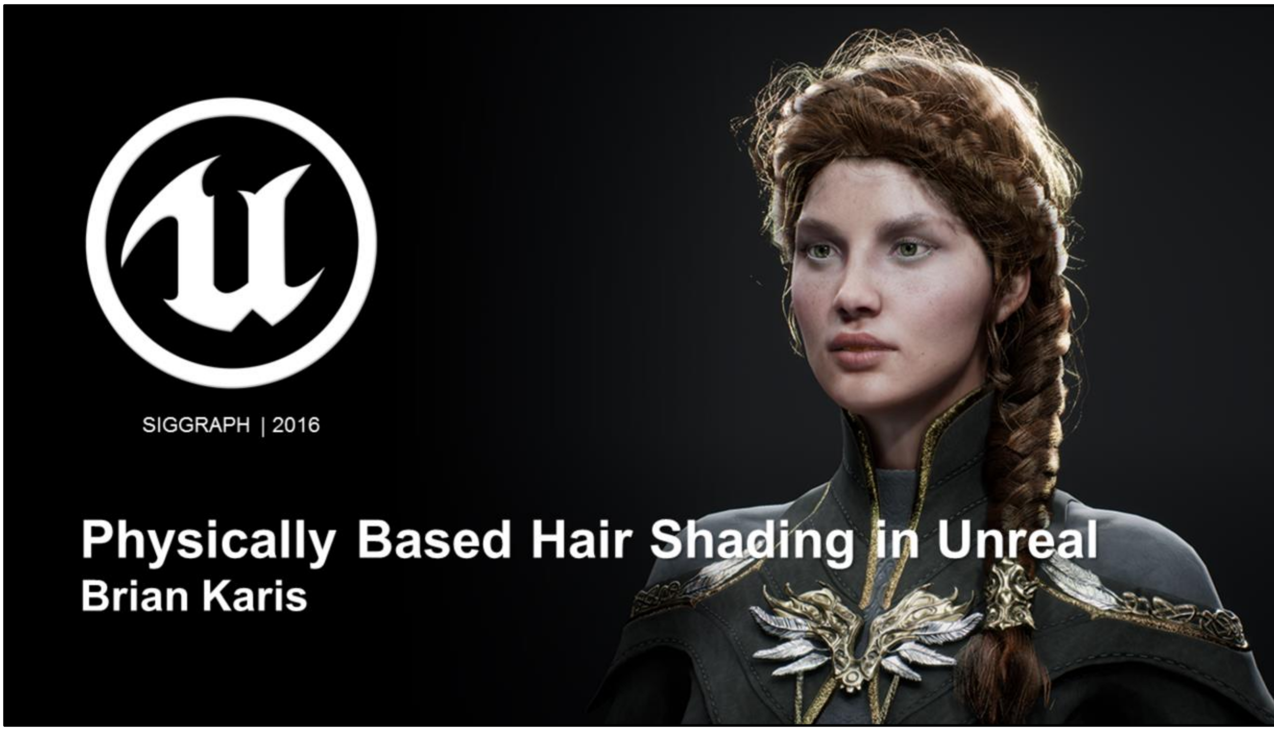
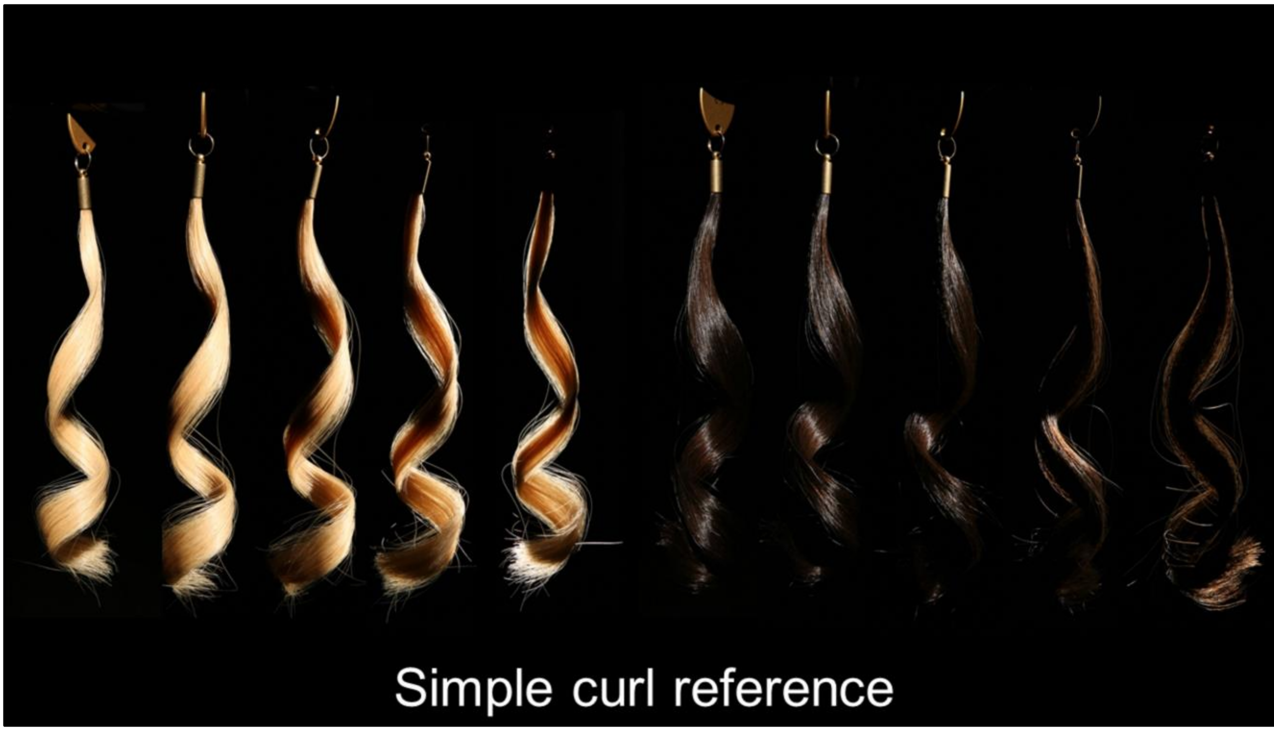

SIGGRAPH | 2016
Physically Based Hair Shading in Unreal
Brian Karis

A physically based shading model for hair

- The motivation of this work:
  - Create a physically based shading model for hair usable for games
- Tons of great advancements recently in real-time hair rendering
  - Translucency
  - Antialiasing
  - Simulation
- Most everyone in games still using the Kajiya-Kay shading model
  - Film and academia have pushed much further
  - Let's bridge this gap!

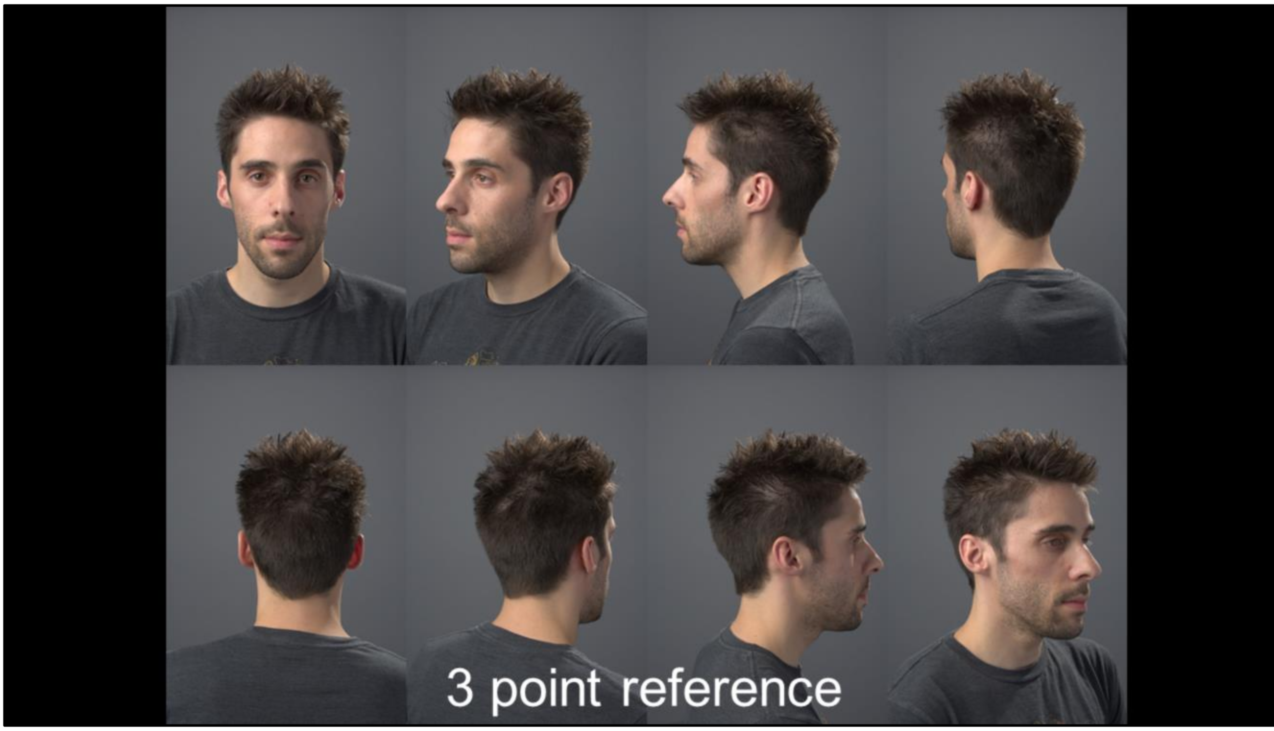
[Hubert et al. 2005]

There have been a ton of great advancements in hair rendering recently but not really in the shading realm.



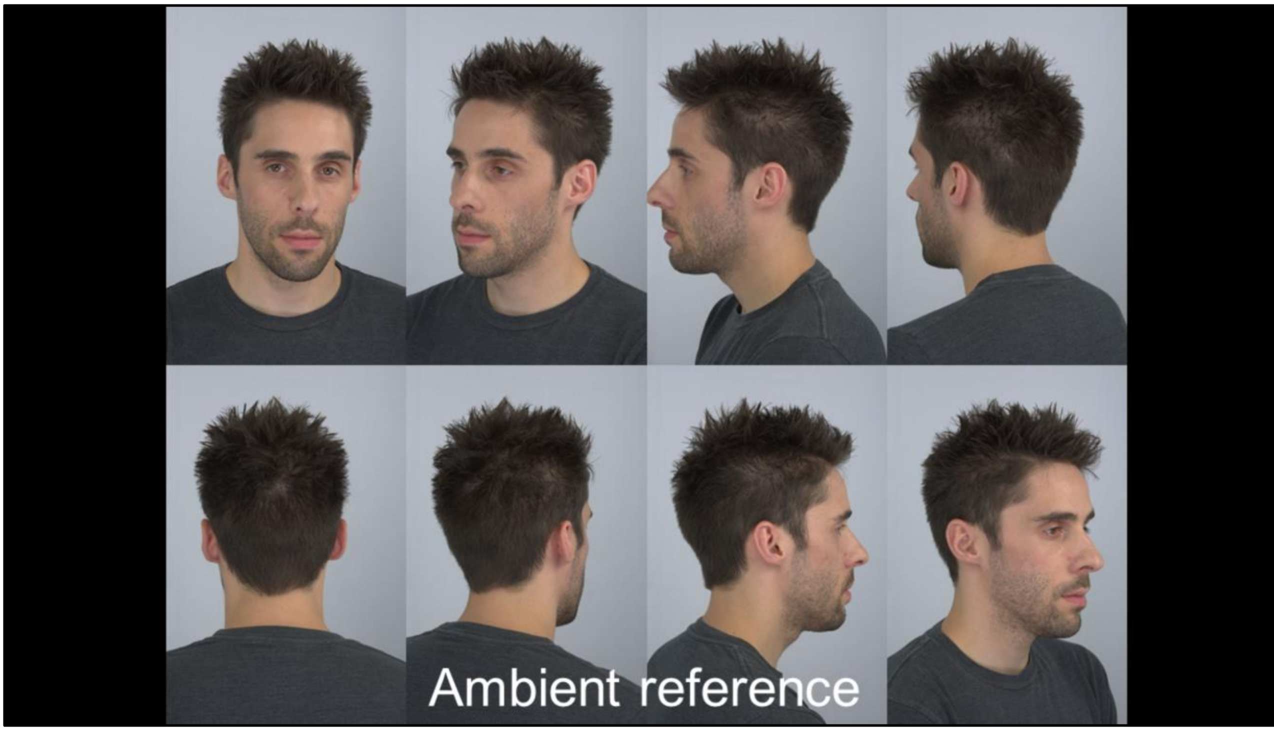
Before we start any work like this we like to get a ton of photo reference to compare our results to.

These are real human hair extensions to get us a wide variety of colors

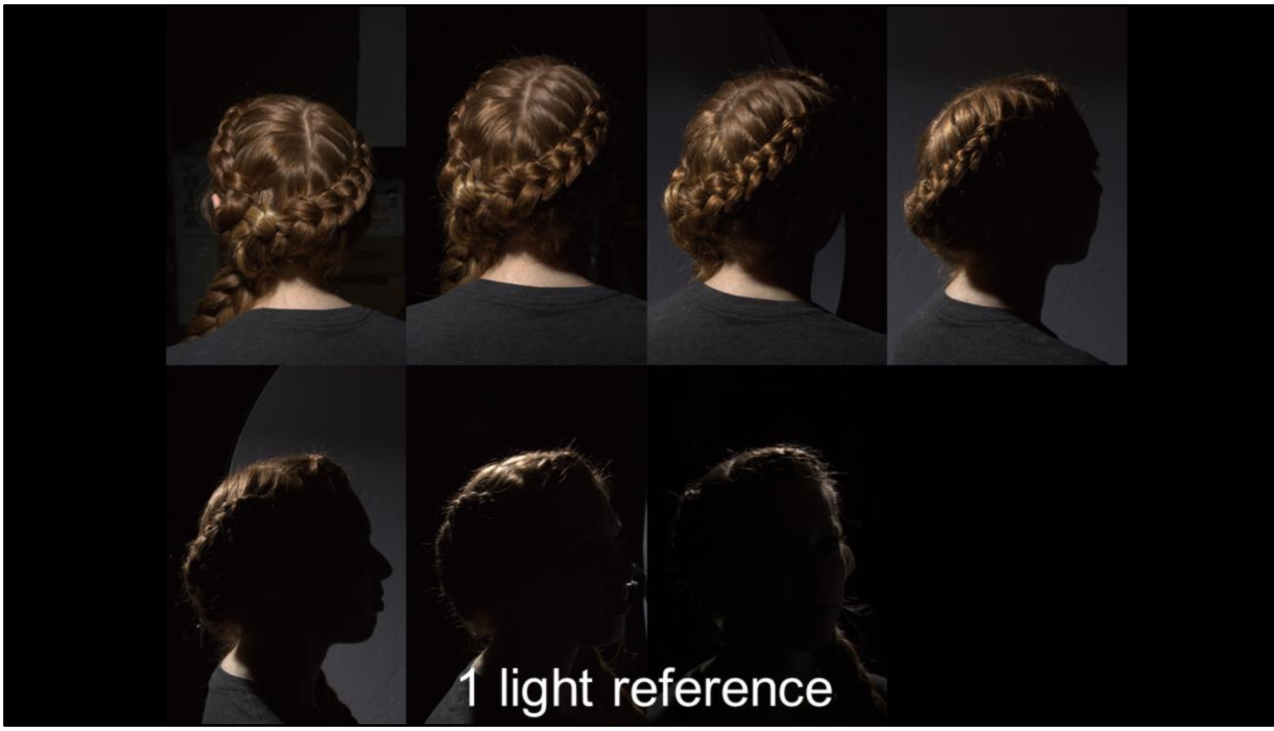


We also brought in a professional stylist to style some peoples hair like those of some of the characters in Paragon.

This is a 3 point lighting set up for the character twin blast



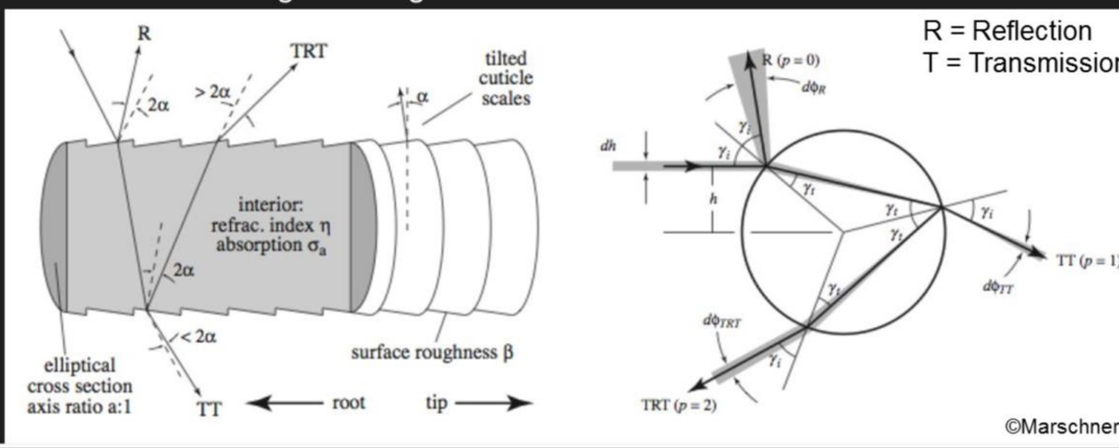
This is ambient lighting



This is a single light for the character Sparrow

Hair shading

- Based on Marschner's light path breakdown for hair fibers
- Models shading of a single hair fiber

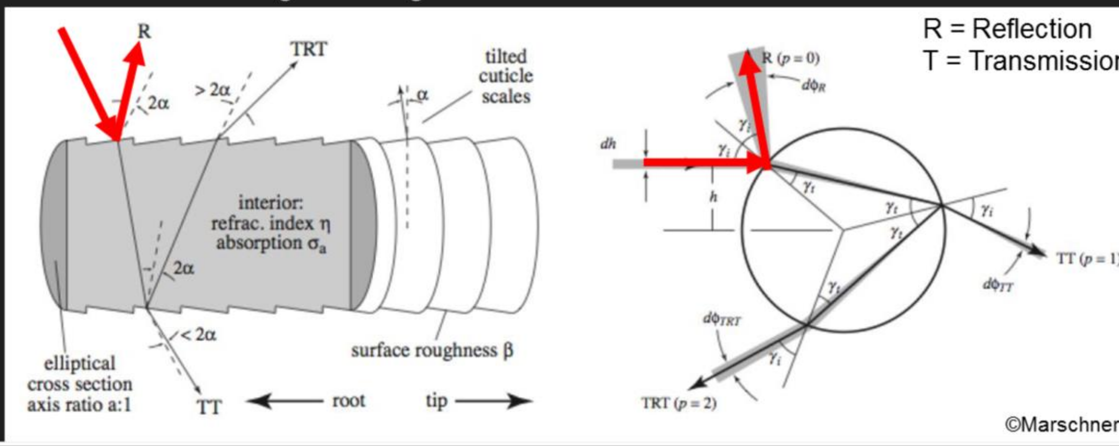


#UE4 | @UNREALENGINE [Marschner et al. 2003] UNREALENGINE.COM

The paths are named by the R for reflection and T for transmission events that the path encounters.

Hair shading

- Based on Marschner's light path breakdown for hair fibers
- Models shading of a single hair fiber

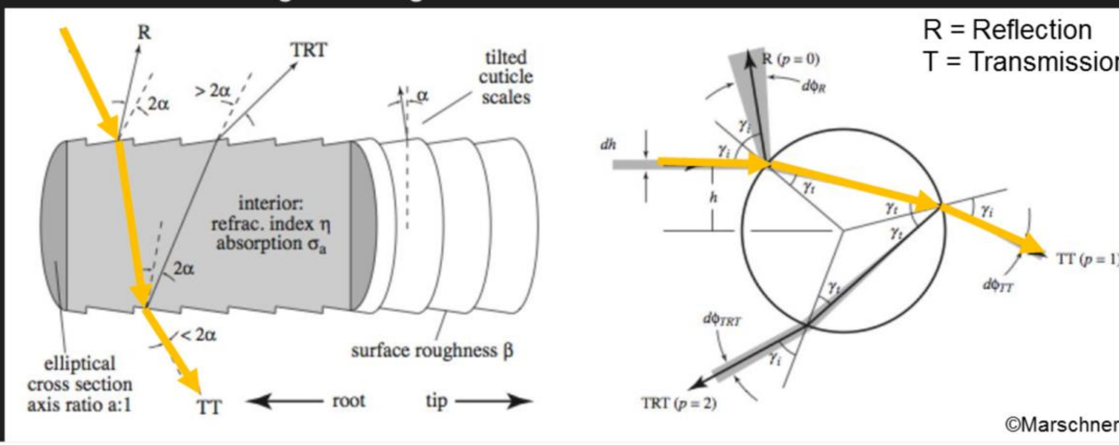


#UE4 | @UNREALENGINE [Marschner et al. 2003] UNREALENGINE.COM

This is the R path.
The light ray reflects off the front of the hair fiber

Hair shading

- Based on Marschner's light path breakdown for hair fibers
- Models shading of a single hair fiber

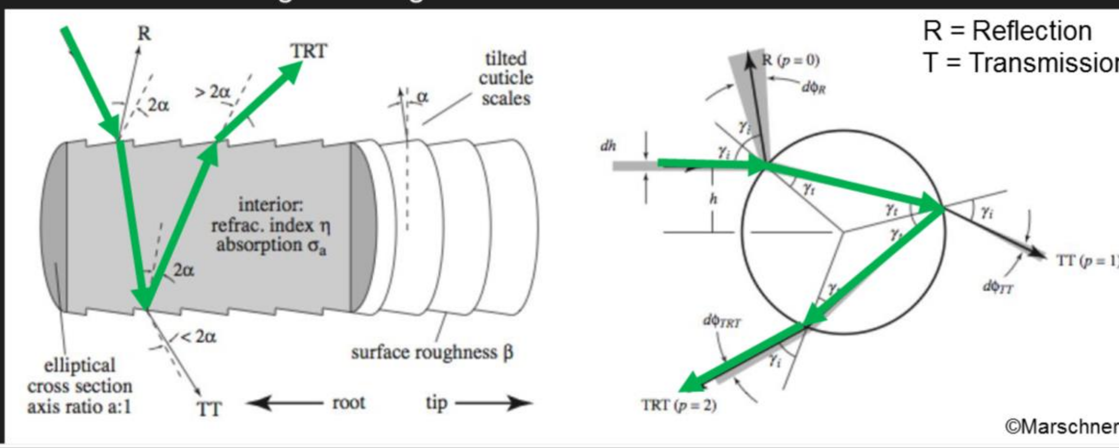


[Marschner et al. 2003] UNREALENGINE.COM

This is the TT path.
The light ray transmits through the front of the fiber,
passes through the interior which colors it due to absorption
And then transmits through the other side

Hair shading

- Based on Marschner's light path breakdown for hair fibers
- Models shading of a single hair fiber

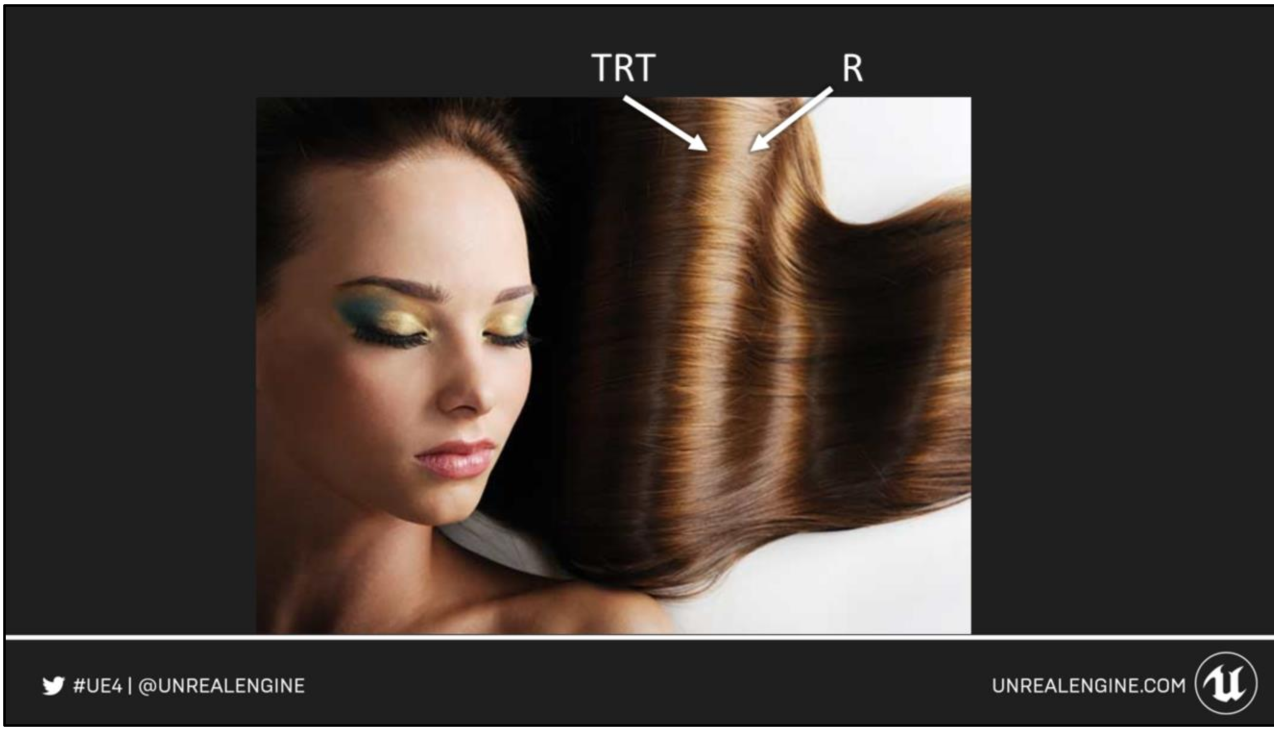


#UE4 | @UNREALENGINE

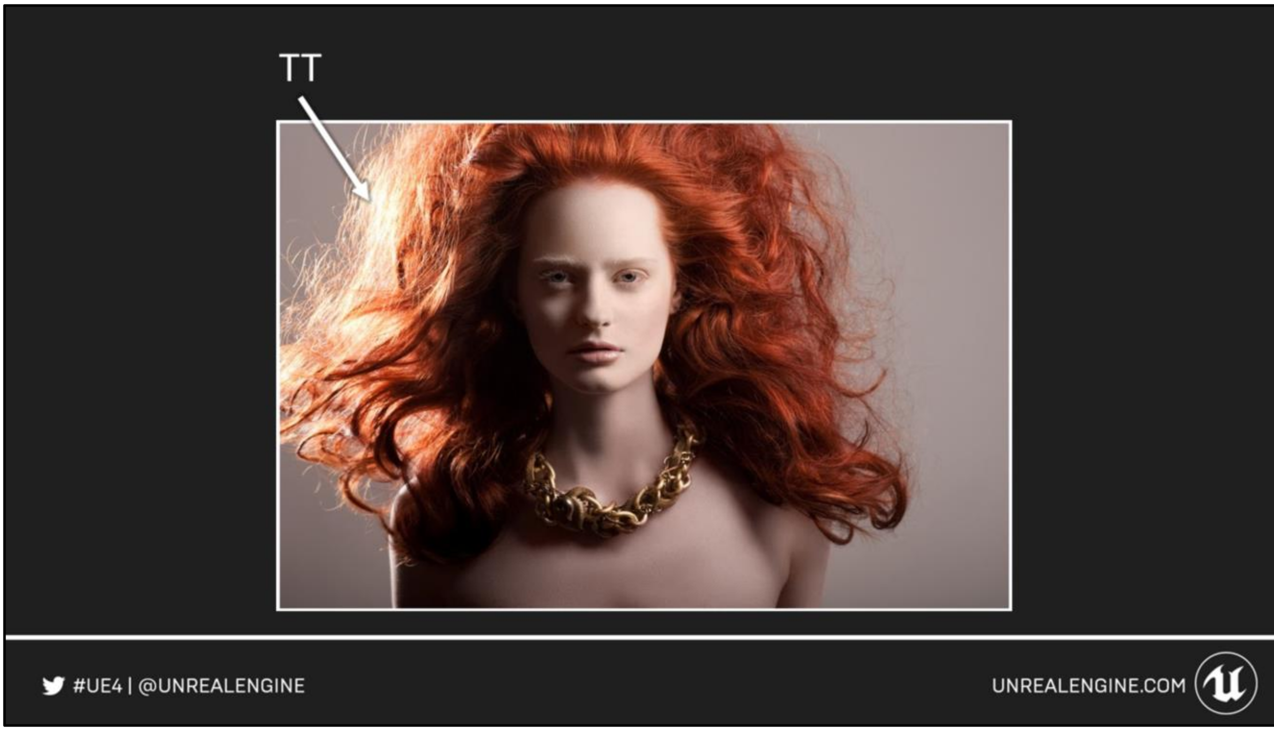
[Marschner et al. 2003] UNREALENGINE.COM

This is the TRT path.
The light ray transmits through the front of the fiber
Passes through the colored interior
Reflects off the opposite side,
Passes through the interior again and transmits out the front.

We could keep going with more internal bounces but these typically are pretty dim due to lots of absorption.



The R path is white due to not passing through the interior of the fiber
The TRT is colored due to passing through the fiber twice.
They are separated from one another due to the tilted cuticle scales



The TT path can be seen from the other side.
It is also colored since the path passed through the interior of the fiber but is less saturated than the TRT since it only passes through once.

Marschner's factorization

Sum of lobes:

$$S(\theta_i, \theta_r, \phi) = \sum_{p=0}^{\infty} S_p(\theta_i, \theta_r, \phi)$$

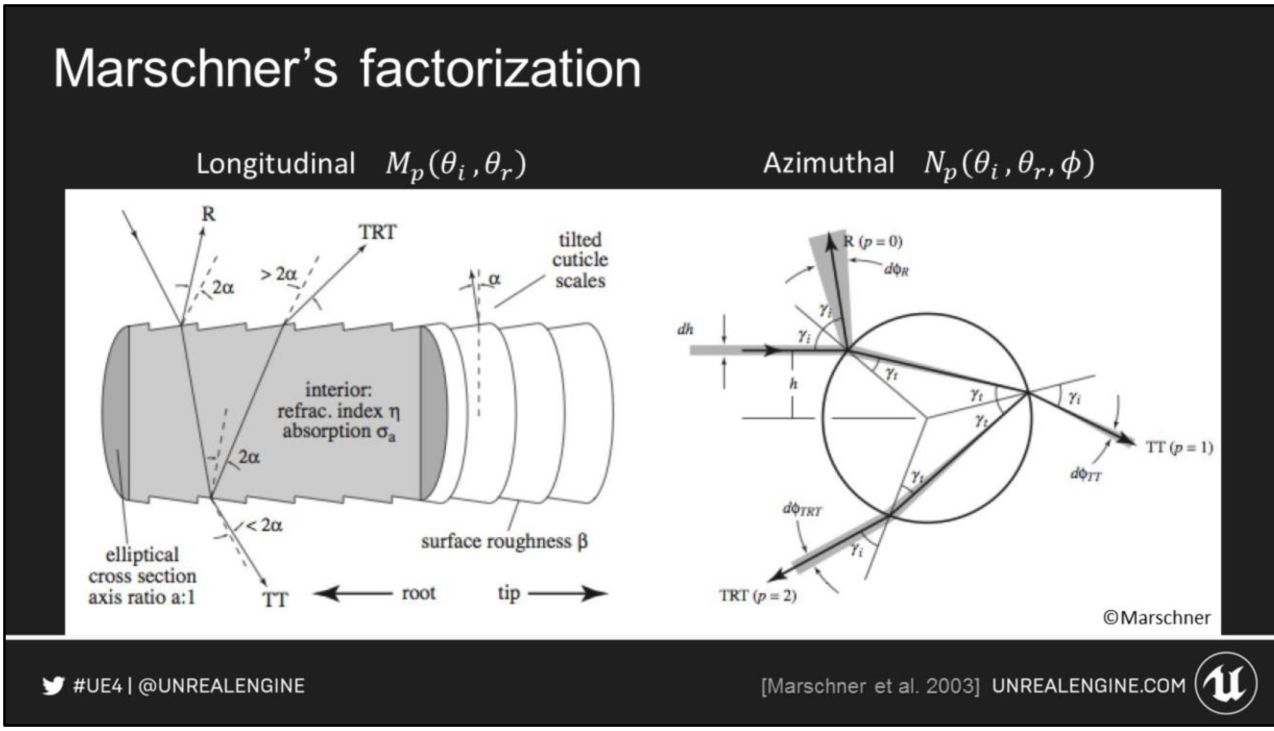
Each lobe broken into 2 functions:

$$S_p(\theta_i, \theta_r, \phi) = M_p(\theta_i, \theta_r) N_p(\theta_i, \theta_r, \phi)$$

#UE4 | @UNREALENGINE [Marschner et al. 2003] UNREALENGINE.COM

Marschner's factorization of the shading separates things into a sum of these lobes, p standing for the number of times the path passes through the interior

Each lobe is also broken into 2 functions, M and N…



These are the longitudinal and azimuthal scattering functions M and N

The the longitudinal function describes scattering down the length of the fiber and the azimuthal function describes it radially around the fiber

Real-time Marschner

- The Marschner model has been made performant before
- Used Look Up Tables (LUTs) to be efficient
- Requires baking all hair parameters into LUTs

#UE4 | @UNREALENGINE

[Hubert et al. 2005] UNREALENGINE.COM

That isn’t practical for us.

We need many colors of hair, roughness, etc.
In some cases those need to vary within the same hair style, streaks of color, patterns in fur.

There are many characters coming in Paragon that haven’t been announced yet and I can’t show but trust me; every variation of hair under the sun needs to be supported

Our shading model

- Approximation of model used at Weta
  - "An Energy-Conserving Hair Reflectance Model"
  - Created reference implementation first

- We model R, TT, TRT paths
- We don't handle eccentricity

- Lots of curve fitting and function approximation
- Please refer to the original papers for definition of all symbols and theoretical background

[d'Eon et al. 2011]

Bonus:

Longitudinal scattering function $M_p(\theta_i\,,\theta_r)$

Weta's energy conserving function too expensive

$$M_p(\theta_i\,,\theta_r) = \left(\frac{1}{ve^{\frac{2}{v}} - v}\right) e^{\frac{1-\sin\,\theta_i\,\sin\,\theta_r}{v}} I_0\left(\frac{\cos\theta_i\,\cos\theta_r}{v}\right)$$

$$v = {\beta_p}^2$$

Used Gaussian instead

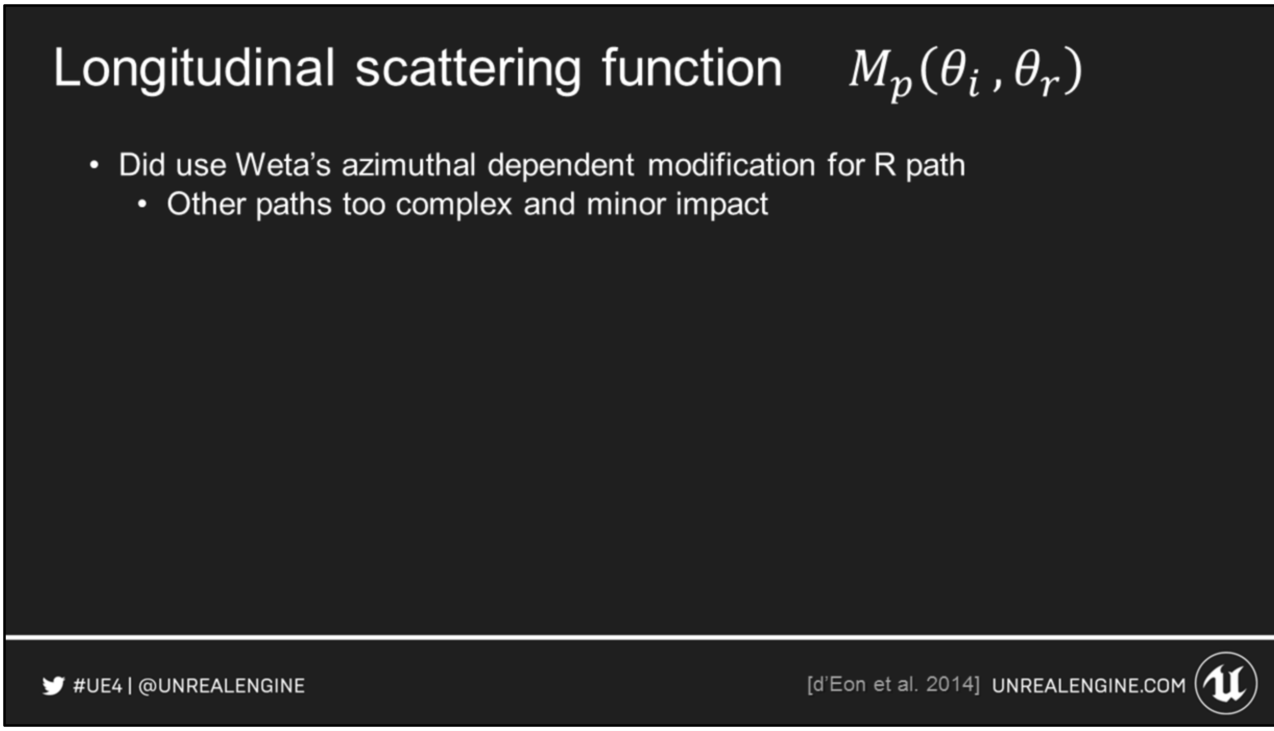
$$M_p(\theta_i\,,\theta_r) \approx \frac{1}{\beta_p\sqrt{2\pi}} e^{-\frac{(\sin\,\theta_i+\sin\,\theta_r-\alpha_p)^2}{2{\beta_p}^2}}$$

#UE4 | @UNREALENGINE [d'Eon et al. 2011] UNREALENGINE.COM

Lets dive in to these functions.
One of the improvements that Weta's model had over Marschner's is an energy conversing longitudinal scattering function

We use a Gaussian instead. Having a cheaper but more accurate alternative to d'Eon's function would be nice. This is one area in particular I'd like to improve.

2 symbols that are good to explain here are beta and alpha
Beta is based on the roughness of the fiber
Alpha is based on the tilt of the cuticle scales



We did use Weta's azimuthal dependent modification from their follow up paper. For R this is cheap enough and looks slightly better.

Azimuthal R

$$N_R(\theta_i\,, \theta_r, \phi)$$

R path is simple:

$$N_R(\theta_i\,, \theta_r, \phi) = \left(\frac{1}{4}\cos\frac{\phi}{2}\right) A(0,h)$$

Attenuation term:

$$A(0,h) = F\left(\eta, \sqrt{\frac{1}{2} + \frac{1}{2}(\omega_i \cdot \omega_r)}\right)$$

Use trig identity:

$$\cos\frac{\phi}{2} = \sqrt{\frac{1}{2} + \frac{1}{2}\cos\phi}$$

Schlick Fresnel:

$$F(\eta, x\,) = F_0 + (1 - F_0)(1 - x)^5$$

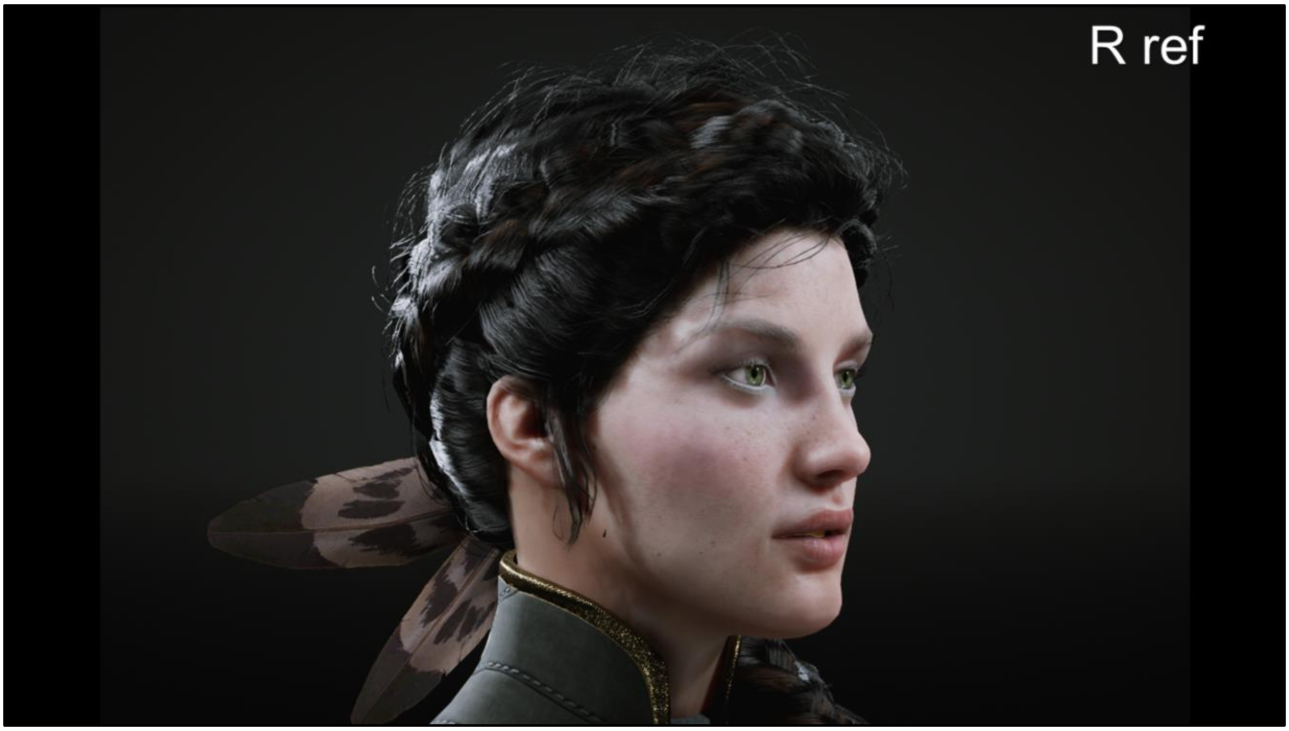
$$F_0 = \frac{(1-\eta)^2}{(1+\eta)^2}$$

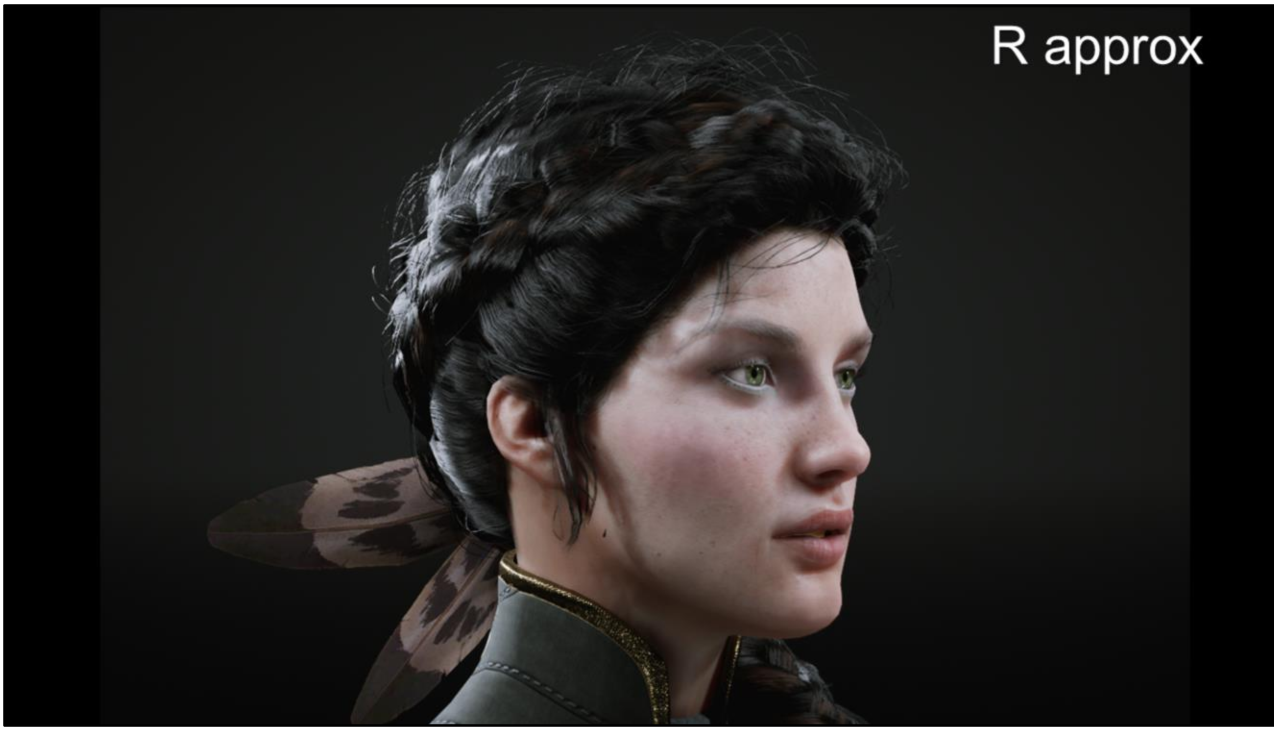
#UE4 | @UNREALENGINE

[d'Eon et al. 2014] UNREALENGINE.COM

Azimuthal for the R path is fairly simple
The distribution function is just CosHalfPhi
Using a trig identity can skip any inverse trig functions

The attenuation is the Fresnel function


R ref


R approx

Azimuthal

$$N_p(\theta_i, \theta_r, \phi)$$

Other paths not so simple:

$$N_p(\theta_i, \theta_r, \phi) = \frac{1}{2}\int_{-1}^{1} A(p,h) D_p\big(\phi - \Phi(p,h)\big) dh$$

Attenuation term:

$$A(p,h) = (1-f)^2 f^{p-1} T(\mu_a, h)^p$$

$$f = F\left(\eta, \cos(\theta_d)\sqrt{1-h^2}\right)$$

Absorption term:

$$T(\mu_a, h) = \exp\left(-2\mu_a \frac{1+\cos(2\gamma_t)}{\cos\theta_t}\right)$$

Weta solved the paths internal to the fiber through numerical integration with a Gaussian detector. This is a very nice generalization of the problem but is far too expensive for our purposes.

It isn't important to understand this integral for the sake of understanding our approximation. Just know this is too expensive to solve as is.

We instead are going to solve other paths as single lobes more in line with how Marschner originally did.

It is useful to understand the attenuation term A(). This is broken into 2 subterms, Fresnel and absorption.

Azimuthal TT

$$N_{TT}(\theta_i\,, \theta_r, \phi)$$

For TT the exact solution for offset h is:

$$a = \frac{1}{\eta'}$$

$$h_{TT} = \frac{\operatorname{sign}(\phi)\cos\frac{\phi}{2}}{\sqrt{1 + a^2 - 2a\operatorname{sign}(\phi)\sin\frac{\phi}{2}}}$$

Using some trig identities:

$$h_{TT}{}^2 = \frac{\frac{1}{2} + \frac{1}{2}\cos\phi}{1 + a^2 - 2a\sqrt{\frac{1}{2} - \frac{1}{2}\cos\phi}}$$

First we need to solve for h which is the offset from the center of the fiber
Eta prime is the modified index of refraction

We can simplify this a bit but it’s still heavier than I’d like

Azimuthal TT

$$N_{TT}(\theta_i, \theta_r, \phi)$$

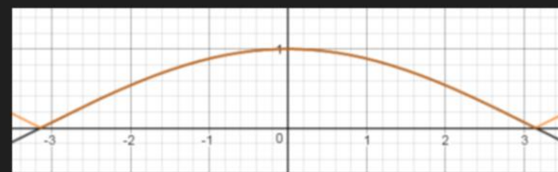
For TT the exact solution for offset h is:

$$a = \frac{1}{\eta'}$$

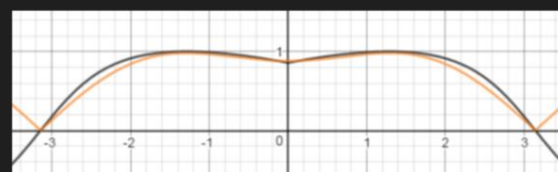
$$h_{TT} = \frac{\mathrm{sign}(\phi)\cos\frac{\phi}{2}}{\sqrt{1 + a^2 - 2a\,\mathrm{sign}(\phi)\sin\frac{\phi}{2}}}$$

Approximation:

$$h_{TT} \approx (1 + a(0.6 - 0.8\cos\phi)\cos\frac{\phi}{2}$$



$a = 0$



$a = 0.6$

… so we approximate!

Graph the function
Function approximation like this is somewhat of a black art.
Make intelligent guesses about what can be removed or replaced. Try and see what works.
Came up with this
Uses CosHalfPhi that we had from before

Ignoring sign of h, which turns out to not be relevant for our use, we have an equation that is cheaper

Modified index of refraction η'

Original:

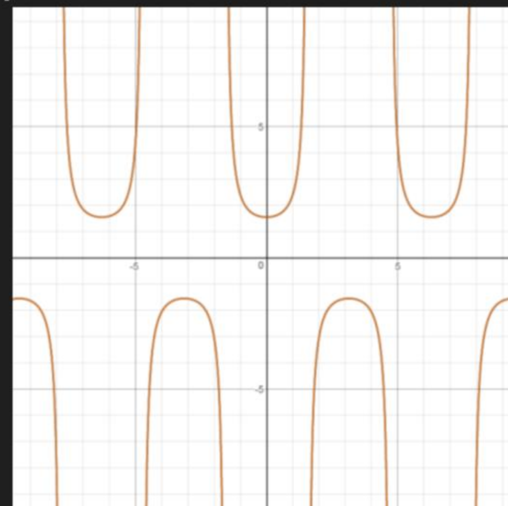
$$\eta' = \frac{\sqrt{\eta^2 - \sin^2\theta_d}}{\cos\theta_d}$$

Approximation:

$$\eta = 1.55$$

$$\eta' \approx \frac{1.19}{\cos\theta_d} + 0.36\cos\theta_d$$

$$\text{error} < 0.68\%$$



Now for the modified index of refraction
If we fix the original index of refraction at 1.55 which is the IOR for human hair, this is a very accurate approximation.

Bonus:
There are multiple ways to arrive at this general form of approximation.

The way I found it is I noticed in terms of cos theta d the function looked very much like 1/x. Adding a linear term made the fit much better. Tweak the constants until the max relative error is low. Done.

Alternatively, a Tayler series of the numerator with cos theta d replaced by x. The first 2 terms have the same form but slightly different constants. The constants I have here are a better fit.

Azimuthal TT

$$N_{TT}(\theta_i, \theta_r, \phi)$$

Absorption term:

$$T(\theta, \phi) = \exp\left(-2\mu_a \frac{1 + \cos(2\gamma_t)}{\cos\theta_t}\right)$$

$$\gamma_t = \sin^{-1}\left(\frac{h}{\eta'}\right)$$

Chose to use Pixar's absorption term instead:

$$T(\theta, \phi) = \exp\left(-\zeta(C)\frac{\cos\gamma_t}{\cos\theta_d}\right)$$

#UE4 | @UNREALENGINE

[Pekelis et al. 2015] UNREALENGINE.COM

Found the look more pleasant

C is the BaseColor of hair

Azimuthal TT

$$N_{TT}(\theta_i, \theta_r, \phi)$$

Pixar's absorption term:

$$T(\theta, \phi) = \exp\left(-\zeta(C)\frac{\cos\gamma_t}{\cos\theta_d}\right)$$

Plug this in and simplify:

$$\gamma_t = \sin^{-1}\left(\frac{h}{\eta'}\right)$$

Chose $\zeta(C)$ to make color picking intuitive:

$$T(\theta, \phi) = C^{\frac{\sqrt{1-h^2a^2}}{2\cos\theta_d}}$$

[Pekelis et al. 2015]

Plug this in for gamma t and simplify. You can see now that the sign of h is irrelevant for us since it is always squared.

Azimuthal TT

$N_{TT}(\theta_i, \theta_r, \phi)$

Started with Pixar's logistic distribution function approximation:

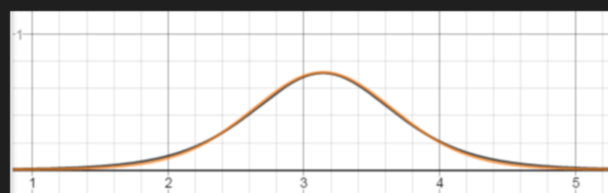
$$D(\phi, s, \mu) = \frac{e^{\frac{\phi-\mu}{s}}}{s\left(1+e^{\frac{\phi-\mu}{s}}\right)^2}$$

But formula for s_{TT} was not worth it for us

Our approximation:

$$D_{TT}(\phi) = D(\phi, 0.35, \pi)$$

$$D_{TT}(\phi) \approx e^{-3.65\cos\phi - 3.98}$$



Compared against logistic

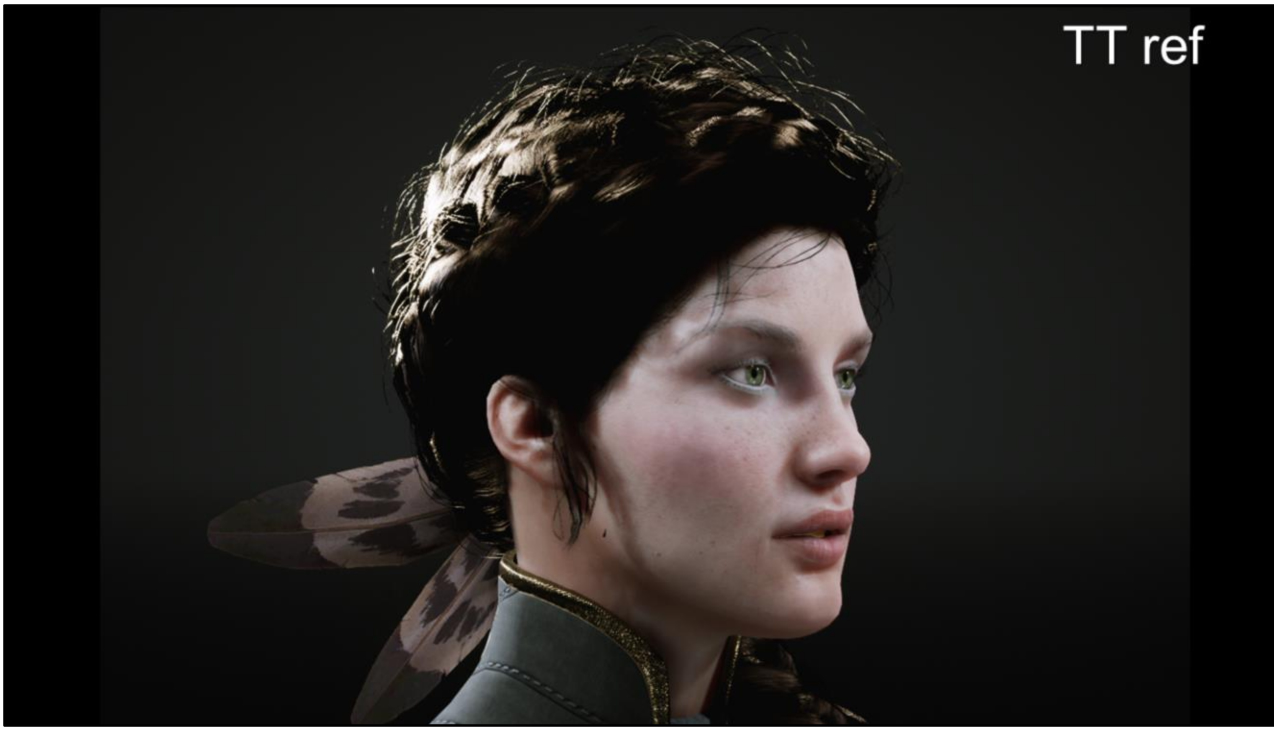
#UE4 | @UNREALENGINE

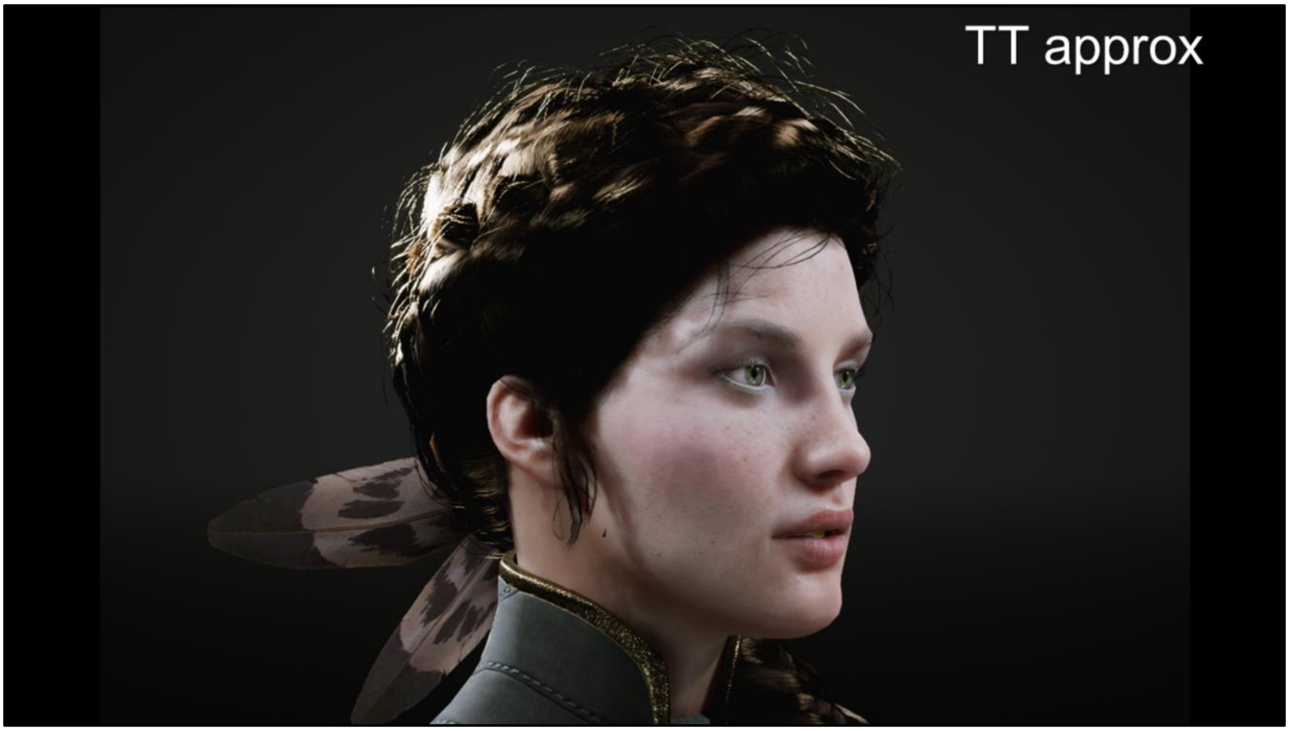
[Pekelis et al. 2015] UNREALENGINE.COM

The formula for sTT was too expensive for us for too little of an improvement.

Instead of using Pixar's solution for s we pick the constant 0.35.
From there we approximate the logistic with a Gaussian in terms of cos phi to avoid inverse trig

Bonus:
Note that Pixar only switched from Gaussian to Logistic to make importance sampling closed form. That isn't useful for us. The reason Logistic was where I started was that I started with Pixar's approximation for sTT. It funny to circle back but that was the history.


TT ref


TT approx

Azimuthal TRT

$$N_{TRT}(\theta_i, \theta_r, \phi)$$

h, the offset for Fresnel:

$$h_{TRT} = \frac{\sqrt{3}}{2}$$

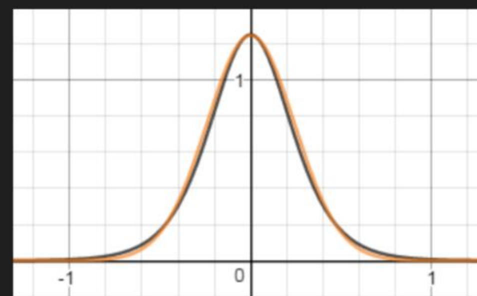
Absorption term:

$$T_{TRT}(\theta, \phi) = C^{\frac{0.8}{\cos \theta_d}}$$

Distribution:

$$D_{TRT}(\phi) = \frac{3}{4} D(\phi, 0.15, 0)$$

$$D_{TRT}(\phi) \approx e^{17 \cos \phi - 16.78}$$



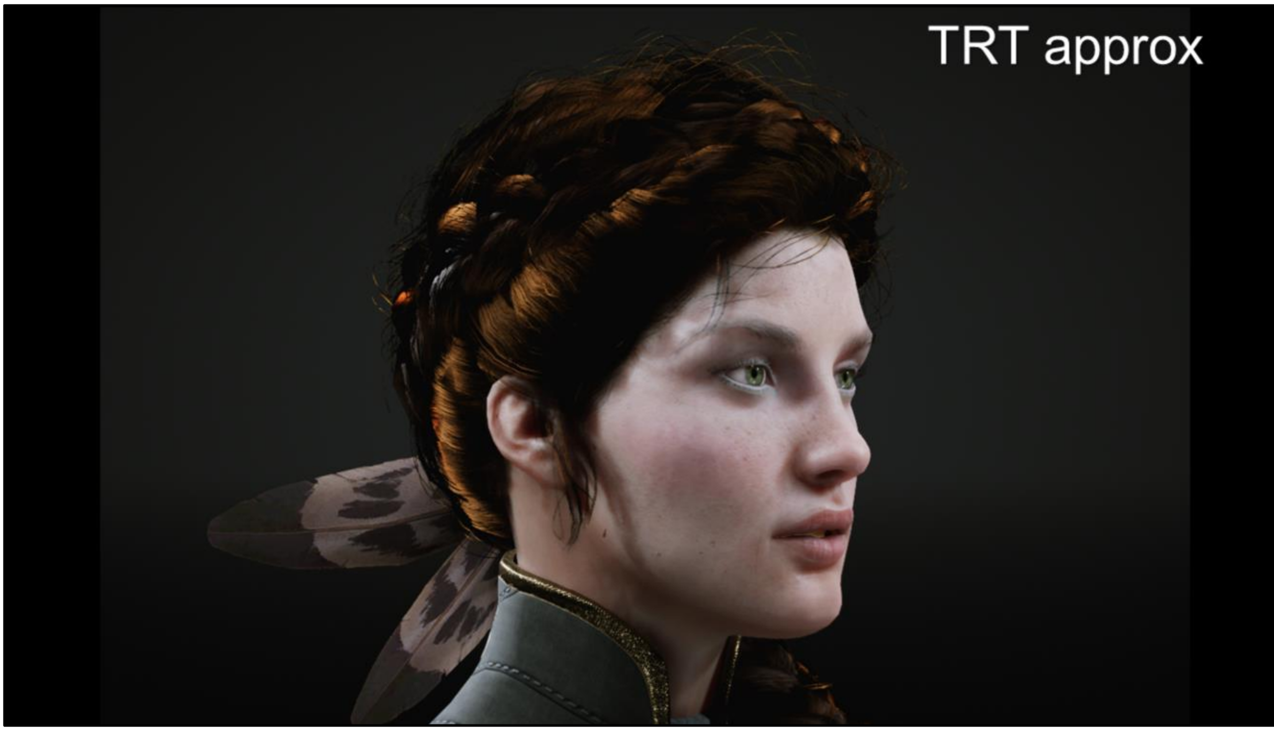
Compared against logistic

For TRT we have to be brutal.

Marschner solved this and glints with root finding for h which has multiple solutions which means multiple lobes.
Weta solved it with heavy numerical integration
Pixar solved it with very large look up tables

I chose a single simple lobe.
h, the offset of the path from the fiber center, for the purposes of Fresnel I use a constant.
In the absorption term all use of h or the modified index of refraction are gone, replaced by a constant.
The distribution I again started with the logistic function and approximated it with a Gaussian in terms of cos phi.


TRT ref


TRT approx

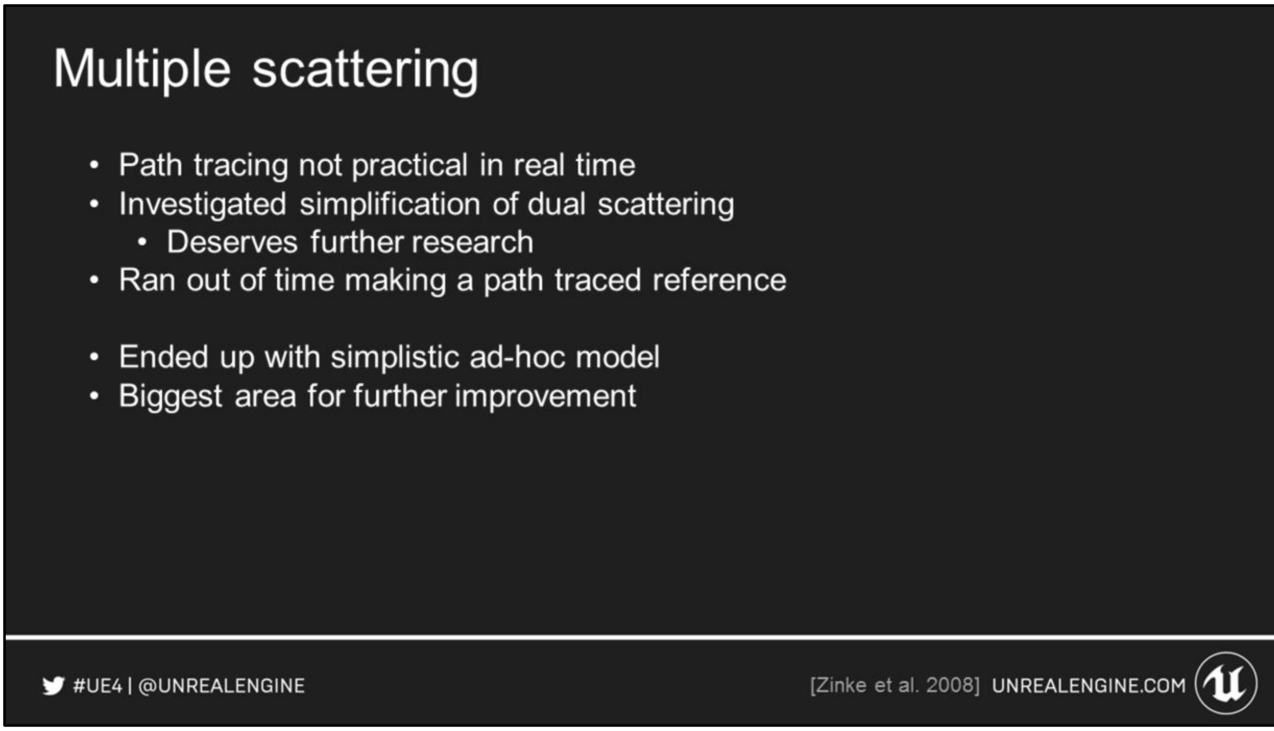

R+TT+TRT ref

But it’s still missing something

Multiple scattering

- Need to render a volume of hair not just a single strand
- This means multiple scattering events of a light path through the volume
  - Very important for look of light hair

Especially for light hair since the amount of absorption per fiber is less



It feels wrong to present this part in a physically based shading course but instead consider this a call to arms.
This is the biggest opportunity for others to improve this model.

Multiple scattering

Fake normal:

$$n = \frac{\omega_r - u(u \cdot \omega_r)}{\|\omega_r - u(u \cdot \omega_r)\|}$$

Shadowing is exponential falloff from shadow map value

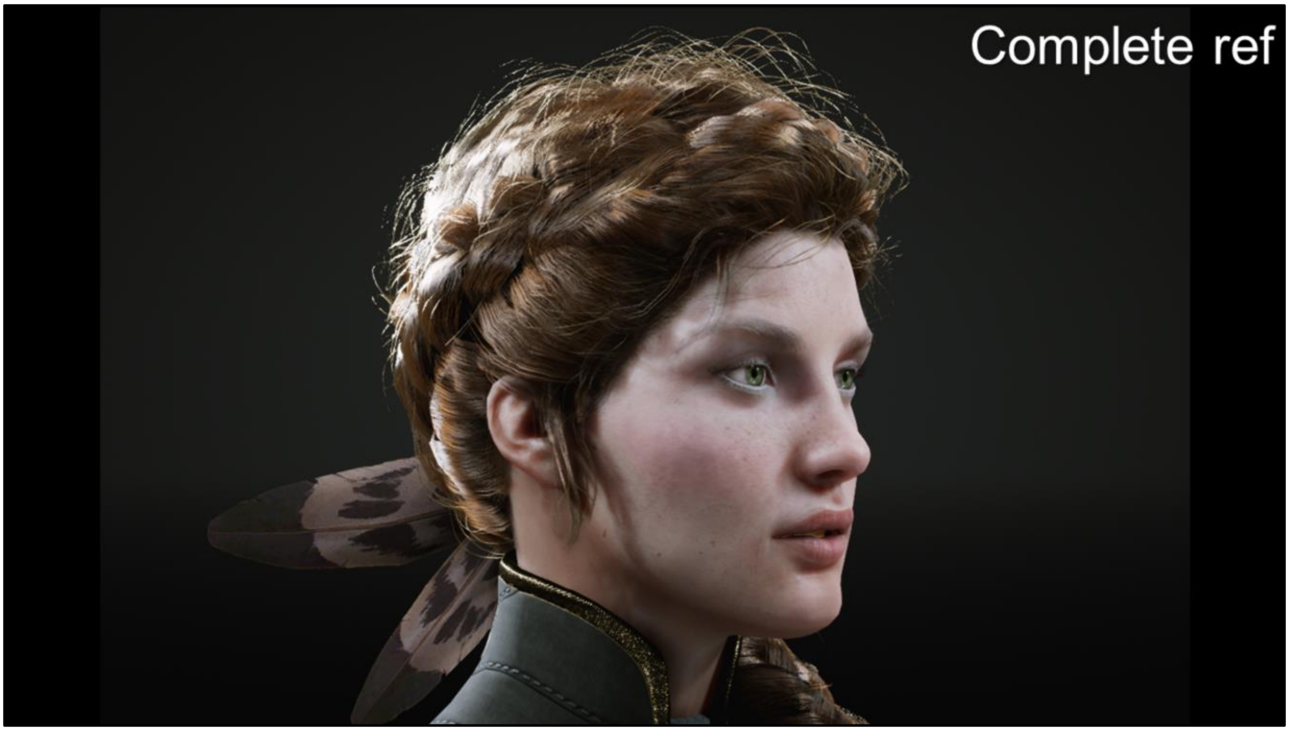
Absorption over direct light path using shadow for path length:

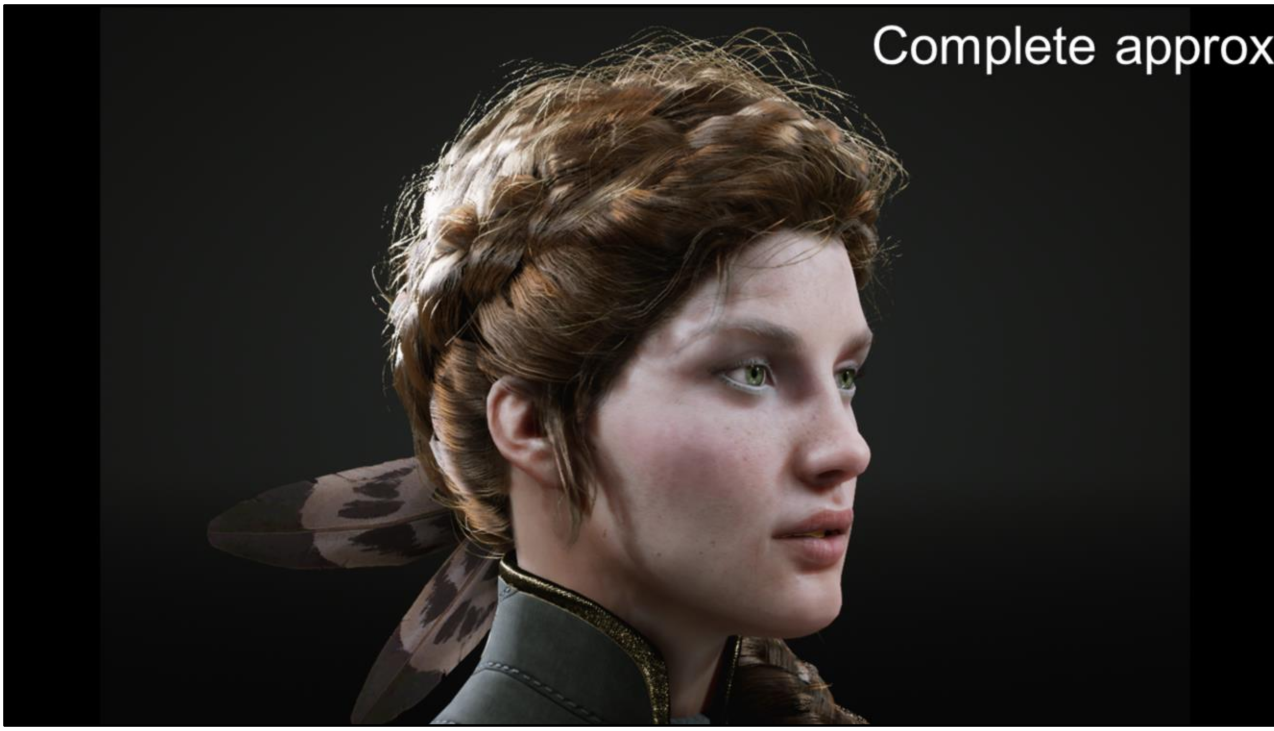
$$S_{scattering} = \sqrt{C}\left(\frac{n \cdot \omega_i + 1}{4\pi}\right)\left(\frac{C}{Luma}\right)^{(1-Shadow)}$$

reference.

Here is what it looks like. I don’t have a reference to compare it to like the other terms.

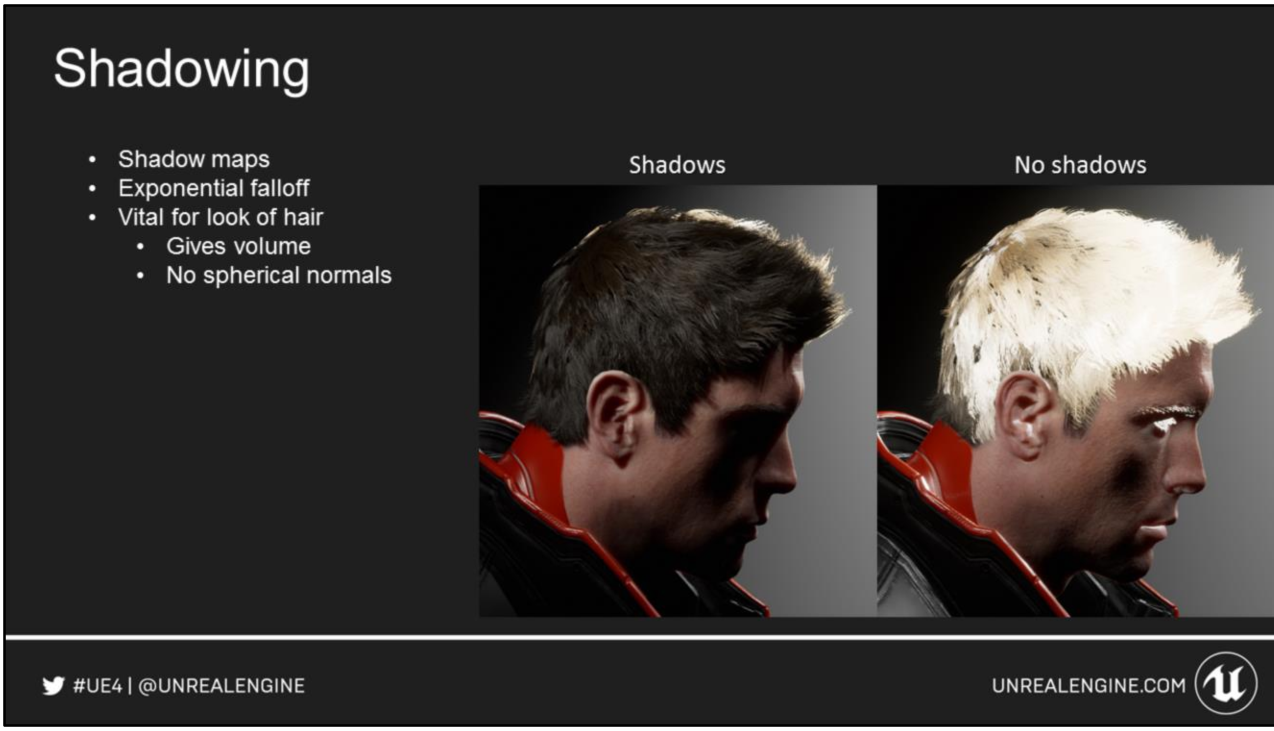
Bonus:
Ignore that highlight on the braid. It is actually from the surface underneath the hair.


Complete ref


Complete approx

Artist parameters

- BaseColor
  - C in the equations
- Specular
  - Scales the R term
- Roughness
  - $\beta_R = Roughness^2$
  - $\beta_{TT} = 0.5\ Roughness^2$
  - $\beta_{TRT} = 2\ Roughness^2$
- Scatter
  - Scales multiple scattering term
- Shift
  - $\alpha_R = -2\ Shift$
  - $\alpha_{TT} = Shift$
  - $\alpha_{TRT} = 4\ Shift$



As mentioned already I rely on shadows for the shading model. It is absolutely vital to the look which is easy to see by disabling them.

Shadow maps are PCF filtered with an exponential falloff instead of a hard comparison.
The result is volumetric looking without any provided normals

Shadowing

- What about non-shadowing lights?
  - FX
  - Environment fill lights
  - Low end hardware
- Screen space ray cast against depth buffer

Shadow maps Screen space shadows

#UE4 | @UNREALENGINE UNREALENGINE.COM

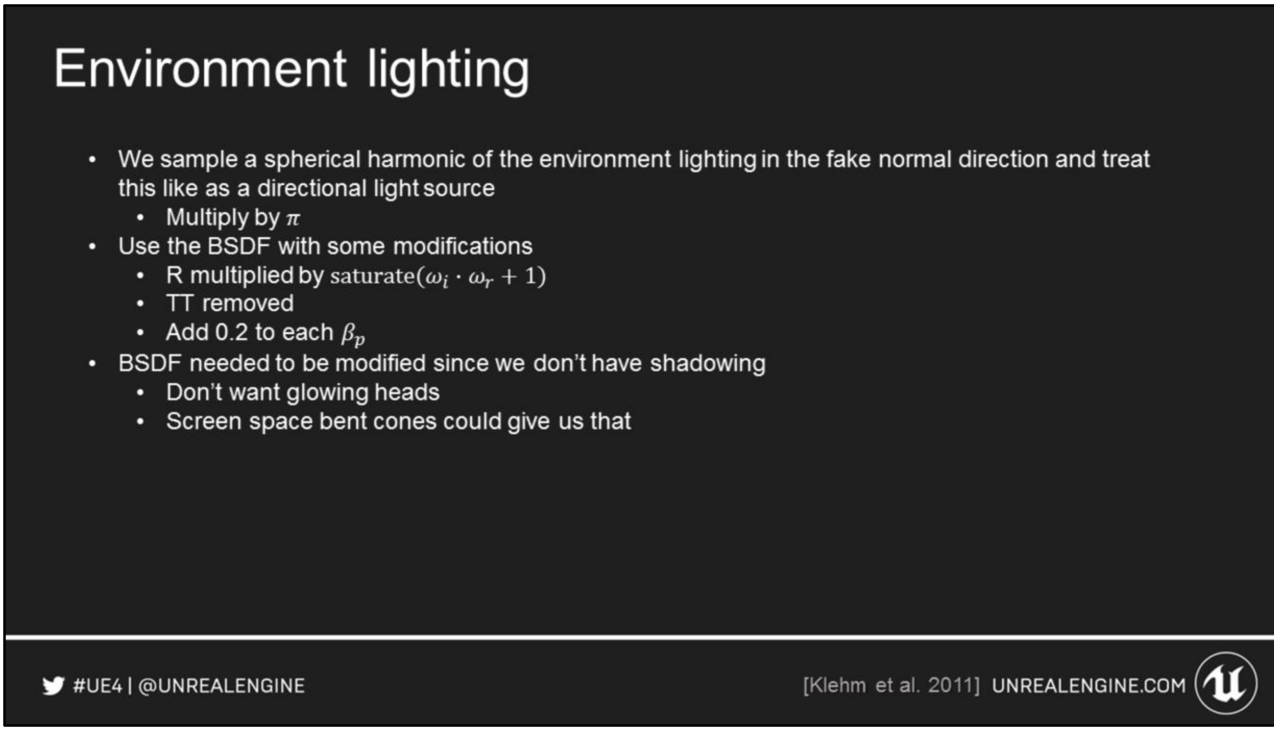
Environment lighting

- Researched a number of approaches
- The most intriguing:
  - “Interactive Hair Rendering and Appearance Editing under Environment Lighting”
  - Approximates Marschner’s model with circular Gaussians that can be analytically integrated against spherical Gaussian lights
- In the end we didn’t need to be that sophisticated

#UE4 | @UNREALENGINE

[Xu et al. 2011] [Mehta el al. 2012] UNREALENGINE.COM

“Analytic Tangent Irradiance Environment Maps for Anisotropic Surfaces” which should be similar cost but more accurate results. In all honesty I didn’t pursue either of these options because I ran out of development time. For the cases I was testing at the time this very simple solution looked good enough.

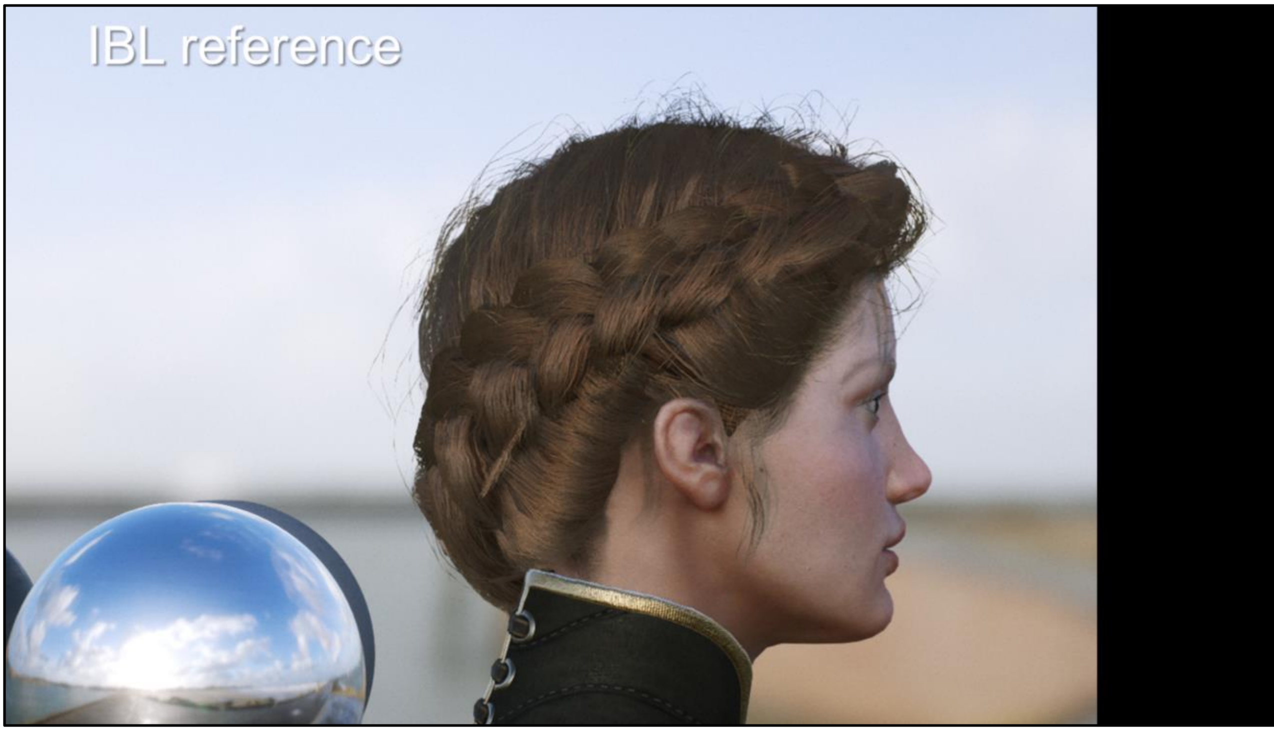


This is what we do

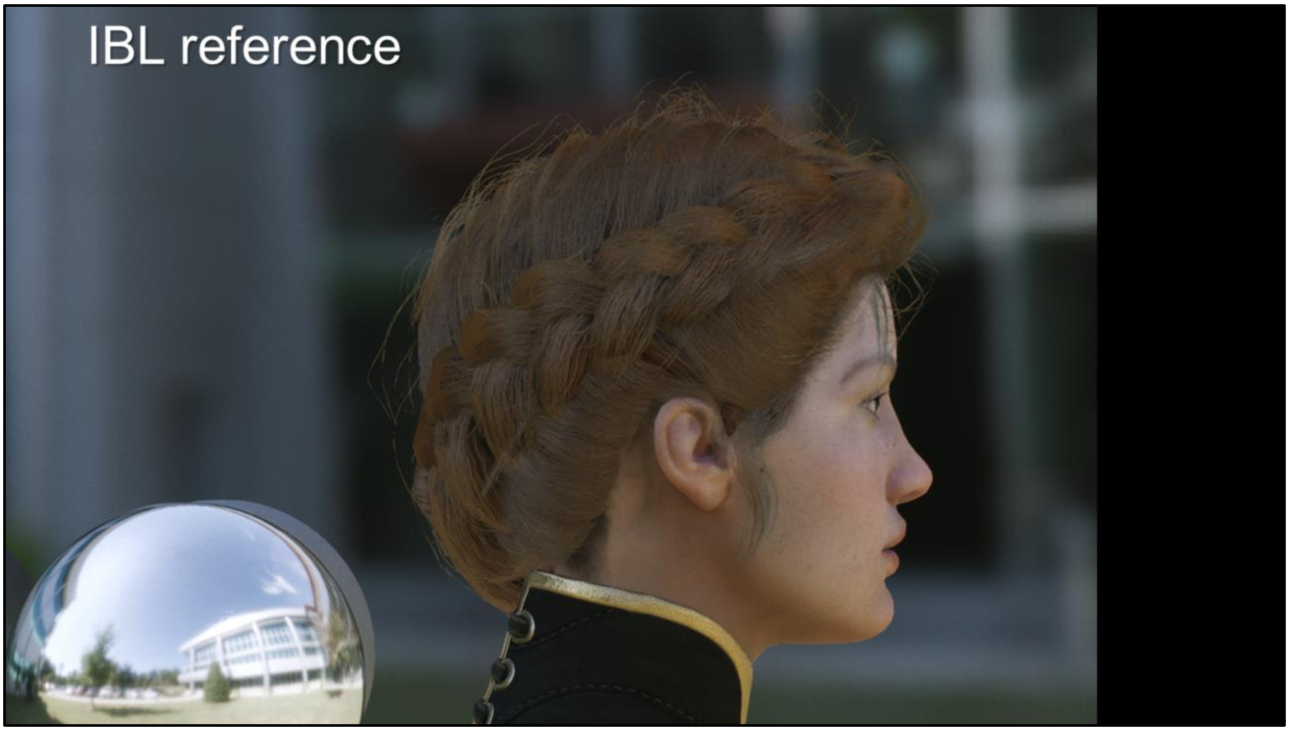
Increase the roughness as an approximation of a wider area light source.

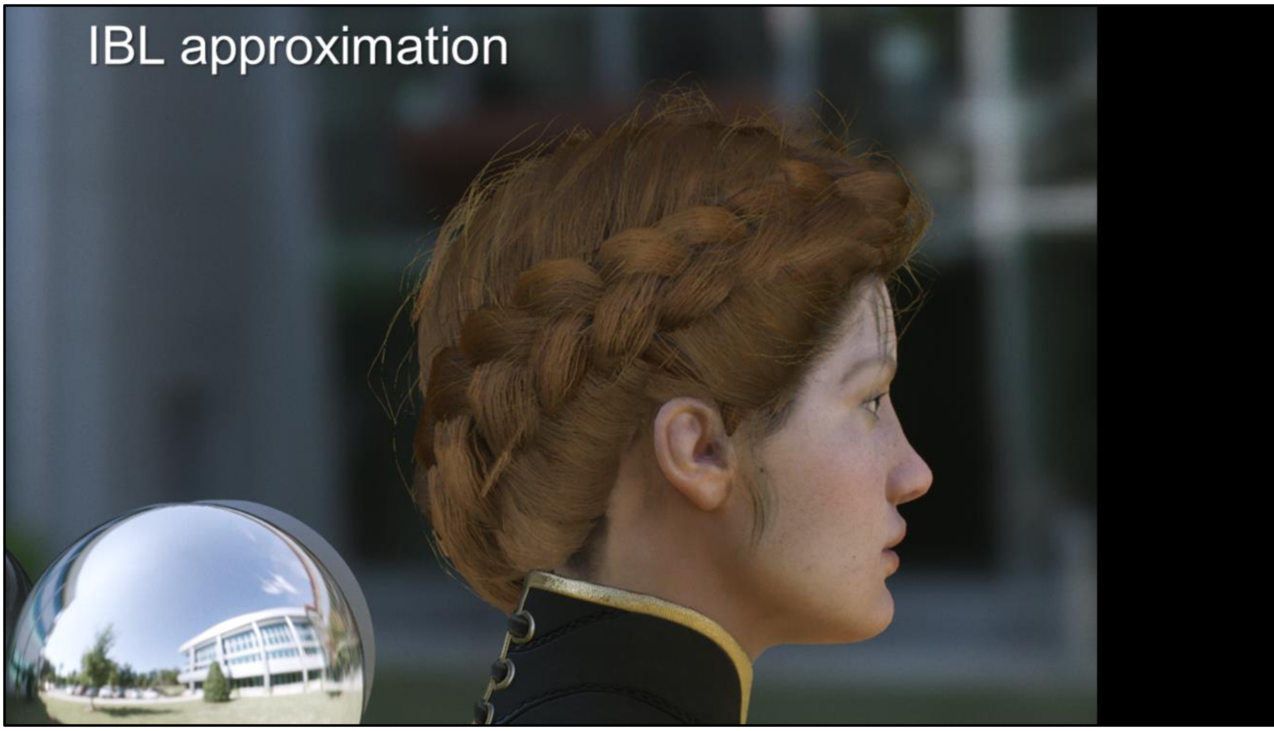
We don't have shadowing from shadow maps so we need to artificially shadow paths that would likely be blocked by a volume of hair. These are primarily those that are coming from the opposite side.

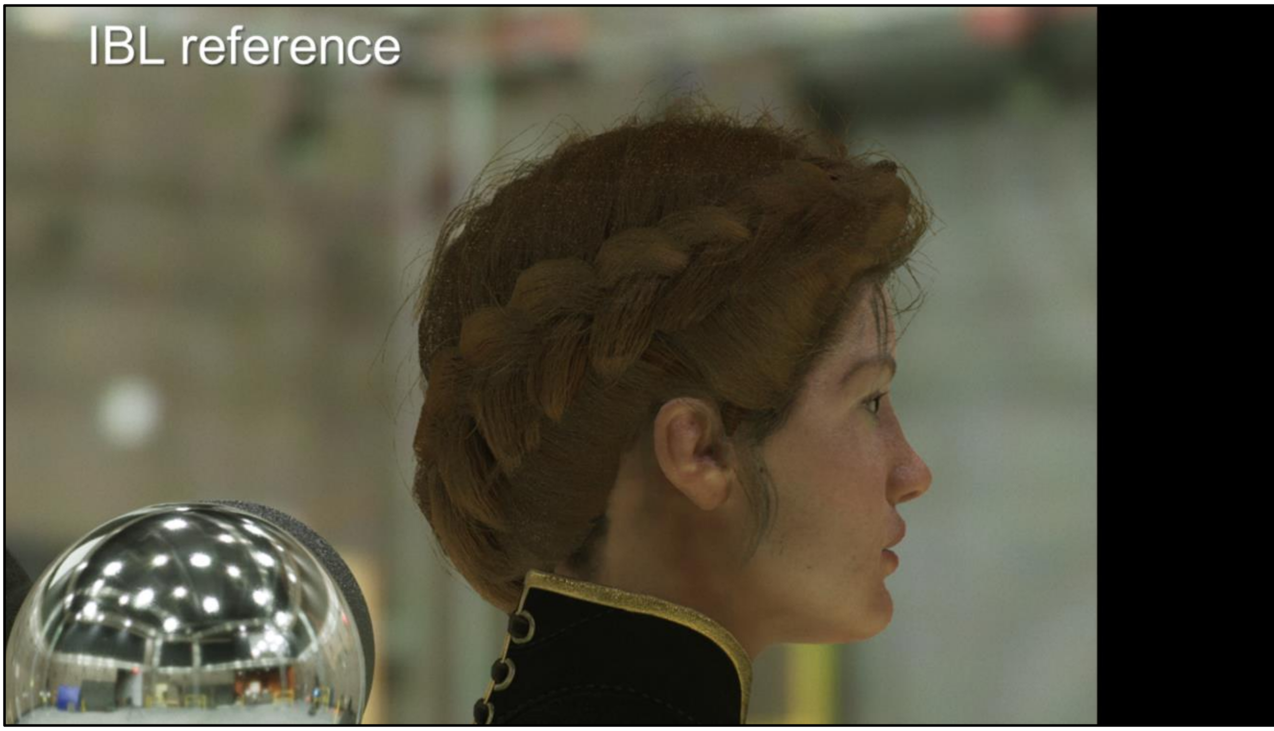
If you had some computed form of directional occlusion such as bent cones we could make less assumptions about which paths are blocked.

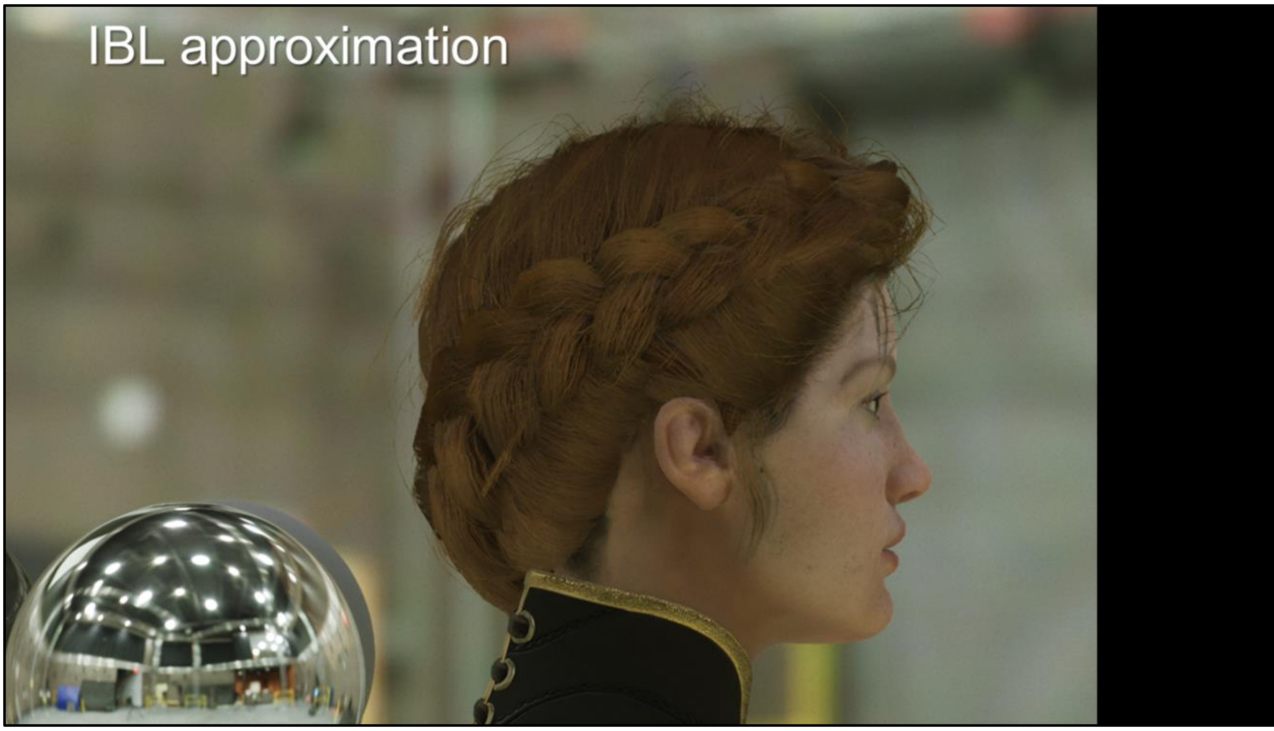

IBL reference


IBL approximation


IBL reference


IBL approximation


IBL reference


IBL approximation

I’ll leave you with some shots of a human head we will be releasing in the next version of Unreal so that there is a high quality example of all of the character rendering work we have done of late.

Lastly thanks to, Epic, our character art team and to Stephen Hill and Stephen McAuley for reviewing these slides

References

- [Kajiya and Kay 1989] "Rendering fur with three dimensional textures"
- [Marschner et al. 2003] "Light Scattering from Human Hair Fibers"
- [d'Eon et al. 2011] "An Energy-Conserving Hair Reflectance Model"
- [d'Eon et al. 2014] "A Fiber Scattering Model With Non-Separable Lobes"
- [Pekelis et al. 2015] "A Data-Driven Light Scattering Model for Hair"
- [Kasyan 2013] "Playing with Real-Time Shadows"
- [Driancourt 2012] "Agni's Philosophy" http://www.4gamer.net/games/032/G003263/20130112002/
- [Zinke et al. 2008, "Dual Scattering Approximation for Fast Multiple Scattering in Hair"
- [Karis 2014] "High Quality Temporal Supersampling"
- [Hubert and Donnelly 2005] "Hair Animation and Rendering in the Nalu Demo"
- [Xu et al. 2011] "Interactive Hair Rendering and Appearance Editing under Environment Lighting"
- [Mehta el al. 2012] "Analytic Tangent Irradiance Environment Maps for Anisotropic Surfaces"
- [Klehm et al. 2011] "Bent Normals and Cones in Screen-space"

Bonus slides

Hair geometry

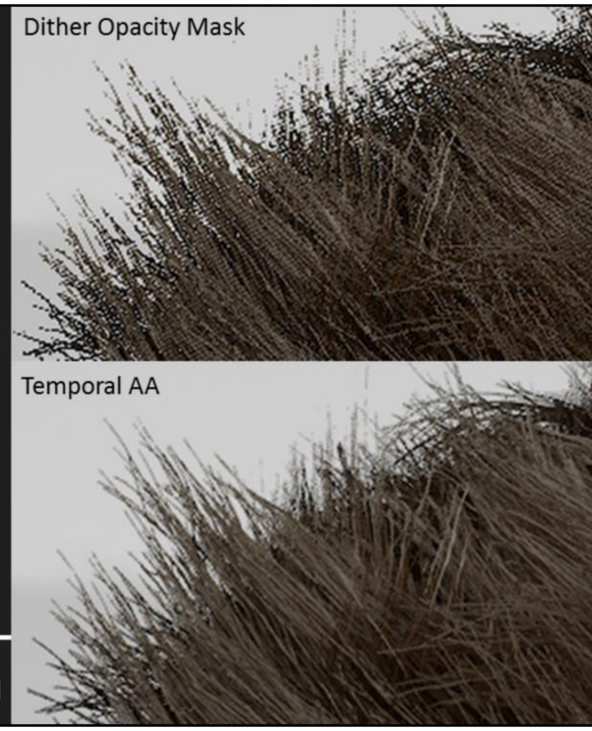
- Paragon is 60hz on PS4
  - Many characters on screen at once
- Pure splines generate too many polys to fill volume
- Tessellation off the table
  - Need to eventually support Dx10
- Mostly cards

Hair AA / translucency

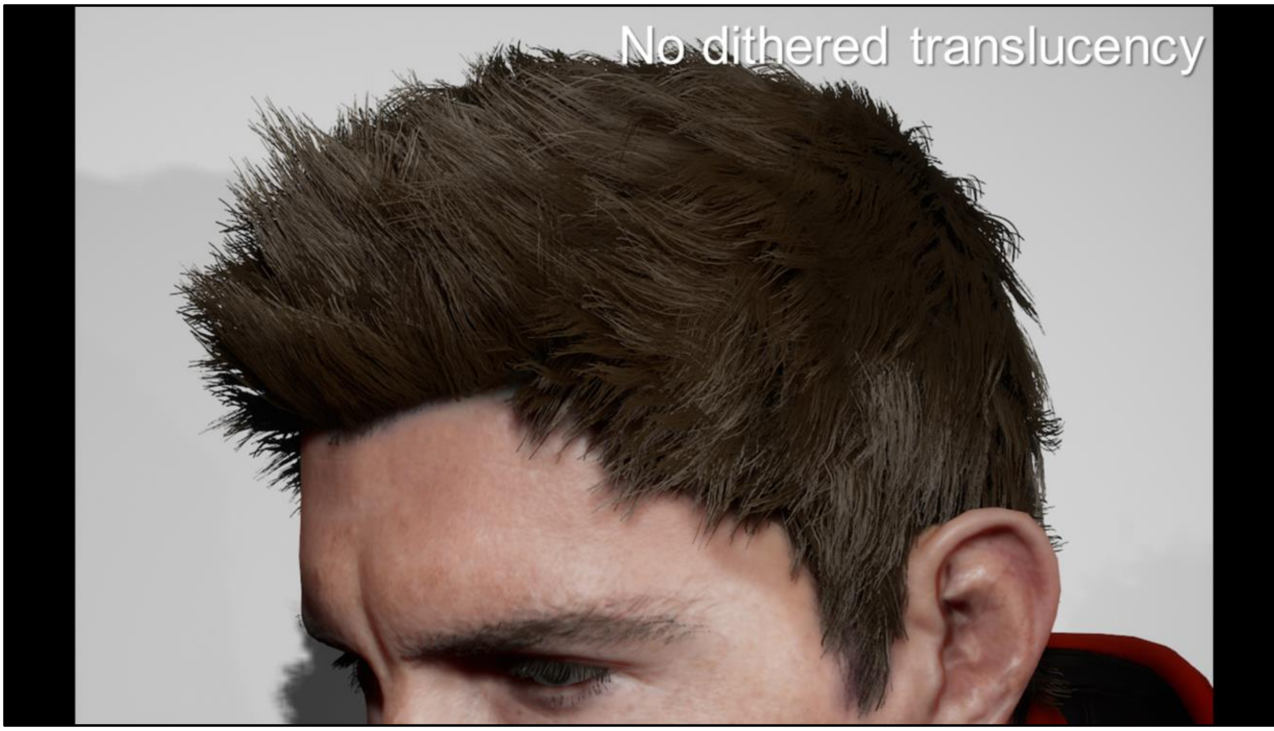
- Alpha blended translucency is problematic
  - Poor engine support for forward shading
  - Overdraw very expensive
  - Sorting complicated
- Opaque / alpha masked
  - Works with deferred shading
  - Single sample shaded per pixel
- Temporal dithering + Temporal AA = Cheap OIT
  - Called Dither Opacity Mask in engine
- Pixel depth offset gives volume to each card
  - Each hair at a different depth
  - Smoother intersections between cards and with scalp

#UE4 | @UNREALENGINE

[Karis 2014]




Soft


No dithered translucency


No depth offset


Soft