Figure 1: *Cars 3* ©Disney/Pixar 2017 is the latest Pixar Film to be rendered entirely with physically based shading

Pixar's Foundation for Materials

Christophe Hery, Ryusuke Villemin, Junyi Ling

1 INTRODUCTION

Pixar's Foundation Materials, `PxrSurface` and `PxrMarschnerHair`, are shipped with regular distributions of `Renderman` starting with v21. You can find ample description and examples of their behaviors at our product pages [Ren16b] and [Ren16a][1].

The philosophies and main principles behind our setups date back to our work on Physically Based Lighting for *Monsters University* and we refer the reader to a previous version of this very course where we described our full system from that time: [HV13]. We also explained our transition to our modern path tracing architecture, named `RIS`, in [VH15].

`PxrSurface` is the standard surface shader developed in the studio for *Finding Dory*, and used more recently on *Cars 3* and *Coco*. `PxrMarschnerHair` is also our internal illumination model for hair fibers, and has been in use since *The Good Dinosaur*. This document summarizes the various features found in these two shaders, then dives into a few unique choices we made, such as sampling, layering abstractions, energy conservation and specific and novel algorithms.

[1] These shaders also go beyond `Renderman`, as we share their inner code in various platforms and languages, for instance in our shading and look development application called `Flow`, which itself sits on top of Nvidia's OptiX. Our critical routines are thus written in both C++, Cuda and ispc, which we have employed since 2012, see [VH12]

In a path tracing rendering environment, surface BXDFs are tightly integrated with the global integrator. We will be presenting our modern integration and path editing techniques within our production physically based system in the companion course [HVS17]. Lastly, volume rendering is beyond the scope of this document. We refer readers to [Fon+17] for further discussions on this topic.

2 PXRSURFACE LOBES

`PxrSurface` contains ten lobes that cover the entire gamut of surface materials, from metals to water to skin, for all our films at Pixar since *Finding Dory*: 1 diffuse, 1 back diffuse, 3 specular, 1 iridescence, 1 fuzz, 1 subsurface, 1 single scatter, and 1 glass lobe. Each of the BSDF lobes is energy conserving. Our basic mode for combining these lobes is a simple linear blend. However we supplement this somehow fixed uber model, with user facing parameter layering material descriptions. This allows us the flexibility of a multilayered shading pipeline, while maintaining efficient and consistent rendering behaviors. Here we call attention to the distinction between **a.** compositing BSDF parameters and **b.** stacks of physical materials such as glass over metal. We will expand on this difference in later sections.

2.1 Diffuse

We find that a single forward facing diffuse lobe is sufficient in covering all use cases that we encountered. Our diffuse lobe can be switched from a Lambertian model to an Oren-Nayar model. We introduced a roughness parameter to seamlessly switch between these two modes. See Fig. 2. An additional diffuse BTDF is added to provide inexpensive backlighting that has a constant positive value for all outgoing directions on the opposite side of the surface from the incident ray – a “lampshade” model.

2.2 Specular

`PxrSurface` contains three separate specular lobes. There is a primary specular lobe designed for simple surfaces. We have also provided clearcoat and rough specular lobes. The reason for the three different speculars will be discussed in section 3. Each specular lobe employs state-of-the-art sampling techniques, such as [HD14], and can be of type “GGX” or “Beckmann”, and its Fresnel behavior can either be controlled “artistically” (i.e. with explicit zero degree and 90 degrees incidence colors) or “physically” (through a complex index of refraction parameter).

2.3 Reflection and Refraction

For transmissive materials such as glass and water, it is important that we closely couple reflection and refraction. Total internal reflection occurs at critical glancing angles with respect to the surface normal of the interface. This critical angle is a function of the index of refraction of the medium. If the reflective value and the refractive value are different then total internal reflection can occur that may be discontinuous. This could result in completely dark gaps in renders. We avoid this by having a single input IOR parameter for both reflection and refraction. Though not recommended, artists can circumvent this restriction by using a normal specular lobe and switching the specular Fresnel mode from “artistic” to “physical” to adjust Fresnel directly.

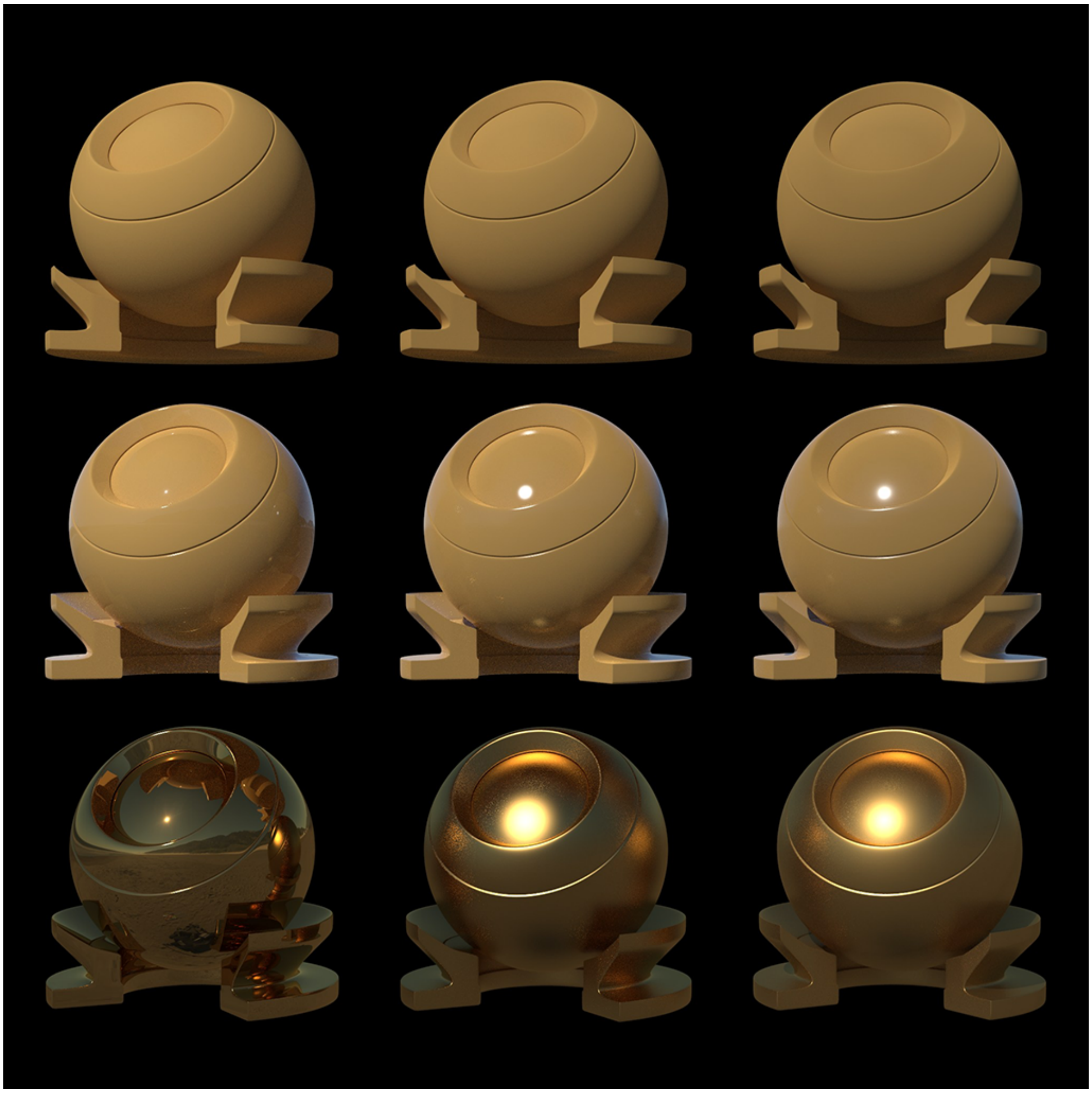
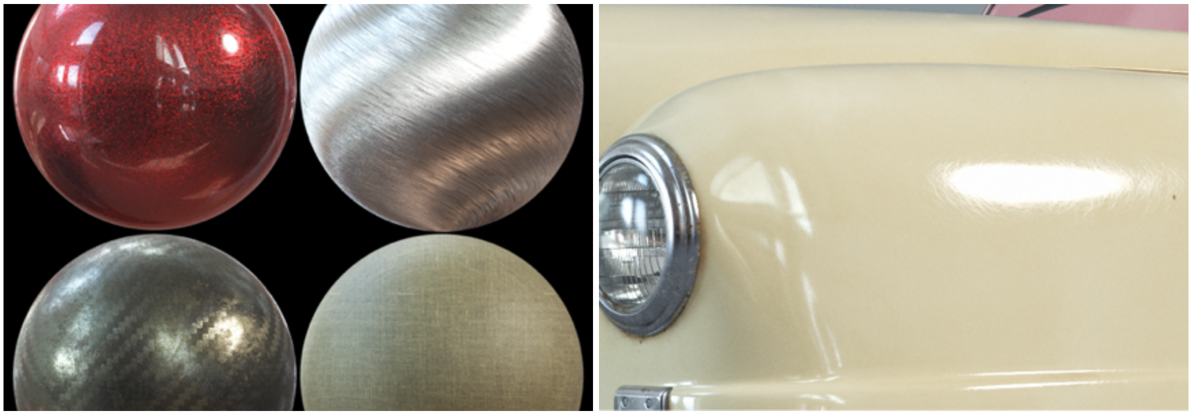


Figure 2: Basic diffuse and specular lobes. Top left to right: diffuse lobes going from Lambertian to Oren-Nayar. The middle row illustrates dielectric specular behavior. The bottom row illustrates metallic specular.

2.4 Bump To Roughness

We leverage Bump To Roughness mapping to recreate sub-displacement-sized micro features like metal flakes and clearcoat scratches. It has been a common problem in computer graphics that bump and displacement maps can appear differently when rendered from different distances. This problem arises from the fact that depending



(a) Array of simple example materials shaded with Bump To Roughness

(b) Concentric micro-scratch patterns that follow strong light-sources

Figure 3: Sub-displacement "microfeatures" in Cars 3, shaded with Bump To Roughness mapping. The spheres represent stripped-down versions of our shader for illustration purposes. The car paint shader focuses on a traditionally difficult-to-achieve look with other methods.

on the filter size of the pixel with relation to the texel, the value of the displacement is different. The render can thus take on vastly different looks depending on that relationship. At distance, a great looking and intricately textured surfaces can be "filtered away" to make that surface look smooth and "plastic-y".

We refer to the following pieces of work: [OB10], [Dup+13] and [HKL14]. They all independently tackled this particular issue and arrived at similar ideas and mathematical models, but with different implementations that suited their end rendering systems and illumination models. The other similar methods are referred to as LEAN and LEADR mapping. Note these these algorithms were mostly derived for Gaussian-style specular lobes (such as Beckmann), but we found that they work pretty well with GGX.

With a micro-facet based specular BSDF, one can turn bump-maps, normal-maps and displacement-maps into statistically-based slope maps. By using standard filtering kernels of these mip-mapped textures, one can reconstruct the microfacet distribution as well as anisotropy of the BSDFs analytically. An additional benefit is that one can shade small sub-displacement-scale features that were previously hard or impossible to achieve. This class of features include things like car paint metal flakes, tiny multi-directional microscratches on hard surfaces that form concentric scratch patterns that follow the sun (like in a metal spoon), carbon fibers, brushed metals with variation in the brushing, as in Figures 3 and 4, as well as high-sheen cloth materials such as silk and satin. These materials previously relied on ad-hoc, per-asset-type shading methods that filtered poorly and sometimes relied on aliased patterns to work.

Figure 4: Louise from *Cars 3* ©Disney/Pixar 2017. She is an older but very well maintained vehicle. Bump To Roughness mapping is also used to achieve subtle aging effects on her chrome and painted materials.

3 LAYERING

3.1 Sampling on Multi-Lobe BSDFs

As described in the previous section, production BSDFs are usually composed of multiple simple BSDFs (that we call lobes). From the point of view of the integrator though, we need to be able to ask for a sample independently of the number of lobes inside an uber BSDF, in a transparent manner. If we have a pure diffuse object, then it's easy: just call a Lambertian cosine weighted sampling function. If we have a mix of a diffuse and a specular lobe, we first need to choose which lobe will be responsible for sampling a direction. Therefore the sampling process is decomposed into two stages:

1. Compute probabilities for each lobe, then select one according to these probabilities.
2. Use the chosen lobe to generate a sampling direction and corresponding BSDF value and pdf.

To perform 1, we need an importance metric for every lobe. For diffuse, we can simply take the albedo. For a more complex lobe like specular, the raw albedo is a very bad estimate because of the strong Fresnel effect. That being said, we can't compute the Fresnel yet, since the outgoing direction $\mathbf{l}$ is unknown until 2. Thus in `PxrSurface` (and `PxrMarschnerHair`), for this importance estimation phase, we use an approximation of the Fresnel, based only on the incoming direction $\mathbf{v}$. Refer to Figure 6c and Figure 6d to appreciate the variance reduction with this approach.

In order to reduce variance even more, we can use one-sample MIS (see [VG95]) between the lobes:

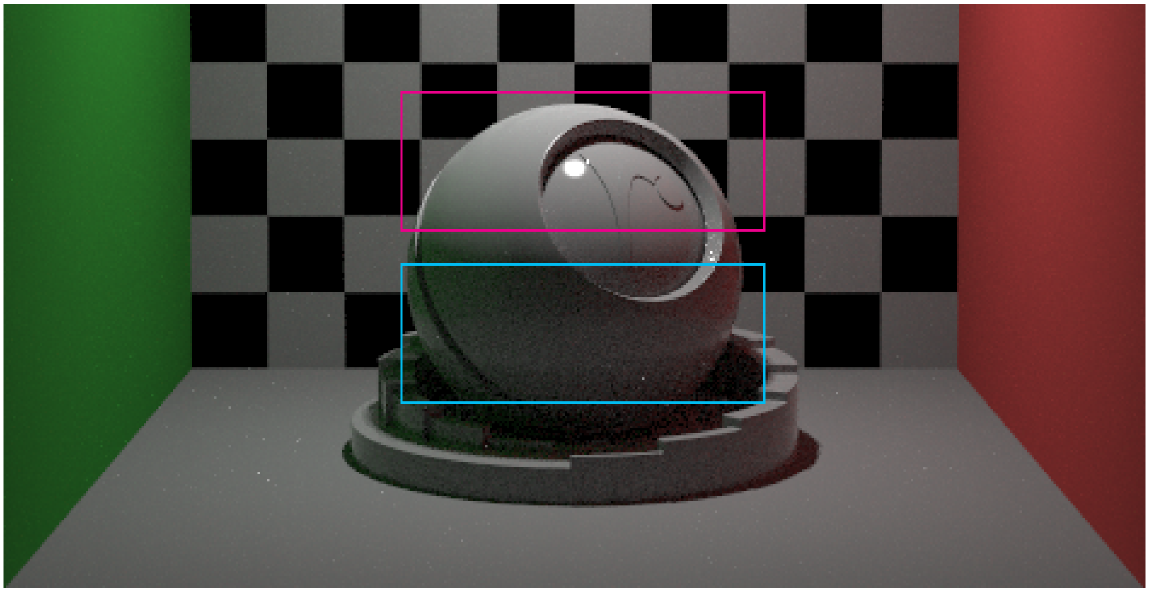
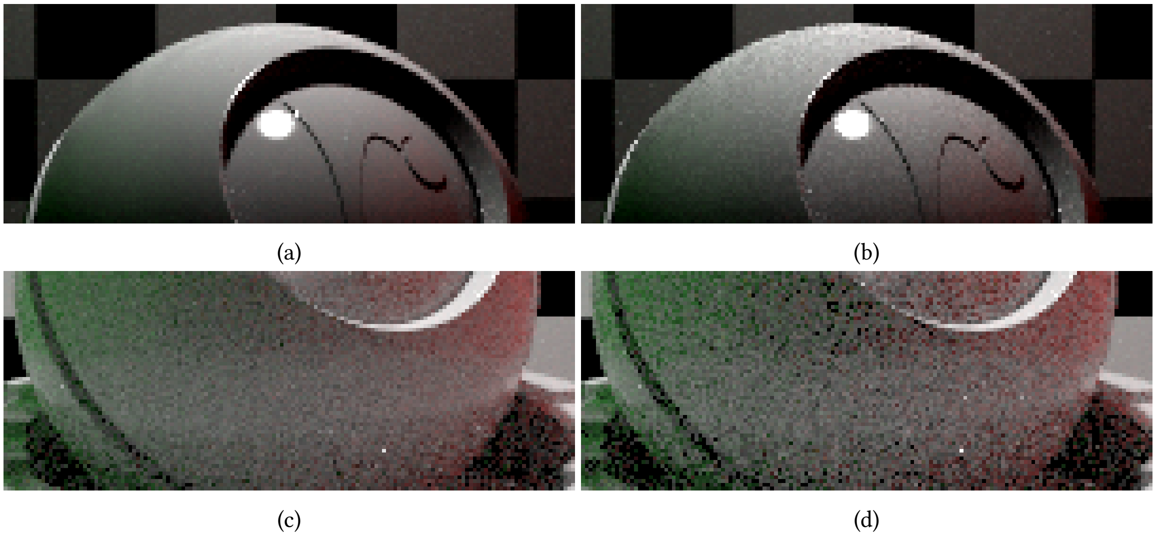


Figure 5: For this Cornell box render, we zoom in to the cropped locations.



(a)

(b)

(c)

(d)

Figure 6: We zoom in two areas of interest (and increase contrast) to show the effect of the sampling strategies discussed in section 3.1. Figures 6a and 6b demonstrate the difference between whether or not we use one-sample MIS after lobe picking. Figures 6c and 6d demonstrate the difference between whether or not we use Fresnel estimates for lobe picking.

1. Compute probabilities for each lobe, then select one according to these probabilities.
2. Use the chosen lobe to generate a sampling direction and compute the corresponding values and pdfs for the whole BSDF (all the lobes).

Resulting differences are shown in Figure 6a and Figure 6b. Going back to our specular-diffuse example, we first compute the pdfs of choosing the lobes, α_{spec} and α_{diff}.

If we don't use MIS, choosing specular results in:

$$\begin{aligned} dir_{\text{result}} &= dir_{\text{spec}} \\ value_{\text{result}} &= value_{\text{spec}} \\ pdf_{\text{result}} &= \alpha_{\text{spec}} \cdot pdf_{\text{spec}} \end{aligned}$$

Choosing diffuse results in:

$$\begin{aligned} dir_{\text{result}} &= dir_{\text{diff}} \\ value_{\text{result}} &= value_{\text{diff}} \\ pdf_{\text{result}} &= \alpha_{\text{diff}} \cdot pdf_{\text{diff}} \end{aligned}$$

If we use MIS, choosing specular results in:

$$\begin{aligned} dir_{\text{result}} &= dir_{\text{spec}} \\ value_{\text{result}} &= value_{\text{spec}} + value_{\text{diff}} \\ pdf_{\text{result}} &= \alpha_{\text{spec}} \cdot pdf_{\text{spec}} + \alpha_{\text{diff}} \cdot pdf_{\text{diff}} \end{aligned}$$

Choosing diffuse results in:

$$\begin{aligned} dir_{\text{result}} &= dir_{\text{diff}} \\ value_{\text{result}} &= value_{\text{spec}} + value_{\text{diff}} \\ pdf_{\text{result}} &= \alpha_{\text{spec}} \cdot pdf_{\text{spec}} + \alpha_{\text{diff}} \cdot pdf_{\text{diff}} \end{aligned}$$

Bidirectional renders add additional computations and constraints on the way a BSDF operates. We saw that the sampling is in two stages, so the final pdf is the product of the lobe choosing pdf, and the direction sampling pdf. When doing bidirectional renders, in addition to the pdfs (forward pdfs), we need to compute the reverse pdfs which represent the probabilities to choose the viewing directions based on the lighting directions. To compute step 1, we need $\mathbf{v}$ and $\mathbf{n}$, which, for the reverse PDFs become $\mathbf{l}$ and $\mathbf{n}$. Unfortunately, at this stage we don't know $\mathbf{l}$ yet since it will be generated at step 2. So the sampling for bidirectional renders has now 3 stages:

1. Compute probabilities for each lobe, then select one according to these probabilities.
2. Use the chosen lobe to generate a sampling direction and compute the corresponding values and pdfs for the whole BSDF (all the lobes).
3. Compute reverse probabilities for each lobe.

The corresponding evaluate functions are trivial in both cases, because for them $\mathbf{v}$ and $\mathbf{l}$ are known from the beginning.

3.2 Parameter Compositing

Historically, we implemented our surface shaders to support an arbitrary number of physically-based BSDFs and composited the result of the illuminated BSDFs. During *Monsters University* and *The Good Dinosaur*, we also leveraged coshaders in the Reyes rendering architecture for dynamically allocating BSDFs as needed (see again [HV13]).

In contrast, `PxrSurface` has a fixed number of BSDFs. The material parameters are precomposited in parameter space and illumination is run on these fixed BSDFs. In practice, because materials can be layered in abstractions for end-user artists, our current system is a good compromise between the flexibility of production and the performance and predictability of the fixed number of BSDFs.

The input parameters are layered upstream with a pattern shader that comprises of multiple "material layers". In Pixar language, a "material" is an abstraction of a type of substance that can be identified. For example "Skin" is a material and a "Dirt" material can be layered on top of "Skin". Materials can be daisy-chained with these pattern shaders. Each material layer's input and output are standardized such that they can be composited with any other material layer. For example the same "Dirt" material layer can be used to cover a "Metal" material layer or a "Plastic" material layer. The final input parameter results are fed into `PxrSurface` to be illuminated in the renderer.

To use a concrete example, Figure 7 is a render of McQueen from *Cars 3* and Figure 8 is the composition diagram of its metal flake car paint. Car paint is a complex material. It is composed of a metal flakes medium with a clear polyurethane layer on top so there are one diffuse and two specular lobes in this composite material. The energy not reflected by the clearcoat layer is transmitted down through to the undercoat metallic layer and then reflected back out through the clearcoat layer again. Further more, we have sticker layers, dirt-and-grime layers, mud-layers that composites on top of the car paint layer. This can get fairly complicated.

The trick here is to think of each of the visual components seperately. Car Paint itself is composed of a regular diffuse, a specular BRDF and a clearcoat specular BRDF. A set of combined diffuse and specular BRDFs model the solid paint (that contains the metal flakes). The clearcoat specular models the clear, shiny polyurethane layer, which covers the base paint. The next layer is a metallic sticker layer. The metallic sticker is fully opaque and is a simple material with one specular lobe. When we apply the metallic sticker layer on top of the base paint layer, it makes sense to composite it over the specular layer, and kill the presence of the clearcoat layer. We want to composite metallic specular of the sticker over metallic specular and the car paint because of their simularity in material properties. In this case, sticker does not have a clearcoat component.

However for the dirt layer, we need to composite diffuse, clearcoat and specular BRDF parameter values, even though we normally don't associate the dirt layer itself as being very specular. Since the coverage is not 100% opaque, we would like to modify the roughness values of the specular and clearcoat lobes underneath. For this we add low intensity but high roughness specular parameters and composite these parameters over the other layers. For most of the parameters, we can composite with traditional [PD84] "Over" operations. The fact that the "Dirt" material modified color, specular intensity and roughness of "Metallic Sticker" and "Car Paint" makes the final illumination look like the dirt is on top. Thus the final BSDF input parameters are just the aggregate data from the operations that happened upstream of `PxrSurface`.

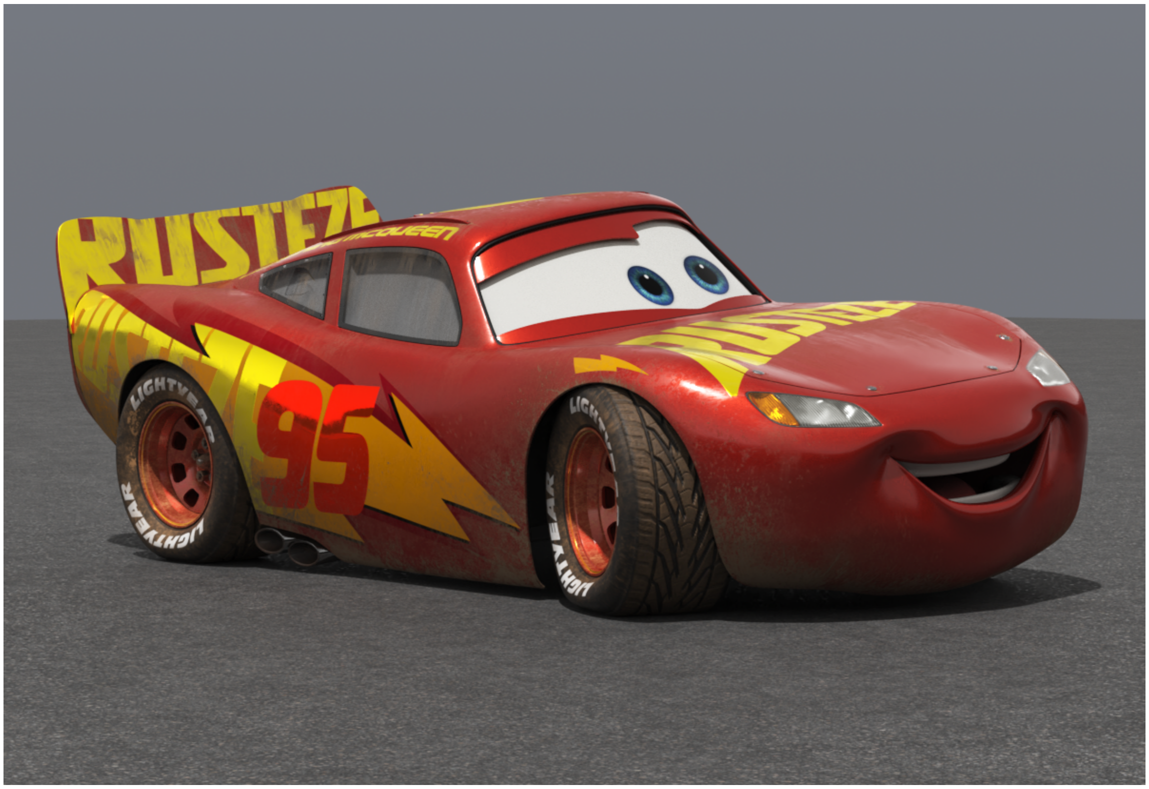


Figure 7: McQueen's shading from *Cars 3* ©Disney/Pixar 2017 has many layers of materials. Note that the dirt composites over the chrome wrap logos and both composite over the base paint layer. At render time the material parameters are composited to run over the same BRDFs.

One additional note is that we draw distinctions between clearcoat, specular and roughspecular BRDFs. Even though fundamentally they are based on the same Beckmann or GGX models that can be set with a variety of roughness and Fresnel ranges, we make some conceptual distinction for practical reasons. "Clearcoat" is meant to be the dielectric interface at the top of a material. It usually has a very low roughness that is lower than 0.1, unless it is moderated with a dusty material on top. The "specular" BXDF usually describes a rougher material. Usually it somewhere between 0.1 and 0.4. This can be the top interface of materials such as plastic or simple metal, or a sub-layer of a compound material such metal flakes in car paint, or the fibrous layer of varnished wood. Some complex materials can require longer tails in their BXDF profiles. Although GGX provides us with one way to achieve this, sometimes this rougher specular needs to be art-directed. We provide an additional roughspecular BRDF for that very purpose. This "roughspecular" BRDF is usually rougher than "specular", with roughness in the range that is greater than 0.4. It can be used by itself or in conjunction with the regular "specular" BRDF to create appealing highlights. This mental model makes the layering of material types more intuitive to the end users and we achieve better visual results.

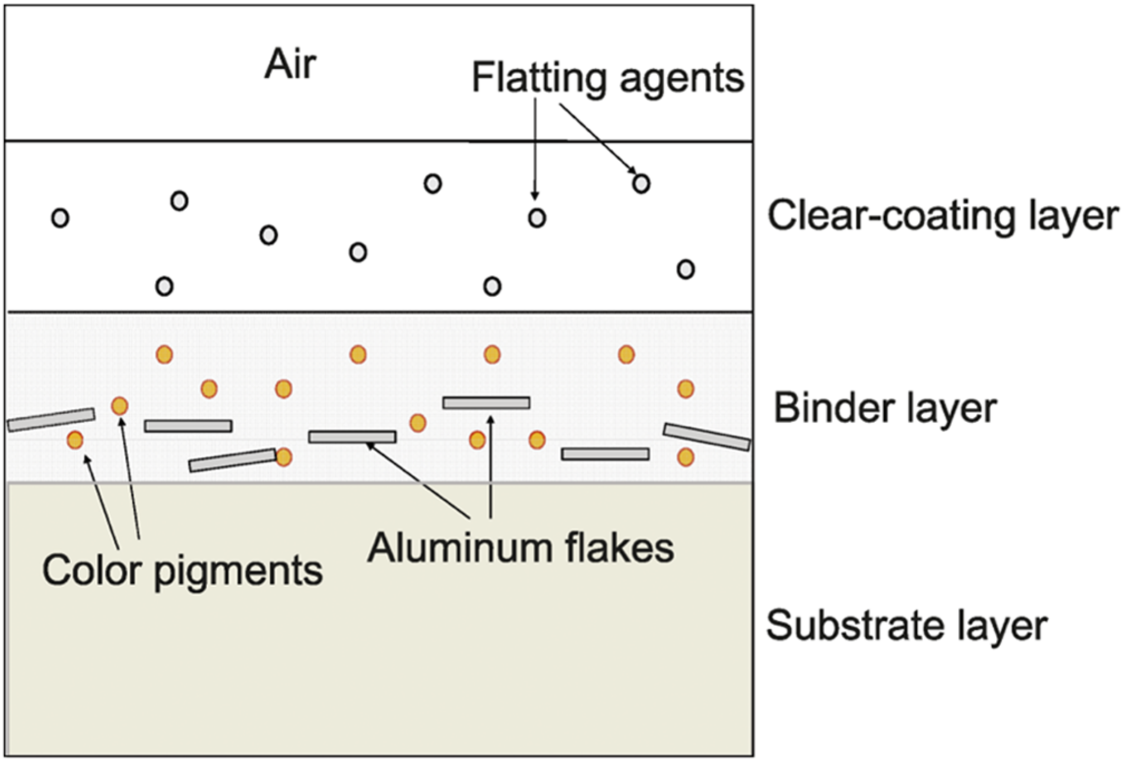
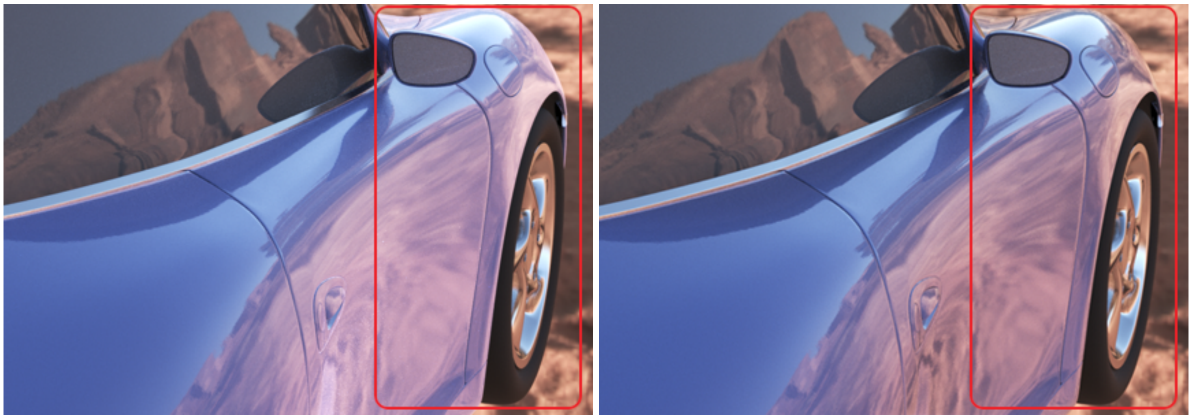


Figure 8: Physical car paint layering diagram

3.3 Energy Compensation

When layering materials, it is important to account for the proper amount of energy absorbed and reflected by the interfaces between the layers. Light from incident rays reflects off of the interface on top before it is transmitted to the BSDF below. This is a problem tackled by [Jak15]. Unfortunately, a physically based general solution that also allows texturing is still an unsolved problem in computer graphics. We obtained emperical results for a common special case, where the clearcoat interface for the first dielectic interface is accounted for and we fit our simulated results to a function. This compensates for the energy reflected by the Fresnel effect of the clearcoat layer. See visual results in Figure 9 of what we refer to as Fresnel Energy Compensation. This feature is not strongly enforced. We left the presence of this "Energy Compensation" effect as adjustable parameters to fit to end-user's artistic sensibilities. Clearcoat sitting on top can attenuate all lobes, via an explicit "Clearcoat Energy Compensation" parameter, while the two speculars can attenuate all lobes but clearcoat via "Specular Energy Compensation".



(a) Without energy compensation

(b) With energy compensation

Figure 9: Fresnel Energy Compensation for the clearcoat layer. We note the differences in the reflection in the highlighted areas of this figure. The image on the left looks washed out and "dusty" without energy compensation. Where the right image with energy compensation looks more physically realistic and glossy.

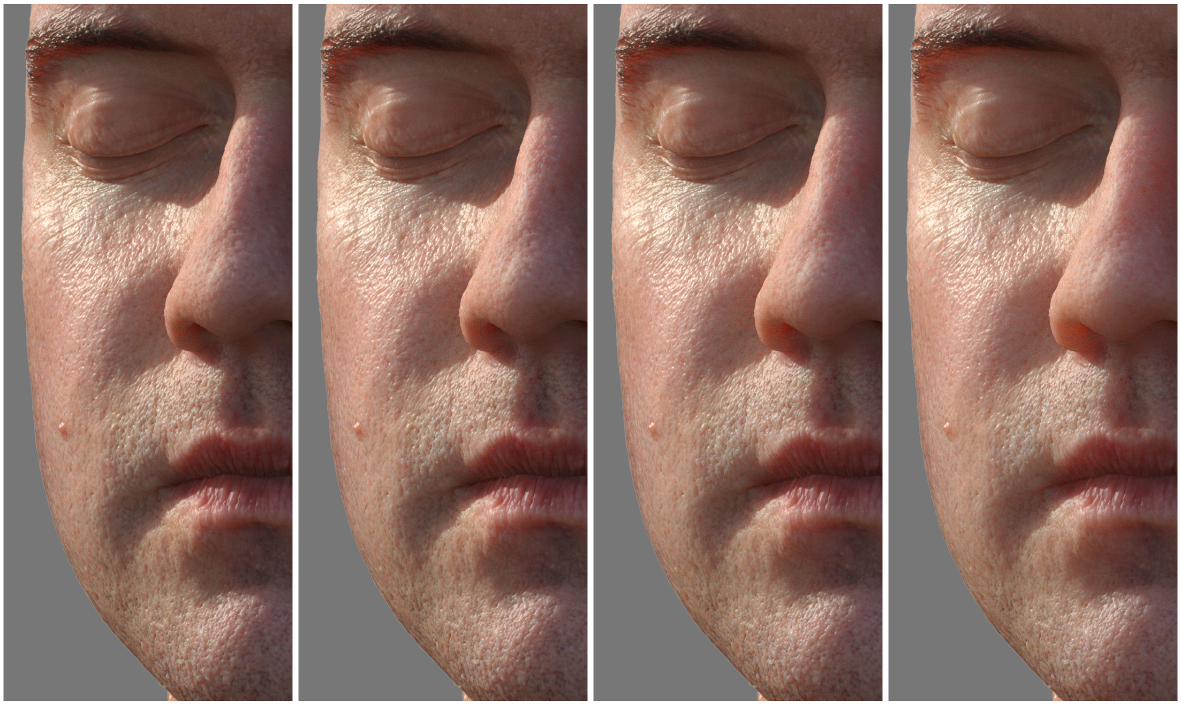
4 SUBSURFACE SCATTERING

Another example is that PxrSurface ships with "Jensen", "D'Eon" and "Burley" diffusion profiles (for reference, the latter is described in [Chr15]) . Additionally, we implemented a novel subsurface approach using path-traced volumetric scattering which represents a significant advancement. It captures zero and single scattering events of subsurface scattering implicit to the path-tracing algorithm. The user can also adjust the phase-function of the scattering events and change the extinction profiles: Figure 10 shows the effect of manipulation the new bleed parameter in the non-exponential profile. It comes with the standardized color inversion features for intuitive albedo input. To the best of our knowledge, this is the first commercially available rendering system to model these features and the rendering cost is comparable to classic diffusion subsurface scattering models. The implementation details of this technique is beyond the scope of this course note. Please refer to [WVH17] for details.

The parameters of various BSSRDF models internal to the algorithms themselves are often physically based and hard to understand. All of our subsurface models have albedo inversion to make it easier for end users (see [Her12] for what this means under the hood). That is, the end user can always specify an albedo color and a Diffuse Mean Free Path. An additional benefit of this approach is that when the DMFP becomes shorter than the pixel width, we can safely switch from a BSSRDF to a simple Lambert BRDF to achieve a dynamic LOD of our rendering methods without any user intervention.

4.1 Subsurface Models

In our production environment, the end user have access to all four subsurface models for specific art directions. The visuals provided by the "Jensen" and "D'Eon" models are softer (see Figures 11a and 11b). Being pure diffusion based models, we can achieve highly gummy-type of looks with these models. And if sharp shadowing



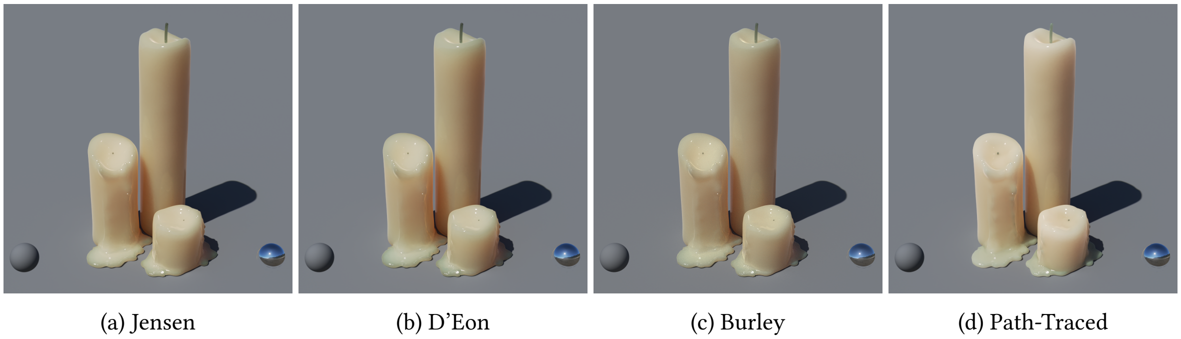
(a) bleed=0.0

(b) bleed=0.1

(c) bleed=0.3

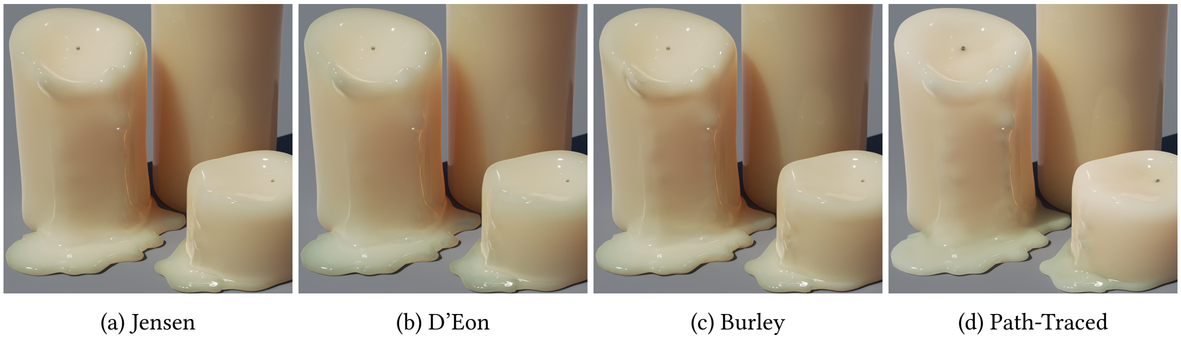
(d) bleed=0.5

Figure 10: Path-traced subsurface with non-exponential profile. We vary the “bleed” prameter. A “bleed” of 0 is equivalent to the exponential profile. Head data courtesy of Infinite Realities via Creative Commons.



(a) Jensen (b) D'Eon (c) Burley (d) Path-Traced

Figure 11: Visual differences between the provided subsurface illumination models in `PxrSurface`. Path-traced includes zero and single scatter and is thus brighter.



(a) Jensen (b) D'Eon (c) Burley (d) Path-Traced

Figure 12: Subsurface illumination models provided by `PxrSurface`. Note the behavior of the scattering in the shadowed areas.

termination is desired, we can use the "Burley" profile (Figure 11c). Refer to Figure 12 for a closer view over the shadow area. The "path-traced" models (Figure 11d) are fairly new for our productions. The advantage of these techniques is that we can eliminate much of the visual artifacts present in the diffusion-based approaches, because they do not make the regular assumptions of the geometry being a locally flat semi-infinite medium.

4.2 Notes on Bidirectional Renders

Subsurface sampling is tricky with bidirectional renders, because the diffusion model is not symmetrical. Although it is practically free to compute reverse pdfs for sampling lobes (especially if we do forwards and backwards at once). This is not true for SSS. Diffusion models usually uses the entry point to perform albedo inversion and completely ignores the parameters at the exit point. There are techniques, such as [SHK17], that try to solve the non symmetrical nature of SSS, but they usually require special features from the renderer (in this paper's case, a way to quickly average σ_s and σ_t coefficients). We could in theory compute the adjoint sampling once we found the exit point, but that would easily double the cost of SSS, which is already higher than normal lobes. And this is all the more true if the SSS sampling is using advanced sampling techniques like [VHC16]. We are not satisfied by this limitation, and hopefully in the future, renderers will be fast enough to treat SSS like any normal multi-scatter volumetric effect.

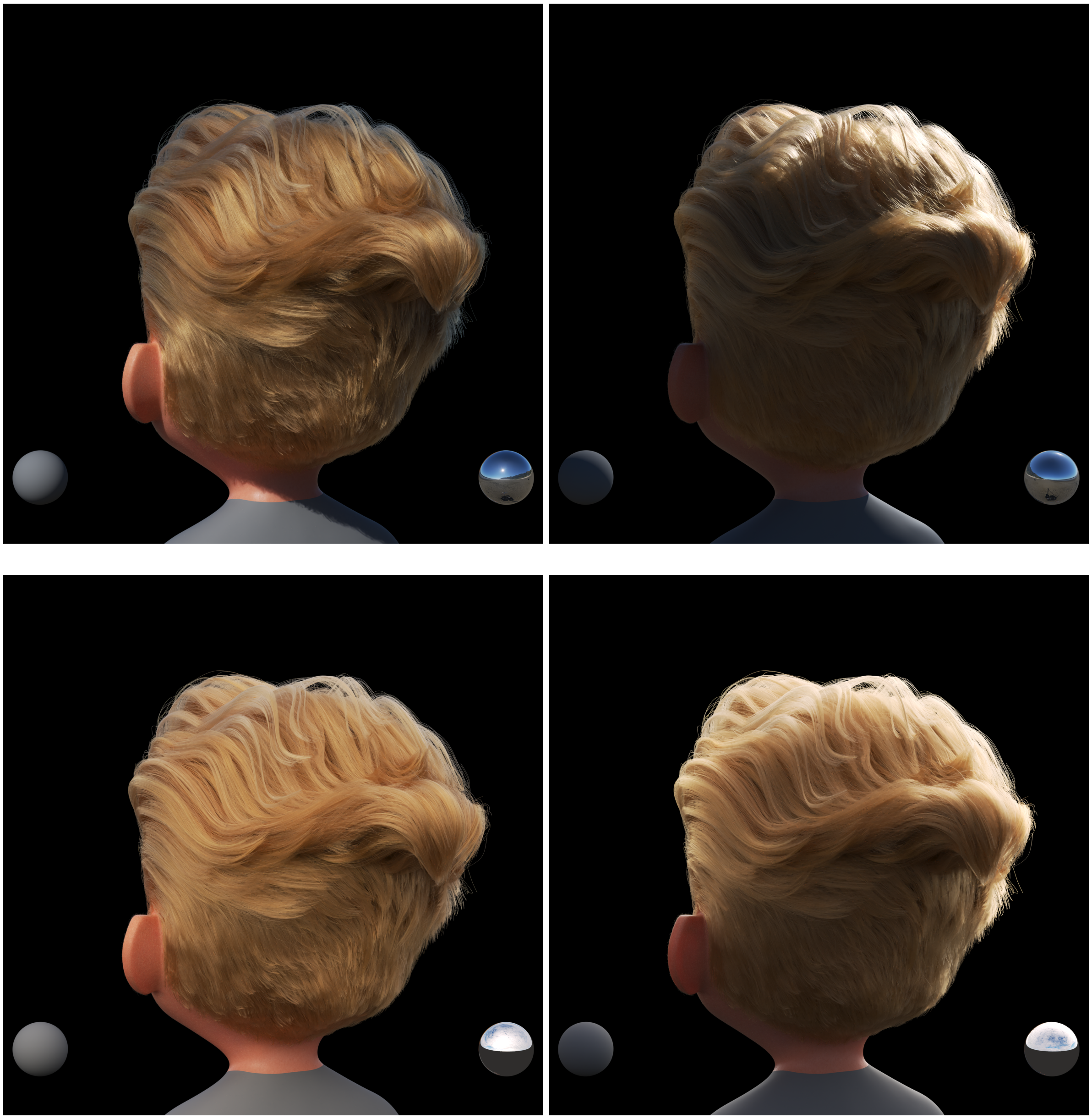
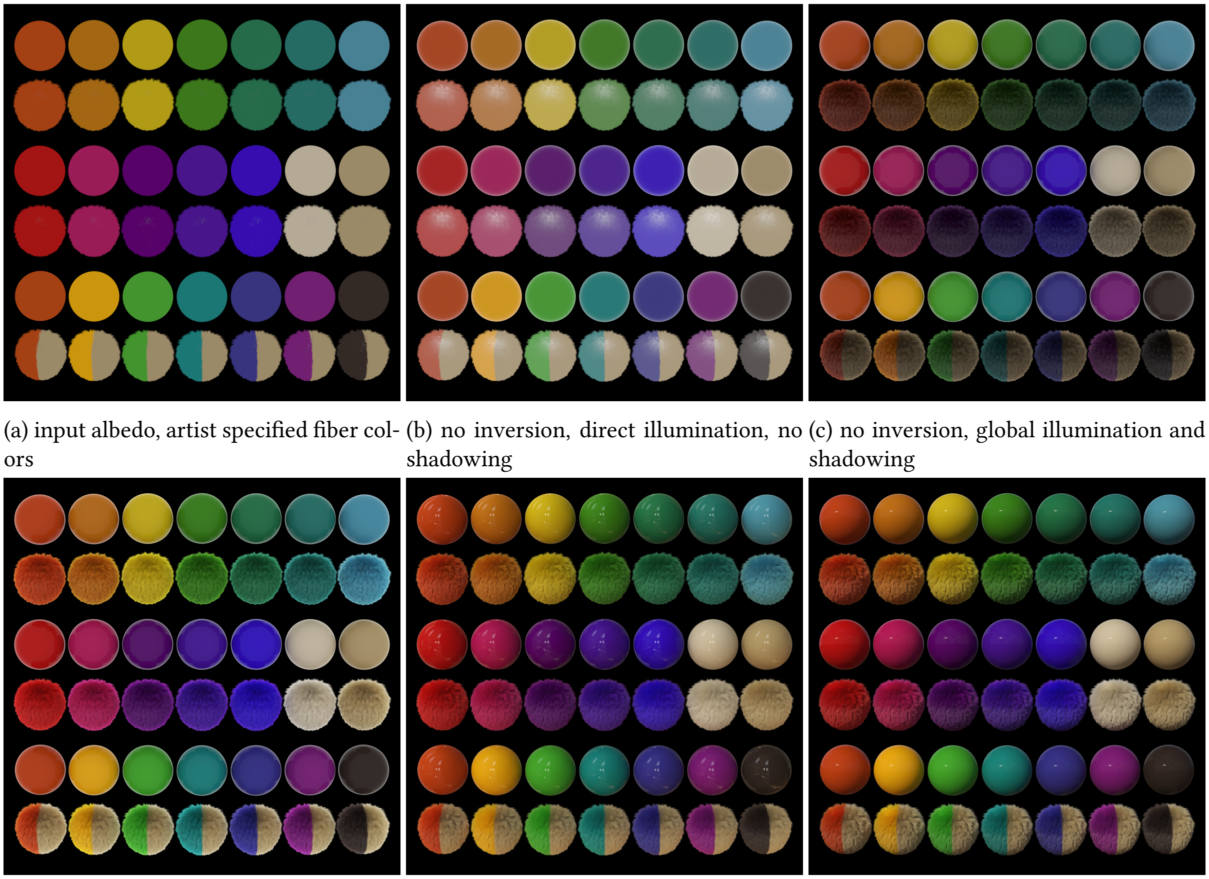


Figure 13: Historically, blond hair has been harder to achieve than darker hair types in shading. `PxrMarschnerHair` can achieve the blond look effectively with path-traced multiple scattering

5 PXRMARSCHNERHAIR

PxrMarschnerHair implements [Mar+03]'s seminal hair illumination model with importance sampling. We refer to our tech memo [Pek+15] for full details. PxrMarschnerHair also accounts for the residual energy left after the R, TT, TRT and Glints lobes through an additional diffuse lobe (which we recommend to set at type "Zinke" rather than "Kajiya"). This hair shader can reproduce dark and light hair (recent example of blond hair in Figure 13) and animal fur effectively in a path-traced production context.



(a) input albedo, artist specified fiber colors

(b) no inversion, direct illumination, no shadowing

(c) no inversion, global illumination and shadowing

(d) with inversion, white light

(e) with inversion, interior lighting

(f) with inversion, exterior lighting

Figure 14: Volumetric hair albedo inversion test grid

Unlike surface shading, volumetric scattering and shadowing from fiber to fiber changes the perceived lightness, hue and saturation of a groom. To illustrate this problem see Figure 14. Figure 14a shows the direct input albedo for TT, TRT and GLINTS and the Residual Diffuse. The combined BXDF is energy–balanced based on the first principles described by [Mar+03]. Figure 14b is illuminated by a white furnace environment (i.e. a light source emitting 1 everywhere) without shadowing. Note that the R term is white, hence the slightly more desaturated results. Also note that the Fresnel effect is illustrated in Figure 14b, where R is more intense near the “north pole” of the groom spheres. However, with shadowing and multiple scattering the hair “volume” appears much darker and more desaturated in Figure 14c and the perceived hue is also altered.

Similar in principle to [Chi+15], an inversion method was empirically derived by rendering hair examples with various albedo inputs. We back-solve for a curve that would invert the input perceptual albedo with an desired output fiber albedo. Figures 14d, 14e and 14f demonstrate that this inversion scheme provides predictably illuminated results under various lighting conditions.

Related to this, in `PxrSurface`, we utilized the R term of `PxrMarschnerHair` and created a fuzz BXDF [Ren16b] for inexpensive, illumination-based fuzz. Generally used to simulate short fuzzes in shading, we can render these surfaces without creating additional grooming geometries. This BRDF uses the surface normal instead of the the curve tangent as it geometric input. A cosine term is multiplied into the result so there is no abrupt terminator transitions.

6 CONCLUSION

In these course notes, we discussed Pixar’s two primary shaders, `PxrSurface` and `PxrMarschnerHair`. We provided detailed descriptions, the design principles and some usage examples for both shaders. These two shaders have been a work in progress for many years and contain advancements in many areas, some of which we highlighted. Although the specific implementation details of each are beyond the time and space that is available here, many of these are available to readers and are referenced throughout the document.

At the time of this course, Pixar is in the process of finishing *Coco*. This feature film will be the third to be rendered entirely with our physically based shading system implemented for the `RIS` path tracer. The previous two films were *Finding Dory* and *Cars 3*. There were also two companion short films completed with `RIS`: *Piper* and *Lou*. Going back to 2010, our shading and sampling technology finds its roots in the work we did for *Monsters University*, *Inside Out* and *The Good Dinosaur*, along with the shorts *The Blue Umbrella* and *Volcano* (all implemented under the `REYES` architecture). Production-wise, this has made the behavior of lighting and shading more consistent across assets, shots and different productions. These advancements have provided our artists new and exciting tools to create ever more sophisticated visuals in our films.

Most importantly, we would like to thank our colleagues in production, tools, research and the ®Renderman team for their work in making everything possible. Film making is a collaborative effort, and we are so thankful for the opportunity to work with these amazingly talented people on a day-to-day basis to create great art and technology together. We would also like to thank the executive team for their support and leadership in making this entire process possible and for fostering this amazingly creative environment.

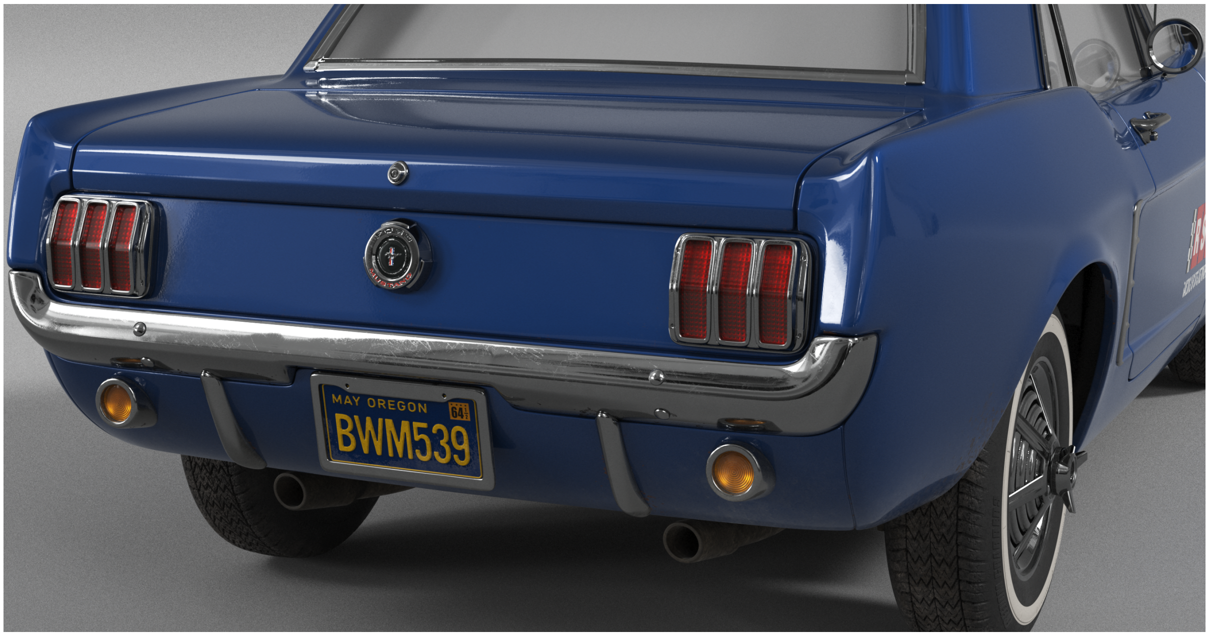


Figure 15: One production shading example on *Cars 3* ©Disney/Pixar 2017

REFERENCES

[Chi+15] M. J.-Y. Chiang, B. Bitterli, C. Tappan, and B. Burley. "A Practical and Controllable Hair and Fur Model for Production Path Tracing". In: *ACM SIGGRAPH 2015 Talks*. SIGGRAPH '15. Los Angeles, California: ACM, 2015, 23:1–23:1. ISBN: 978-1-4503-3636-9. DOI: 10.1145/2775280.2792559. URL: http://doi.acm.org/10.1145/2775280.2792559.

[Chr15] P. H. Christensen. "An Approximate Reflectance Profile for Efficient Subsurface Scattering". In: *ACM SIGGRAPH 2015 Talks*. SIGGRAPH '15. Los Angeles, California: ACM, 2015, 25:1–25:1. ISBN: 978-1-4503-3636-9. DOI: 10.1145/2775280.2792555. URL: http://doi.acm.org/10.1145/2775280.2792555.

[Dup+13] J. Dupuy, E. Heitz, J.-C. Iehl, P. Poulin, F. Neyret, and V. Ostromoukhov. "Linear Efficient Antialiased Displacement and Reflectance Mapping". In: *ACM Transactions on Graphics*. Proceedings of Siggraph Asia 2013 32.6 (Nov. 2013), Article No. 211. DOI: 10.1145/2508363.2508422. URL: https://hal.inria.fr/hal-00858220.

[Fon+17] J. Fong, M. Wrenninge, C. Kulla, and R. Habel. "Production-Volume Rendering". In: *Production-Volume Rendering, ACM SIGGRAPH 2017 Courses*. Los Angeles, CA, USA: ACM, 2017.

[HD14] E. Heitz and E. D'Eon. "Importance Sampling Microfacet-Based BSDFs using the Distribution of Visible Normals". In: *Computer Graphics Forum* 33.4 (July 2014), pp. 103–112. DOI: 10.1111/cgf.12417. URL: https://hal.inria.fr/hal-00996995.

[Her12] C. Hery. *Texture mapping for the Better Dipole model*. https://graphics.pixar.com/library/TexturingBetterDipole/. Nov. 2012.

[HKL14] C. Hery, M. Kass, and J. Ling. *Geometry into Shading*. https://graphics.pixar.com/library/BumpRoughness/. Apr. 2014.

[HV13] C. Hery and R. Villemin. “Physically Based Lighting at Pixar”. In: *Physically Based Shading in Theory and Practice, ACM SIGGRAPH 2013 Courses*. Anaheim, CA, USA, 2013. URL: http://graphics.pixar.com/library/PhysicallyBasedLighting/index.html.

[HVS17] C. Hery, R. Villemin, and T. Schmidt. “Emeryville: where all the fun light transports happen!” In: *Path Tracing in Production: Part 2 - Making Movies, ACM SIGGRAPH 2017 Courses*. Los Angeles, CA, USA: ACM, 2017.

[Jak15] W. Jakob. “layerlab: A computational toolbox for layered materials”. In: *SIGGRAPH 2015 Courses*. SIGGRAPH ’15. New York, NY, USA: ACM, 2015. DOI: 10.1145/2776880.2787670.

[Mar+03] S. R. Marschner, H. W. Jensen, M. Cammarano, S. Worley, and P. Hanrahan. “Light Scattering from Human Hair Fibers”. In: *ACM Trans. Graph.* 22.3 (July 2003), pp. 780–791. ISSN: 0730-0301. DOI: 10.1145/882262.882345. URL: http://doi.acm.org/10.1145/882262.882345.

[OB10] M. Olano and D. Baker. “LEAN Mapping”. In: *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*. I3D ’10. Washington, D.C.: ACM, 2010, pp. 181–188. ISBN: 978-1-60558-939-8. DOI: 10.1145/1730804.1730834. URL: http://doi.acm.org/10.1145/1730804.1730834.

[PD84] T. Porter and T. Duff. “Compositing Digital Images”. In: *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’84. New York, NY, USA: ACM, 1984, pp. 253–259. ISBN: 0-89791-138-5. DOI: 10.1145/800031.808606. URL: http://doi.acm.org/10.1145/800031.808606.

[Pek+15] L. Pekelis, C. Hery, R. Villemin, and J. Ling. *A Data-Driven Light Scattering Model for Hair*. https://graphics.pixar.com/library/DataDrivenHairScattering/. Feb. 2015.

[Ren16a] Renderman. *PxrMarschnerHair*. https://rmanwiki.pixar.com/display/REN/PxrMarschnerHair. 2016.

[Ren16b] Renderman. *PxrSurface*. https://rmanwiki.pixar.com/display/REN/PxrSurface. 2016.

[SHK17] H. Sone, T. Hachisuka, and T. Koike. “Parameter Estimation of BSSRDF for Heterogeneous Materials”. In: *Eurographics 2017 Short Papers*. 2017.

[VG95] E. Veach and L. J. Guibas. “Optimally Combining Sampling Techniques for Monte Carlo Rendering”. In: *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’95. New York, NY, USA: ACM, 1995, pp. 419–428. ISBN: 0-89791-701-4. DOI: 10.1145/218380.218498. URL: http://doi.acm.org/10.1145/218380.218498.

[VH12] R. Villemin and C. Hery. *Porting RSL to C++*. https://graphics.pixar.com/library/PortingRSLtoCpp/. Aug. 2012.

[VH15] R. Villemin and C. Hery. “RenderMan RIS”. In: *Art and Technology at Pixar, from Toy Story to Today, ACM SIGGRAPH Asia 2015 Courses*. SA ’15. Kobe, Japan: ACM, 2015, 5:1–5:89. ISBN: 978-1-4503-3924-7. DOI: 10.1145/2818143.2818155. URL: http://doi.acm.org/10.1145/2818143.2818155.

[VHC16] R. Villemin, C. Hery, and P. Christensen. *Importance Resampling for BSSRDF*. https://graphics.pixar.com/library/Resampling/. May 2016.

[WVH17] M. Wrenninge, R. Villemin, and C. Hery. *Path Traced BSSRDF*. https://graphics.pixar.com/library/PathTracedSubsurface/. July 2017.