

# GafFour and Sequence-based Lighting

Xinling Chen  
Sony Pictures Imageworks  
cchen@imageworks.com

Lucas Miller  
Sony Pictures Imageworks  
lmiller@imageworks.com



Figure 1: Four key shots in the sequence named fun in *Hotel Transylvania 3 Summer Vacation*. ©2018 Sony Pictures Animation Inc. All rights reserved.

## ABSTRACT

Sequence-based lighting has become increasingly popular to further improve efficiency at Imageworks as we are producing thousands of full CG shots each year. However, Katana slows down dramatically as the lighting setup becomes more complicated to accommodate the growing number of nodes which may make up the different shots. We analyzed a large number of sequence-based lighting scene files that were identified as slow and found that these setups usually had tens of gaffer nodes, which each contained thousands of nodes for light creation and manipulation. To solve this problem we implemented a custom gaffer node for Katana, GafFour, which greatly reduced the total number of nodes per scene file.

## CCS CONCEPTS

• **Computing methodologies** → *Computer graphics*;

## KEYWORDS

lighting, key-lighting, workflow, efficiency

### ACM Reference format:

Xinling Chen and Lucas Miller. 2018. GafFour and Sequence-based Lighting. In *Proceedings of SIGGRAPH '18 Talks, Vancouver, BC, Canada, August 12-16, 2018*, 2 pages.  
<https://doi.org/10.1145/3214745.3214773>

## 1 INTRODUCTION

Katana is an asset-based approach to look development and lighting for 3D computer generated scenes, providing scalability to meet the needs of today's most demanding CG-rendering projects. At Imageworks, Katana has been the essential tool used by the look development and lighting teams for animated features and VFX

projects. [LaVietes et al. 2008] Its flexible node-graph and rule-based workflows lets artists naturally build up templates as they work, which can then be easily shared with other team members or reuse on other shots; Its gaffer node, GafferThree, allows artists to manage light creation, light linking, constraints and light filters from a one-stop control panel; create large sets of lights and control their intensity, color and other parameters from a single item; and build up hierarchical, inheritance-based sequence lighting. [The Foundry 2018]

In the past few years, as the production time became shorter for each project, lighting leads started to look to sequence-based lighting workflows to reduce the amount of time to light an individual shot. Key lighting often takes the largest chunk of time in finalizing a sequence. This is because the look at this stage allows a director or supervisor to set how a sequence is going to look with fewer artists required than shot lighting. Once the key-lighting is approved, it usually only takes a couple days for a few shot lighters to finish the individual shots that make up the sequence. For key lighters, the benefits of having one single scene that handles a sequence of shots is obvious. There is no need to open a different scene when you need to work on the next shot. One change in the key gaffer can affect all shots set up in that scene. Also it is much faster to check the influence of the change on other shots and dial in subsequent changes (Fig. 2). The major reason that some key lighters shy away from this workflow is because the Katana scene can become unbearably slow. Not only is the live scene affected, but loading, auto-saving, and live-rendering takes much longer. Simply moving the light accurately via manipulator can become a futile and frustrating experience.

We analyzed the production scenes that were reported as being slow and found that about half of the nodes in the scene were created due to light creation and edits in the GafferThree nodes which is shipped with Katana. As we had three animated feature films, *Hotel Transylvania 3: Summer Vacation*, *Smallfoot*, and *Spider-Man, Into the Spider-Verse*, starting lighting at about the same time, we decided this was an area worth improving.

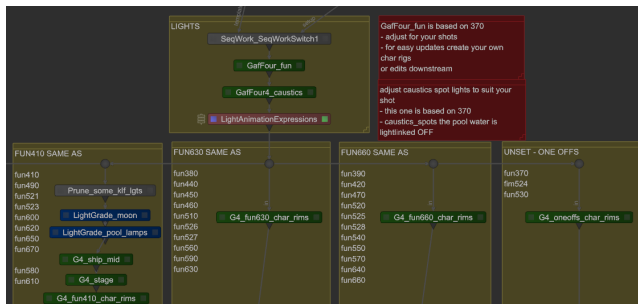
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party 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/10.1145/3214745.3214773>



**Figure 2: A screenshot of the sequence-based lighting setup for the sequence shown in Fig.1. The key gaffer, GaffFour\_fun, sits on top of 4 branches, affecting all shots set up the scene. Each branch is a key shot using the downstream-edit feature to make local overrides. ©2018 Sony Pictures Animation Inc. All rights reserved.**

## 2 SOLUTION AND IMPLEMENTATION

The primary job of Katana as a lighting tool is to create and edit lights. Unfortunately, the one-stop control panel, GafferThree, does not scale well when you get beyond around a hundred lights. It is just a wrapper that gathers different nodes and relies on each node to perform their duty, so a single light can be more than a handful of nodes. As the number of lights and rigs increases, the number of nodes explodes.

The solution was to write a gaffer that uses a limited number of nodes to handle light creation and editing on multiple lights. Every member node in the gaffer needs to be able to handle multiple locations. There are two types of member nodes in our gaffer, one is for creation, the other is for editing. For nodes that create locations such as lights and rigs, we use arrays to store parameters required for that type of location. Each parameter array grows as more locations are added. For nodes that edit locations, we store the data in sparse form – only when an edit was made to a specific location, that information would be saved on the node. This method effectively minimized the number of nodes and number of parameters to be used in the GaffFour node.

As for how to present the data and parameter controls in a familiar way to the user, a proxy layer was introduced which let users view the parameters and dial them at a single location. This layer could retrieve the data for the selected location and push back the updated value of a parameter to the correct slot in the array for that parameter or to the node responsible for editing that parameter as a sparse edit. With this design, we were also able to address other requests from our key lighters. For example, duplicating incoming lights or rigs in downstream gaffers, renaming adopted lights and rigs that are not present in the incoming scene, hierarchical master material support, and others.

## 3 RESULTS

Unsurprisingly, GaffFour is much faster than GafferThree that ships with Katana. In our stress test, it can handle 40k lights without any problem. The node graph will not grow as more lights are added, because GaffFour uses a limited number of nodes to handle

light creation and edits. This makes a huge difference for scenes set up for sequence lighting. For example, we found that after we converted all GafferThree nodes in an early sequence lighting scene from *Hotel Transylvania 3* to GaffFour nodes, more than 2000 nodes were gone, and Katana became noticeably more responsive.

The challenges we faced to speed up sequence based lighting scenes motivated us to develop our own solution for Katana to efficiently create and edit large sets of lights. Our custom gaffer node, GaffFour, has become the default at our facility, allowing key lighters to take advantage of sequence-based lighting workflows.

## REFERENCES

- Steve LaVietes, Brian Hall, and Jeremy Selan. 2008. Katana Lighting Pipeline. In *ACM SIGGRAPH 2008 Talks (SIGGRAPH '08)*. ACM, New York, NY, USA.
- The Foundry. 2018. KATANA. <https://www.foundry.com/products/katana>. (2018).