Grip and Filament: A USD-Based Lighting Workflow

Steve Agland
Animal Logic
stevea@al.com.au

Simon Bunker
Animal Logic
simonbu@al.com.au

Jakub Jeziorski
Animal Logic
jakubj@animallogic.ca

Manuel Macha
Animal Logic
manuelm@al.com.au

Eoin Murphy
Animal Logic
eoin.murphy@gmail.com

Francesco Sansoni
Animal Logic
francescos@al.com.au

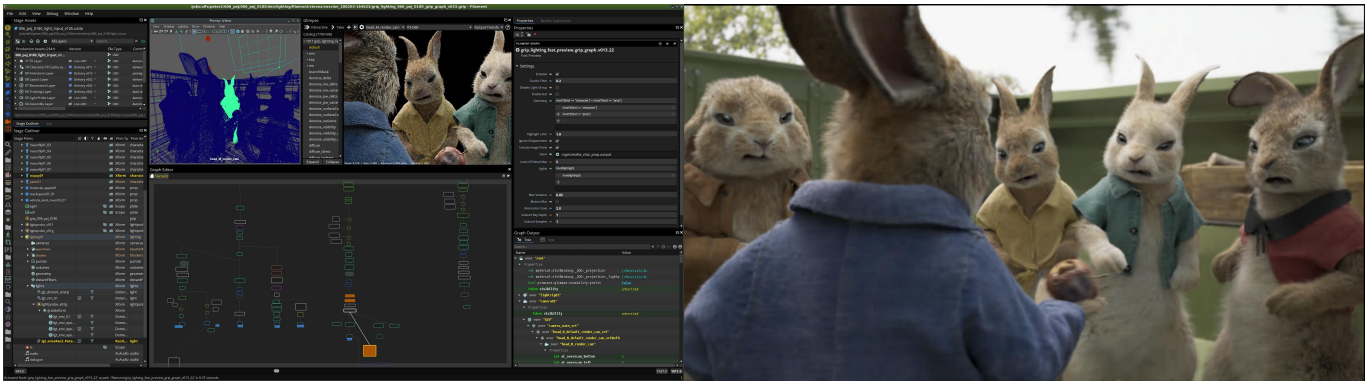

Figure 1: The Filament application at work on a *Peter Rabbit 2* shot. PETER RABBIT and all associated characters ™ & © Frederick Warne & Co Limited. PETER RABBIT™ 2, the Movie © 2020 Columbia Pictures Industries, Inc. All Rights Reserved.

ABSTRACT

Animal Logic recently overhauled its outmoded lighting workflow for the film *Peter Rabbit 2*. Since Pixar's Universal Scene Description (USD) was being adopted as the primary scene description format throughout the studio pipeline, this technology became a natural backbone around which to implement the new lighting toolkit. Following previous work to integrate USD into our animation pipeline[Baillet et al. 2018] we introduce *Grip*, a USD-native library which provides a node-based approach to authoring procedural modification of scenes; and *Filament*, a Qt-based application serving as the artist front end for interacting with a USD scene, the Grip engine, the production renderer, and pipeline tools.

CCS CONCEPTS

• **Computing methodologies** → **Computer graphics**.

KEYWORDS

USD, VFX pipeline, lighting

ACM Reference Format:

Steve Agland, Simon Bunker, Jakub Jeziorski, Manuel Macha, Eoin Murphy, and Francesco Sansoni. 2020. Grip and Filament: A USD-Based Lighting Workflow. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Talks (SIGGRAPH '20 Talks), August 17, 2020.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3388767.3407350

SIGGRAPH '20 Talks, August 17, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7971-7/20/08.
https://doi.org/10.1145/3388767.3407350

1 HISTORY

For a decade prior to *Peter Rabbit 2*, Animal Logic's Lighting team used a text-based render management tool known as *RSS* (Render Submission Script). Working scenes were assembled in Maya, but contained a mixture of native Maya data, Alembic files, Glimpse archives and other in-house formats, all "bridged" into Maya in different ways.

The RSS system had a number of disadvantages. When working interactively, it made destructive/non-undoable changes to the scene. Light rigs were built in Maya as a separate asset to the associated RSS script. There was only limited support for reusability, and limitations on which scene changes could be made.

We experimented with in-house solutions to some of these problems in the form of *CSD - Common Scene Description* (an extension of Glimpse's GSS Format[Fascione et al. 2019, p. 110]) and *Glance* (a Glimpse-centric pre-render scene modification system, with some resemblance to Renderman RiFilters).

Meanwhile, Pixar shared an early cut of USD with us, and we started to use it in Animation in 2016. Soon after, Animal Logic embarked on an ambitious project to build an end-to-end USD-based pipeline, and by 2018 Lighting was ready to adopt the technology as part of the effort to replace RSS with a next-generation toolkit. The goals were to reduce the number of data representations to a minimum, move to a more interactive lighting workflow, unify several separate pipeline tools, and retain the best aspects of the

RSS system. These included automatability, consistency, and rapid software iteration based on feedback from senior lighting artists. We carefully considered off-the-shelf options. At the time, SideFX's *Solaris* project was only nascent and the Foundry's *Katana* would require data conversions. We also wanted to retain the ability to customize the design.

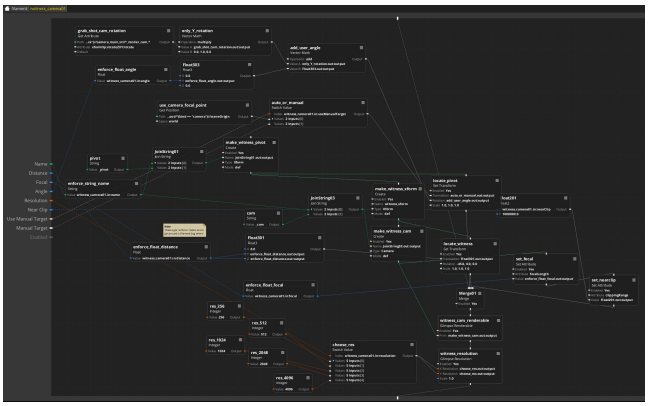

Figure 2: Grip node graphs represent prim operations and flow control (vertical), as well as queries and value processing (horizontal).

2 THE SOFTWARE STACK

2.1 Grip

Developed by Jakub Jeziorski, *Grip* is a USD-native library which provides a node-based approach to procedural modification of scenes. It is designed to be lean (USD is Grip's only dependency) and support diverse use cases. This includes interactive use by Lighting artists as well as efficient offline scene processing, such as preparation of review renders for departments throughout the pipeline.

Grip offers a number of node types: prim operations (creating prims, setting attributes, switching variants, etc); flow control (merge, switch, iterate, etc); path queries (finding prim paths with various predicates); and value processing (math operations on time-sampled numeric values, string and array manipulation, etc).

Grip lazily evaluates a graph of operations and writes the changes to a private layer. The layer content is transferred to an arbitrary stage layer in a single operation. "Read" operations use an immutable stage, while "write" operations go to a mutable off-stage target layer. Operations apply to a subset of prims using composable query nodes, or using *GEL* ("Grip Expression Language") statements, which allow for sophisticated filtering during scene traversal. Grip uses USD for serialising its node graphs, leveraging USD schemas to generate python bindings.

Complex Grip graphs can be encapsulated by TDs or artists as subgraphs with custom interfaces (Fig. 2), and can be published and referenced elsewhere. Most nodes that artists work with are these higher-level nodes. "Session values" can be used to provide hints to the execution context (such as interactive vs offline, the current shot or render quality level), allowing for flexible conditional behaviour.

2.2 AL_USDMaya

AL_USDMaya[Bateman et al. 2019] was used to author static light rigs on *Peter Rabbit 2*, with changes translated interactively to the USD stage, and Maya providing a familiar viewport.

2.3 Filament

Filament (Fig. 1) is a Qt-based application which sits at the top of our software stack, allowing lighting artists to interact with USD scenes, design Grip graphs and perform routine pipeline actions. *Filament* can be run standalone or inside Maya. It is written in Python, easily extensible by TDs, while performant operations are handled by optimised low-level libraries. It glues together several systems: USD, Grip, Glimpse, AL_USDMaya, render submissions, production browsing, asset resolution overrides, etc. It is built on Nucleus, our internal modular UI framework.

3 CONCLUSION

Reinventing our lighting toolkit based on a single data representation has simplified development in many respects. Less developer time is spent handling corner cases and more time is spent improving the user experience. Complications arose from our attempt to support editing of static light rigs in AL_USDMaya while running Grip. We hope to avoid this in future by authoring all lights dynamically through Grip.

4 FUTURE WORK

We are planning to convert our Glimpse-native material format to *USDShade* and integrate material editing into Filament, reusing work from our Grip UI. Scalability is a concern for our next major production and we're exploring more intuitive support for delayed payloads and proxy geometry. We are also looking at extending Grip evalution generate interim "snapshots" partway through the graph. Filament is embedded in Maya to make use of its interactive viewport, but this adds complex dependencies. Since Lighting's viewport requirements are relatively modest, we are investigating building an independent viewport for USD scene display with support for selection and manipulation of lights.

ACKNOWLEDGMENTS

Thanks to Craig Welsh, Daniel Heckenberg, Eddie Hoyle, JP Collins, Jonathan Penner, Rodrigo Janz, Callum Howard, Prethish Bhasuran and the Lighting crew on *Peter Rabbit 2*. Thanks also to Sebastian Grassia, George ElKoura and the USD team at Pixar for their support.

REFERENCES

Aloys Baillet, Eoin Murphy, Oliver Dunn, and Miguel Gao. 2018. Forging a New Animation Pipeline with USD. In *ACM SIGGRAPH 2018 Talks (SIGGRAPH '18)*. Association for Computing Machinery, New York, NY, USA, Article Article 54, 2 pages. https://doi.org/10.1145/3214745.3214779

Rob Bateman, Eoin Murphy, Fabrice Macagno, Paul Molodowitch, and Aloys Baillet. 2019. AL_USDMaya. https://github.com/AnimalLogic/AL_USDMaya.

Luca Fascione, Johannes Hanika, Daniel Heckenberg, Christopher Kulla, Marc Droske, and Jorge Schwarzhaupt. 2019. Path Tracing in Production: Part 1: Modern Path Tracing. In *ACM SIGGRAPH 2019 Courses (SIGGRAPH '19)*. Association for Computing Machinery, New York, NY, USA, Article Article 19, 113 pages. https://doi.org/10.1145/3305366.3328079