Matte Painting a Brighter Future: A USD-Based Toolset in Nuke

Michael De Caria
michaeldec@al.com.au
Animal Logic
Sydney, NSW, Australia

Prethish Bhasuran
prethishb@al.com.au
Animal Logic
Sydney, NSW, Australia

Mitja Müller-Jend
mitjam@animallogic.ca
Animal Logic
Vancouver, BC, Canada

Manuel Macha
manuelm@al.com.au
Animal Logic
Sydney, NSW, Australia

Figure 1: The Nuke application showcases Animal Logic's various in-house USD-based panels used by matte painting artists. Presented is the USD ALab Open Source Scene.

ABSTRACT

We introduce Animal Logic's advanced 3D matte painting toolset `AL_USDNuke`, which seamlessly integrates Nuke into our USD-centric pipeline. We detail the integration of numerous components such as our path-traced GlimpseViewport for instant representation of large USD stages, a user-friendly node-based toolset to modify USD stages, visualisation by our in-house renderer Glimpse for high-fidelity ground-truth feedback, and complementary views to efficiently manage USD stages inside Nuke. This was achieved by a specialised Nuke to USD translator and our brand-new framework `Plasma`, an enhancement of Animal Logic's in-house Nucleus framework for large-scale application development, tailored to Nuke. These developments improve matte painting artists' efficiency on complex USD stages and allow them to publish their work into shots for the benefit of all other upstream as well as downstream departments.

CCS CONCEPTS

• **Computing methodologies** → **Animation**.

KEYWORDS

matte painting nuke USD animation pipeline

ACM Reference Format:
Michael De Caria, Prethish Bhasuran, Mitja Müller-Jend, and Manuel Macha. 2023. Matte Painting a Brighter Future: A USD-Based Toolset in Nuke. In *The Digital Production Symposium (DIGIPRO '23), August 05, 2023, Los Angeles, CA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3603521.3604294

DIGIPRO '23, August 05, 2023, Los Angeles, CA, USA

ACM ISBN 979-8-4007-0238-9/23/08…$15.00
https://doi.org/10.1145/3603521.3604294

1 INTRODUCTION

The Matte Painting department at Animal Logic traditionally sat at the back end of the pipeline, alongside lighting and compositing, creating predominantly 2D matte paintings in Photoshop projected onto 3D geometry inside Nuke [Foundry 2023]. However, these assets were not intended or able to be re-ingested into the pipeline by upstream departments. Animal Logic's company-wide adoption of USD (Universal Scene Description) [Pixar 2023] for our pipeline

foundation, which occurred during the production of *Peter Rabbit 2 (2021)*, presented an opportunity to completely reassess our approach to matte paintings.

We soon discovered that the consumption of a shot's entire USD stage would quickly overwhelm Nuke's native 3D system. We explored other viable alternatives such as DreamWorks' USD plugins [DreamWorks 2023] which had been open-sourced around the same time, as well as Pixar, who generously provided us access to their usd-nuke plugin. After the initial investigation, it was decided to implement proprietary USD read and write nodes in part due to requiring an integrated UI that enables users to load sub-selections of large USD stages into Nuke. This was the first step in our journey to provide USD workflows.

Our USD pipeline integration in Nuke, known as `AL_USDNuke`, started around 2020 and the first project to adopt it was *DC League of Super-Pets (2022)*. The design revolves around a custom USD translator for Nuke nodes and is enhanced by existing in-house UI components. This allowed our tools to leverage the power of USD's stage composition to publish DMP (digital matte painting) assets to all departments, with artists from any craft group able to review their shots in context with fully integrated matte paintings.

2 TOOLSET DEVELOPMENT

Feedback from artists would progressively lead to enhancements of the toolset on subsequent projects with the goal to improve intuitiveness and meet creative requirements. The toolset's workflow has evolved to encompass the following:

- Provide node-based capabilities to manipulate the USD stage such as adding geometry, setting up materials, camera projections and modifying prim attributes.
- Manage and context switch between multiple USD stages and their associated session layers.
- Serve as a toolkit for artists to push their DMP work from Nuke directly into one or multiple shots.
- Preview, load and render complex USD stages directly inside Nuke by integrating multiple in-house views.

2.1 Approach

Python is the primary programming language used to integrate the various components of the `AL_USDNuke` toolset. However, numerous of those components are leveraging lower-level languages like C++ to enhance performance. This approach provided TDs with an environment that allows fast iterations for implementing new features into `AL_USDNuke`.

Our initial development efforts were focused on translating Nuke node networks to USD layers.

To direct USD prim-overrides into their designated layers we utilised established components of Animal Logic's Entity & Fragment pipeline [Collins et al. 2022].

We also adopted our pre-existing in-house Qt UI framework, Nucleus, and seamlessly unified it with Nuke.

2.2 In-render compositing

Our former methodology of compositing the matte painting with an image from the renderer, led to inconsistencies in lighting and perspective, making it appear disconnected from the rest of the scene.

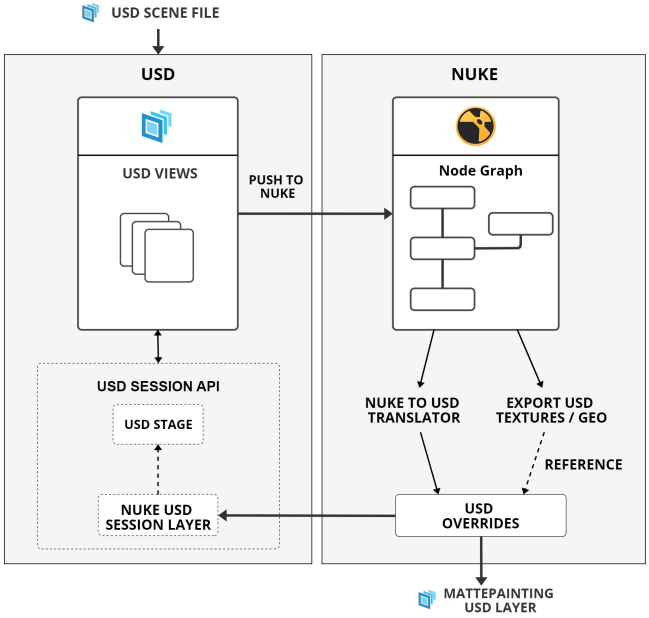

Figure 2: Diagram representation of the AL_USDNuke's implementation and scene management. The matte painting layer represents the generated output from Nuke, while a USD stage serves as the initial entry point.

With the introduction of `AL_USDNuke`, the work now produced by the Matte Painting department is rendered by Glimpse [Heckenberg et al. 2017] and viewed directly within Nuke. This workflow essentially enables in-render compositing of DMP work within the 3D scene, making for more seamless, believable renders and the matte painting environment interacts correctly with the lighting, shadows, and reflections more realistically. This DMP work will be delivered to the Lighting department where subsequent renders are produced using Filament [Agland et al. 2020].

2.3 Skydomes

As skydome matte paintings are frequently requested, the toolkit includes one-click solutions to create the required geometry and shader setup. The key aspect of this is provided by the Virtual-Breakdown API [Collins et al. 2022] which facilitates the breakdown recipe for skydomes. The creation of complex geometries was not discouraged; artists had the freedom to generate additional geometry using Nuke or to import it from other DCC applications like Maya.

3 NUKE TO USD TRANSLATOR

The translator component of `AL_USDNuke` fundamentally works by interpreting a Nuke node network and then generating the associated USD overrides for each node which is targeted to a session layer. Scene translation occurs as a 2-step process, USD Geometry creation and look development. The latter can be seen in Figure 3.

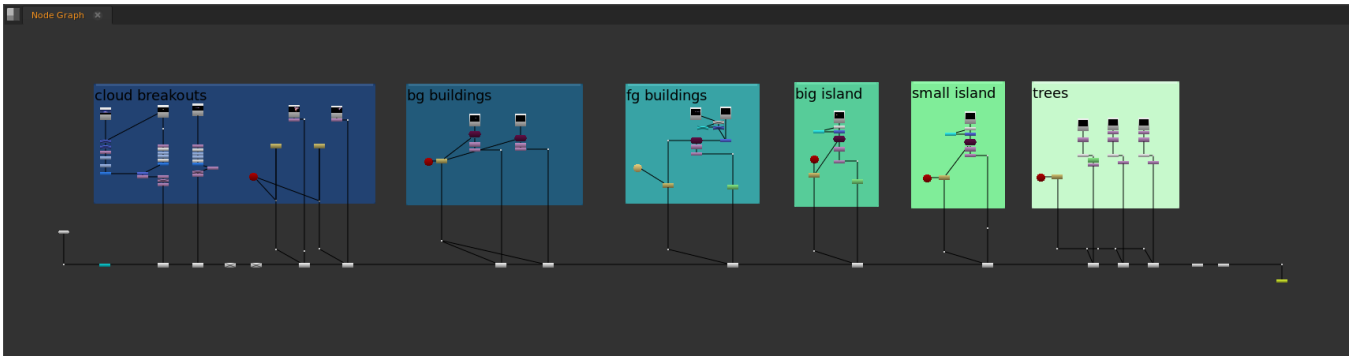

Figure 3: A Nuke node graph consisting of AL_USDNuke look nodes constructing a multi-layered matte painting.

3.1 Geometry and look development

Geometry conversion during translation was done using our NDK (Nuke Developer Kit)-based USD writer and was introduced by the node `AL_USDAddGeo`, capable to input any 3D geometry supported by Nuke. Look development of the USD geometry was the second process; all material-based look nodes have a predefined associated USDShade shader network and support Prim assignment, texture inputs and shader parameters. Where possible, native Nuke nodes are being reflected by each USD look node. To give an example, the Camera projection node `AL_USDCameraProjection` would use a Project3D and ApplyMaterial with exposed inputs to provide a camera and textures.

3.2 Layer generation

The layer generation of translated Nuke nodes takes place either through local rendering or by publishing the work to our asset management system. Rendering will use a Nuke node we call `AL_USDGlimpseRender` which internally is driven by an NDK node `GlimpseImage`, which will pass the image buffer rendered back to Nuke's 2D system. Publishing the USD layer created by the node network is managed by the `AL_USDCheckin` node. Both nodes will invoke the translator and traverse up the network, performing the translation of each Nuke node to USD equivalent, performing validation which then notifies the artist if any operation is incompatible.

The generated USD layers will follow our internal Entity and Fragment USD structure [Collins et al. 2022] and trigger a review to be rendered through our automated review system. In essence, the geometry will be part of the *geo* fragment and the look development will be published to the *look* fragment, both referenced under a top-level *domain* for the Matte Painting department which is the USD layer composed in shots.

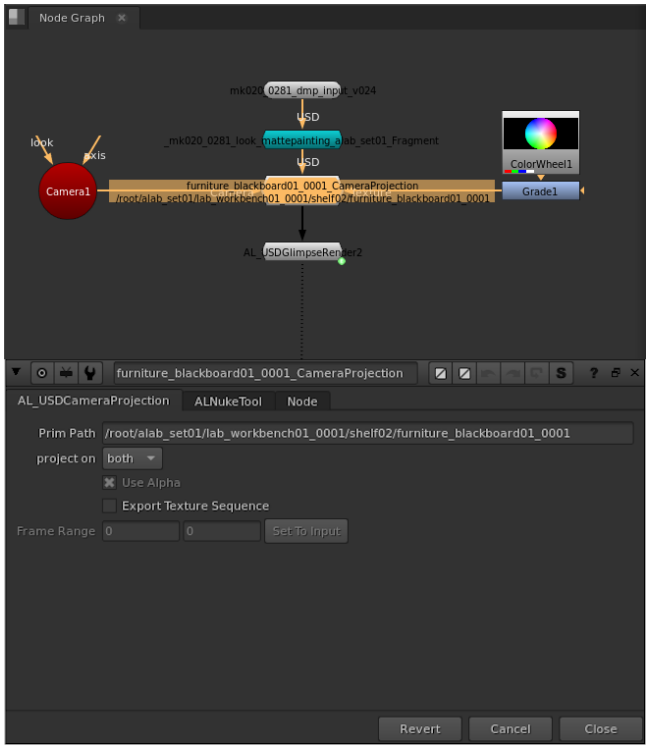

Figure 4: On the top is a simple AL_USDNuke look network with a texture and camera plugged into an AL_USDCameraProjection node. Displayed on the bottom are the knob properties of the selected AL_USDCameraProjection node.

3.3 Texture baking

Texture baking played a crucial role in enhancing render efficiency and preventing artists from having to spend time pre-processing their textures. Texture inputs to Look nodes can be any Nuke network that can be rendered to a 2D image, which is subsequently then automatically written out to disk. To know if any adjustments are made to the texture's Nuke node network, and a rebake is required: a node hash is implemented using *node.opHashes()* in combination with a node map cache. In cases where Reads are the only input and no colour space conversions are required, the baking can be skipped, and the renderer can be provided with a reference to the texture directly.

4 PLASMA

Plasma is a Nuke-specific enhancement built with Animal Logic's in-house large-scale application Nucleus framework. It allows us to leverage Animal Logic's existing library of UI components/Widgets developed for USD which significantly accelerates the development process as minimal UI components were required to be written from scratch. Within this framework, we automate Nuke menu creation, panel creation to show the views and Nuke's workspace tools to control how the views were positioned in the UI. Nuke's callback system is also integrated to autogenerate events, for example, timeline updates could be used to trigger an action in the USD views.

The main advantage of Plasma is that new views added can easily be made to interact with any existing ones. When we started looking into the UI part of the USD integration, we wanted to let the artists use UIs that they were already familiar with, such as an Outliner, Shelf, and predominantly a customisable 3D viewport.

4.1 Pushing to the viewport

During testing, we found that Nuke's native 3D system did not scale well when it came to large scenes once converted. On the other hand, our in-house USD GlimpseViewport could display and handle large USD scenes without any performance degradation. To maintain native Nuke 3D nodes support, Figure 5 is an example where artists could visually select parts of the scene and *push* parts of the 3D geometry into Nuke.

4.2 Supplemental USD views

Other USD-related panels made available in Nuke were a USD Properties view for viewing or modifying USD attributes and a USD session view for dealing with the opening or closing of USD stages. These adjustments were only for the current Nuke session and don't get saved with the Nuke scene. If the artist wanted modified attributes to be baked out into USD for future sessions, they needed to be made using `AL_USDModifier` node and added to the node graph.

5 FUTURE WORK

The recent release of Nuke-14 and its fully redesigned USD-based 3D system will open enticing, new possibilities. Future work will be focused on enhancing `AL_USDNuke` with those new features and enable workflows which are currently hard to implement due to the separation of USD and Nuke's native 3D system. We will also explore direct texture rendering without intermediate files on disk to further improve turnaround times on production shots. Furthermore, we are preparing to make `AL_USDNuke` accessible to Lighting and Compositing artists so that they can also benefit from its feature set.

6 CONCLUSIONS

The deployment of `AL_USDNuke` had a big impact on the productivity of matte painting artists at Animal Logic by streamlining previously fragmented and unfeasible workflows due to the limited native USD features inside Nuke. Animal Logic's USD pipeline design played a critical part in enabling these modern workflows. Although we incurred the development cost of early adoption of USD within Nuke, it still resulted in immediate value and provided us with a clear direction for the future. Artists are now able to be more efficient and have the flexibility to deliver their work directly into shots via USD stage composition, providing a distinct advantage over our previous approach to integrating digital matte painting.

ACKNOWLEDGMENTS

The authors would like to thank Aaron Barclay, Scott Russell, John Rix for their significant contributions and their integral role in realising the original vision. We would also like to thank Yara Du for her recent contributions to `Plasma` and Zhicheng Ye for his foundational work on the Nuke USD Reader/Writer.

REFERENCES

Steve Agland, Jakub Jeziorski, Manuel Macha, Simon Bunker, Francesco Sansoni, and Eoin Murphy. 2020. Grip and Filament: A USD-Based Lighting Workflow. In *ACM SIGGRAPH 2020 Talks* (Virtual Event, USA) *(SIGGRAPH '20)*. Association for Computing Machinery, New York, NY, USA, Article 33, 2 pages. https://doi.org/10.1145/3388767.3407350

Jon-Patrick Collins, Romain Maurer, Fabrice Macagno, and Christian Lopez Barron. 2022. USD at Scale. In *The Digital Production Symposium* (Vancouver, BC, Canada) *(DigiPro '22)*. Association for Computing Machinery, New York, NY, USA, Article 11, 6 pages. https://doi.org/10.1145/3543664.3543677

DreamWorks 2023. *DreamWorks Animation USD Plugins GitHub Page*. Retrieved May 12, 2023 from https://github.com/dreamworksanimation/dwa_usd_plugins

Foundry 2023. *Nuke Product Page*. Retrieved May 12, 2023 from https://www.foundry.com/products/nuke-family

Daniel Heckenberg, Luke Emrose, Matthew Reid, Michael Balzer, Antoine Roille, and Max Liani. 2017. Rendering the Darkness: Glimpse on the LEGO Batman Movie. In *ACM SIGGRAPH 2017 Talks* (Los Angeles, California) *(SIGGRAPH '17)*. Association for Computing Machinery, New York, NY, USA, Article 8, 2 pages. https://doi.org/10.1145/3084363.3085090

Pixar 2023. *Pixar's Graphics Technology Page*. Retrieved May 12, 2023 from https://graphics.pixar.com/

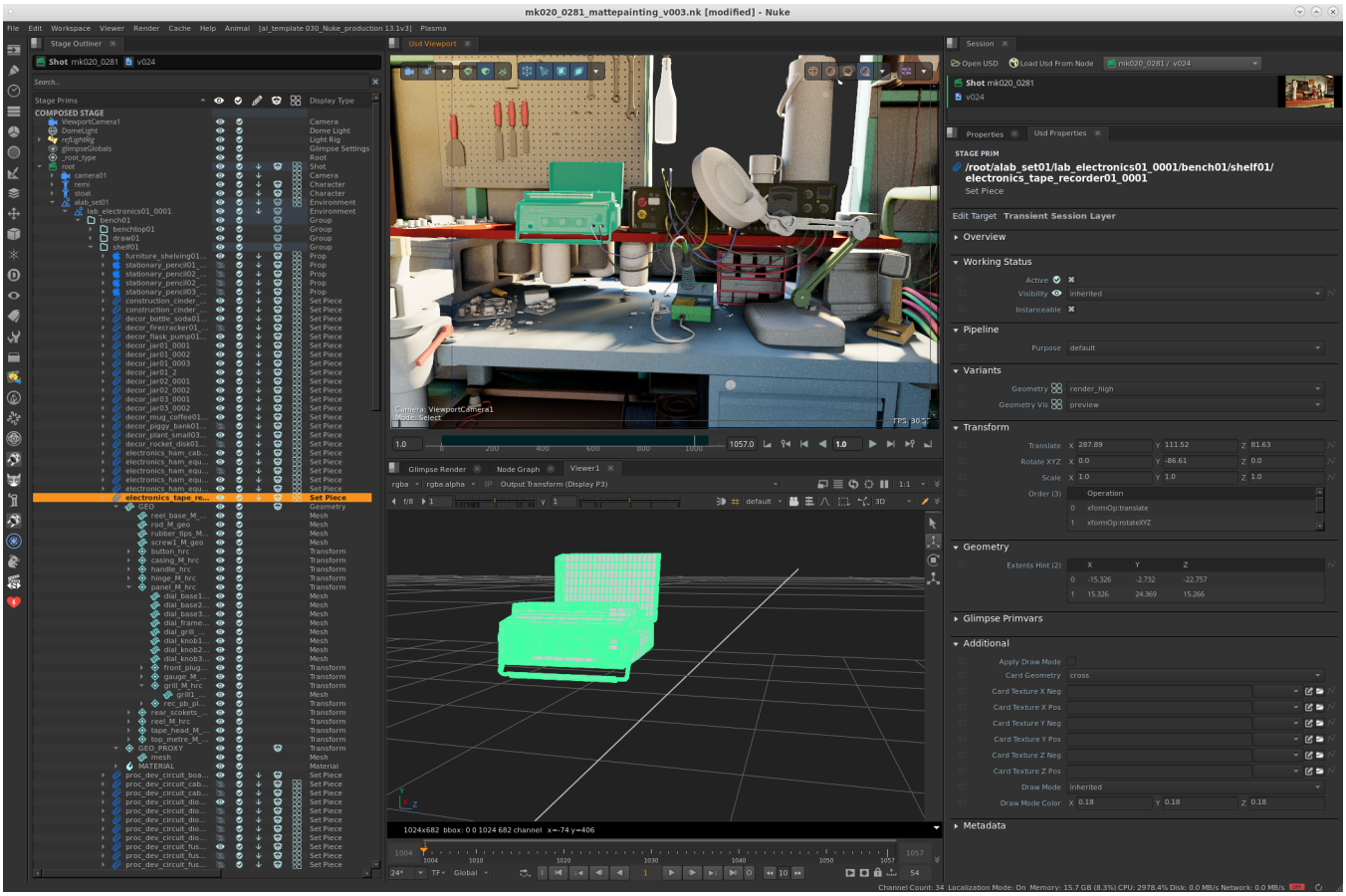

Figure 5: A singular object pushed and converted into Nuke's native 3D system with USD views including Stage Outliner, USD viewport and USD Properties.