

# Using USD Shading to Provide The "Extra" Touch on Incredibles2

Kiki Poh  
Michael Kilgore  
Pixar Animation Studios

Tom Wichitscripornkul  
Gary Monheit  
Pixar Animation Studios



Figure 1: Background characters in Incredibles 2 ©Disney / Pixar.

## ABSTRACT

A shading pipeline which allows us to be able to provide the immediate basic shading for the BG human skin and hair at creation, while adapting and evolving to show needs and artistic direction. Along with our new hair system in **Presto**, we also had new usd applications such as **usdview** were widely used from simple asset context viewing, though all attribute and parameter inspecting/debugging processes. For the first time, groom, model, rig, and simulation artists and were working together on the same platform as our Animators. And while shading wasn't in **Presto**, Universal Scene Description (USD) wove all things together in a most accessible and inspect able format. This approach allowed us to create significant amounts of background characters within a short time frame.

## CCS CONCEPTS

• **Computing methodologies** → **Shading; Grooming; USD Shading;**

## KEYWORDS

shading, grooming, USD shading, hair transfer

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SIGGRAPH '18 Talks, August 12-16, 2018, Vancouver, BC, Canada

© 2018 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-5820-0/18/08...\$15.00

<https://doi.org/10.1145/3214745.3214765>

## ACM Reference Format:

Kiki Poh, Michael Kilgore, Tom Wichitscripornkul, and Gary Monheit. 2018. Using USD Shading to Provide The "Extra" Touch on Incredibles2. In *Proceedings of SIGGRAPH '18 Talks*. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3214745.3214765>

## 1 INTRODUCTION

Since the first Incredibles, more background characters and more variation has always been requested. So for the cities of Incredibles 2, we created throngs of stylized people. Here's our new approach to the challenge of wide asset deployment of shading and grooming for BGs.

Our new shading pipeline allows us to provide the immediate basic shading for the BG human skin and hair by default, as soon as the asset is created. Along with our new hair system in **Presto**, we also had new usd applications such as **usdview** were widely used for simple asset context viewing, though all attribute and parameter inspecting/debugging processes. For the first time, groom, model, rig, and simulation artists and were working together on the same platform as our Animators. And while shading wasn't in **Presto**, Universal Scene Description (USD) wove all things together in a most accessible and inspect able format.

## 2 LEVERAGING USD

Sharing topology on characters offers obvious advantages for modeling and rigging, but shading and grooming are usually only able to make assumptions (consistent parameterization, ability to transfer

attributes between meshes). In reality these are fragile- for example a point ordering change can make it difficult to realize wins. USD allowed us to instead *guarantee* these things by referencing a fully shaded proto asset (Human/HumanHair). We could add two lines to the leaf level assets' **base.usd** files, and let that opinion reference all the materials, bindings, and any other attributes in the **Human.usd**. There is no transferring, the receiving assets have only the same topology with the points moved. This meant less shading attribute protection for the model/rig team with changes.

## 2.1 Master Shading

The master asset and all contained meshes were shaded similarly to recent shows (via **Katana**, A la **Coco**), albeit with some of the material bindings paths to the prims wildcarded. Concern was also taken to apply the same treatment to things like trace groups, which needed to be unique to the final in shot needs. Maintaining this master asset file for updates, debugging, and additional feature deployment was much easier than inspecting numerous individual assets.

## 2.2 Customization, Overrides, and Flow

Base shading established, each leaf level human had 8 variants. Each had a unique look, and needed customization per variant. Flow was used to load an asset, and dial the components of each variant in the same session, real time. This saved usd file layered overrides on the base shading. When each asset is compiled, unique variants are created. And index linked to the shading variant name explicitly selected the shading done in Flow. Since all variant names were consistent across leaf assets, this was all done in each Katana file via livegroup, maintaining consistency in naming. This also allowed for sub-batch attribute setting, like eyeliner darkening, makeup variation on teens females vs adult males.

```

1 #usda 1.0
2 {
3   "Generated by UsdBake"
4   defaultPrim = "BGildaHair"
5 }
6
7 over "BGildaHair" (
8   active = true
9   add references = </BGildaHair_defaultShadingVariant>
10 )
11 {
12 }
13
14 over "BGildaHair_defaultShadingVariant" (
15   add references = @HumanHair/usd/HumanHair.shading.usd@</HumanHair>
16 )
17 {
18 }

```

Figure 2: USD Shading Reference ©Disney / Pixar.

## 3 NEW HAIR SYSTEM IN PRESTO

On **Incredibles 2**, we adopted a new hair system in **Presto**. This new hair tool was structured on top of our existing modeling rig. It gave us a completely accurate view of hair in a model with rigging while we groom. We also worked closely with rigging department to create symmetry within the hair rig. This allowed us to groom symmetric hair layers like brows and eyelashes using automatic symmetric class instancing which is part of **Presto's** prim-based modeling system.

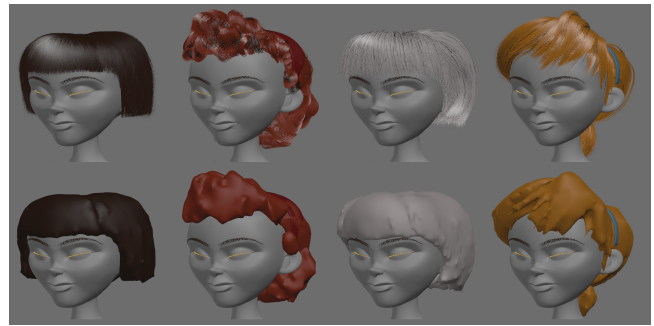


Figure 3: Hair Variants with Generated Hair Shells ©Disney / Pixar.

## 3.1 New features

Each of our background human hair models contain four different hair model variants. The hair transfer tool allows us to transfer hair layers across different models. This tool transfers the hair styles and the hair density textures. Positioning and shaping of the hair guide curves is done automatically with a rig that warps the curves onto the new mesh. In **Presto**, a prim can be defined as a class or an instance. Any hair layer can inherit from a class prim, dynamically. This allows sharing of attribute values and consistency across many hair layers. Having the advantage of editing hair in the same platform with rigging and animation, we set the correct hair color display per model so animators can see the accurate hair color. We also create a hair shell which is the simplified hair mesh generated from hair guide curves. This hair shell is required for animation and crowds scenes to increase frame rate playback speed.

## 3.2 USDView for Debugging

On **Incredibles 2**, we are widely using **usdview** for debugging purposes. Each of our model will get built into a USD file that stores all the information with shading and grooming. **usdview** allows us to do a very quick diagnosis of some of the problems we have before the shots render. Inside **usdview**, we can see the shading composition and how it was layering with **Katana**, **Flow** and referencing. We are also able to see if the simulated hair guides are moving correctly with the model. We also can setup a quick view and render through **usdview** into our global render server.

## 3.3 Conclusion

We presented several changes to our workflow that allowed us to create more stylized characters that are still believable in a physically shaded and lit world. The benefit of having the grooms, modeling and animation on the same software is a huge win. This workflow is much more efficient than what we used to have. Shading in USD give us a lot of flexibility on shading. The ability to be able to layer the different shading info allows artists to shade using the platform they are familiar with.