

# The Technical Art of Uncharted 4

Waylon Brinck   Andrew Maximov   Yibing Jiang



**Figure 1:** *Uncharted 4* rendering tech is capable of representing diverse environments with a visually-consistent style.

## Abstract

Rendering technology for *Uncharted 4* underwent a major overhaul in order to fit the needs of the game and the power of the PlayStation 4. The design requirements specified a huge variety of locations and looks, with many one-off features for particular scenes. The art direction called for a more “painterly” look, while the art team wanted the predictability of physically-based rendering. And everyone wanted to push the graphical fidelity as far as possible.

To help achieve this, we opened up the shading code to technical artists. Core algorithms and framework were still written by the rendering engineers, but technical artists were able to develop technology that was robust, physically-based, and user-friendly. Some examples of this tech are presented here.

## 1 Rendering and Shading

Many features requested by the art team initially sounded like one-off hacks. On closer analysis, these features were able to be implemented in a physically-plausible way that worked consistently across the whole game.

- Color Grading Lookup Tables (LUTs) operate in HDR space. Our tonemapping curve is also baked into the HDR LUT, rather than being parametric.
- We implemented a micro-shadowing BRDF term for analytic lights that uses ambient occlusion as input data.
- Fog color is based on a texture, which allows the fog to match the sky color at the horizon. We use high (blurry) mips closer to the camera to hide obvious texture detail.

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. Copyright is held by the owner/author(s).

SIGGRAPH '16 Talks, July 24-28, 2016, Anaheim, CA,

ACM 978-1-4503-4282-7/16/07.

<http://dx.doi.org/10.1145/2897839.2936731>

- Special-case materials like moss, puddles, and glass are created by combining together simpler physically-based building blocks.

## 2 Vertex Shader Processing

General-purpose per-vertex processing is still fairly new to runtime graphics, but it opens a lot of doors for novel creative features and optimization techniques.

Examples include foliage motion, hair, cloth simulation, object population, morph targets, sub-object billboards and imposters, per-vertex keyframe animation and whole lot more.

By building a robust framework and providing a few key data sources, many of these diverse vertex-processing problems can be solved in a consistent way.

## 3 Character Shading

Here we present two areas where technical art helped our character team achieve fidelity, consistency, and streamlined production.

- Creating Volumetric feeling of hair, within performance constraints. We combine baked shadows with in-game shadows to approach the look of deep opacity shadow maps. We also use a cheap, convincing way to simulate SSS on the hair.
- We used shader packages like fabric, skin and eyes to add more details and realism to our characters and also speed up the production pipeline.