

Walter: An Open Source VFX Framework for USD and Alembic

Guillaume Laforge
Head of R&D
Rodeo FX
glaforge@rodeofx.com

Victor Yudin
Principal Core Developer
Rodeo FX
victor@rodeofx.com

Lucille Caillaud
R&D developer
Rodeo FX
lucille@rodeofx.com

Julien Couet
Software developer
Rodeo FX
jcouet@rodeofx.com

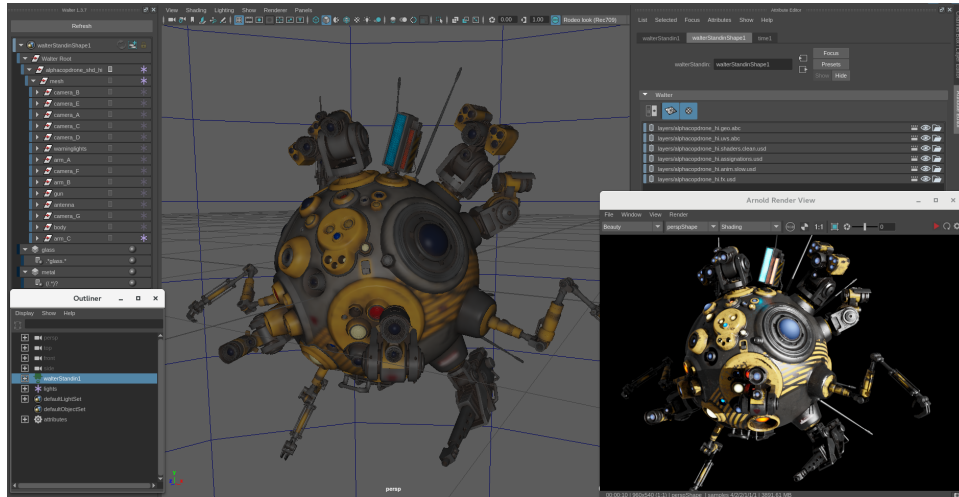


Figure 1: Walter for Maya, one single node to access a scene made of several layers.

CCS CONCEPTS

• **Information systems** → *Join algorithms*; • **Software and its engineering** → *Application specific development environments*;

KEYWORDS

alembic, dcc, framework, pipeline, usd, vfx

ACM Reference Format:

Guillaume Laforge, Victor Yudin, Lucille Caillaud, and Julien Couet. 2018. Walter: An Open Source VFX Framework for USD and Alembic. In *Proceedings of SIGGRAPH '18 Talks*. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3214745.3214772>

1 INTRODUCTION

In a VFX Studio, sharing 3D contents between departments can be very challenging. Departments are producing some artistic work using data coming from previous one(s). Since different software is used, such pipeline is cluttered with many files formats, leading to a lot of data duplication and incompatibility. Finally, as scene complexity is increasing every year, interacting with such data can be extremely difficult for artists.

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.

<https://doi.org/10.1145/3214745.3214772>

To solve those problems, input and output converters are usually used. The main issue being that even simple changes require the whole import/export process. The other approach is to use the renderer file format as the main format moving along the pipeline. Scenes can be loaded quickly, but editing and OpenGL display can be quite limited.

This extended abstract is showing Walter, a new system proposing a more efficient solution to solve those problems in modern VFX pipelines.

2 CHARACTERISTICS OF WALTER

2.1 Main idea

Walter's main idea is to decouple as much as possible the production data from any Digital Content Creation (DCC) formats and to control how to load, edit and display large scenes efficiently. Our departments would only publish a delta of the changes (compared to the incoming data), and each DCCs could interact with such scene at various granularity levels using dedicated tools. Portability across Arnold, Houdini, Katana, and Maya for layout, animation, lookdev, and lighting is the main target of the first version of Walter.

2.2 Leveraging USD and Alembic feature set

In its purest form, a VFX pipeline can be thought of as a simple stack of layers that all together will compose the final shots. Since open source initiatives like Pixar USD and Alembic are already providing excellent solutions to describe our 3D scenes using a layer based approaches, we decided to concentrate our effort on the integration of such technologies into Houdini, Maya, and Katana.

3 CONTROLLING THE SCENE

Loading time, particularly on large production scene, is very dependent on the architecture of the DCC. To solve this problem, Walter encapsulates its scene in a single node inside the DCC. Furthermore, Walter manages the rendering pipeline to display this scene to keep a high level of interactivity and consistency across DCCs.

In short, we are not too dependent on the dependency graph and display/rendering performance of the host.

3.1 Layers

Using Walter, artists can load just part of the sequence or shot they need to work on in Arnold, Maya, Houdini, and Katana. A Walter node load all the data from USD or Alembic sparse layers and display it.

Different departments in a VFX studio produce a different kind of layers.

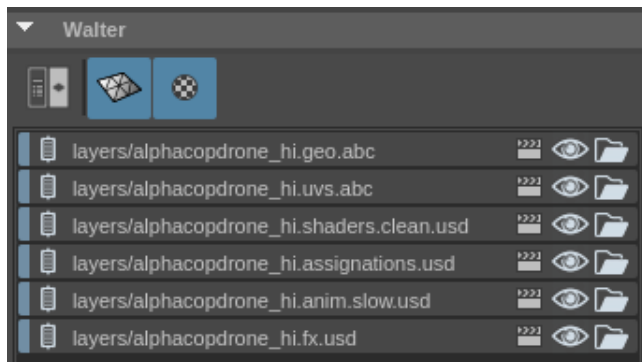


Figure 2: A stack of Walter layers. Their composition is creating a 3D asset, shot or sequence.

3.2 Edits

Once the scene is loaded, artists can create edits on top of the layers coming from previous departments like adding shaders, lights or changing various attributes like transformations or visibility flags on specific objects. For example, if an artist needs to change the level of subdivision, or the translation of an object, only those values exists in the DCC and Walter responsibility is to transfer such data to its scene.

3.3 Sharing Edits

Finally, those edits are saved as new Alembic or USD layers. As the edit is generally small, containing only the changes, publish time is always very fast, even in big scenes.

4 DISPLAYING AND RENDERING THE SCENE

Two renderers are available to display the scene through the Pixar Hydra Renderer plugin architecture.

4.1 OpenGL display

The first solution is using Hydra GL renderer for real-time display. We slightly modified Hydra GL renderer to support UDIMS textures.

This way, Walter can re-use the same textures than the ones used for final renders, simplifying our texture pipeline.

4.2 Hybride OpenGL and Raytracing display

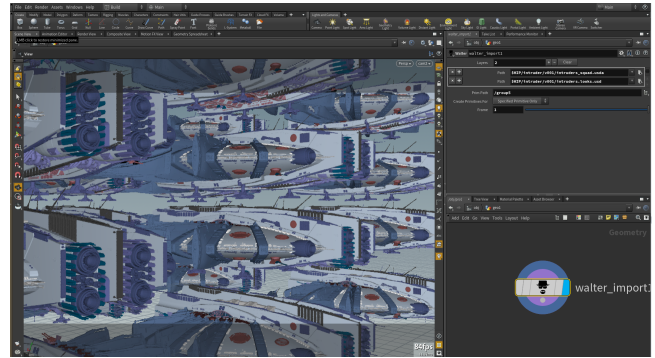


Figure 3: Walter for Houdini. A high polycount scene displayed using the Rodeo Path Tracer.

A VFX shot can describe very high detailed geometries only renderable in so-called offline renderers like Arnold or Prman. For simple display, workarounds like a different level of details or textured cards have been used since a long time. Such techniques were required because traditional OpenGL renderers use rasterization techniques to solve the visibility problem. Unfortunately, it does not scale very well. On the other hand, modern raytracers are using acceleration structures that can manage scenes with an incredible amount of details. Based on those observations, we started to write a path tracer dedicated to fast display in the viewport. This way, the high-resolution version used for the final render can be loaded for any interactive work reducing the amount of work for artists.

4.3 Arnold procedural plugin

For final renders from Houdini or Maya, our Walter Procedural node will generate the scene directly inside Arnold. Also, shading layers creating in Maya can be loaded by Houdini artists seamlessly. For the assignation of shaders on objects, Walter is using own expression engine, letting us assign shaders in a more procedural way using regular expressions evaluated at render time.

4.4 Katana Geolib Op

Lighters working in Katana can create a WalterIn node that can load all the layers coming from other departments including materials and assignations.

5 CONCLUSIONS AND FUTURE WORK

Walter is actively under development since its first use on Game of Throne season 7 for the Wall's fall. Artists feedback is very encouraging as they can now work on several shots much faster than before. Since our interactive path tracer is giving some impressive results, this is an area we would like to explore more in the future, mainly trying to improve the look without losing the framerate. Finally, supporting MaterialX and other offline renderers could be investigated too.