Lighting Up The Smurfs Enchanted Forest

Xinling Chen
Sony Pictures Imageworks

Christopher Kulla
Sony Pictures Imageworks

Lucas Miller
Sony Pictures Imageworks

Alan Chen
Sony Pictures Imageworks

Figure 1: The Enchanted Forest in Smurfs: The Lost Village. ©2017 Columbia Pictures Industries, Inc

ABSTRACT

In *Smurfs: The Lost Village*, the enchanted forest is central to the environment and story. The task was to create millions of exotic plants of hundreds of different species spreading across the forest. The plants in the forest emit light when in the shade, and stop when they are hit by sunlight. We revisited the way in which our lighting tool creates lights to scale to huge light counts without compromising on the ability to fine tune lighting through features like light linking. This lead to simplified workflows for the artists who were able to easily manipulate shots with up to several millions of individual lights.

CCS CONCEPTS

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

KEYWORDS

lighting, lookdev, instancing, workflow

ACM Reference format:
Xinling Chen, Christopher Kulla, Lucas Miller, and Alan Chen. 2017. Lighting Up The Smurfs Enchanted Forest. In *Proceedings of SIGGRAPH '17 Talks, Los Angeles, CA, USA, July 30 - August 03, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3084363.3085049

SIGGRAPH '17 Talks, Los Angeles, CA, USA
 978-1-4503-5008-2/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3084363.3085049

1 BACKGROUND AND CHALLENGES

1.1 Lighting the Enchanted Forest

The enchanted forest is set in multiple sequences throughout the movie; it is where the adventure begins for the smurfs. It is populated with millions of bioluminescent plants covering the forest floor from the distant background, all the way to the foreground, interacting with the Smurfs. As mentioned, the plants emit light only when not being hit by sunlight. To achieve the ideal look, mesh lights are preferred by the lookdev artists for each type of plant, while instancing is preferred for the populating the environment with the plants. This is challenging because the Katana global light list requires each light to explicitly exist in the scene, while the mesh light setup requires the geometry to exist in the scene, both of which inherently contradict with instancing.

1.2 Lighting and Lookdev in Katana

Katana is an asset-based approach to look development and lighting for 3D computer generated scenes, providing scalability to the pipeline[The Foundry 2011]. At SPI, Katana is the essential tool used by the look development and lighting teams for both animated films and VFX projects.

Traditionally, lookdev artists bake out look files, which store textures and shading assignments for an asset separately from the geometry. Lighting artists automatically pick up associated look files for each asset and manage light creation, light-linking, and other lighting-related tasks in per-shot Katana scenes. If light rigs were added by lookdev and baked into a lookfile, these lights would automatically show up in the scene graph hierarchy, but only illuminate when they were activated. This activation process involves adding the lights to the global light list, which stores all the

Figure 2: Mesh lights create accurate interactions between characters and complex shaped light sources. ©2017 Columbia Pictures Industries, Inc

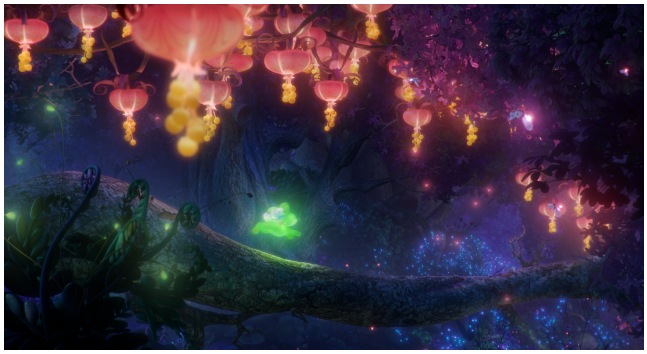

Figure 3: The enchanted walk set, characterized by the hanging Chinese lanterns. ©2017 Columbia Pictures Industries, Inc

lights and light linking information in the scene. The renderer can then refer to this global list to determine the lights to be created.

Lookfile lights proved to be especially powerful for lighting setups in large environments made up of light emitting objects. A typical large shot, as shown in Fig.1, would contain anywhere from 3 to 6 million instances of plants. Although the majority of those instances are simple plants, there may be up to 40,000 lookfile lights with complex plants in the scene. Given the large list of items, Katana would often run out of memory before adding all lookfile lights to the global light list. Activating these and managing the resulting huge light list became increasingly cumbersome for lighting.

2 SOLUTION AND IMPLEMENTATION

In Katana, the traversal of the scene graph is the means by which geometry is loaded (either from disk or procedurally), material information is resolved and per-shot tweaks are combined[Hall et al. 2014]. For large scenes this can take a while, therefore it is only practical to traverse the scene graph once. However, since light linking requires establishing relationships between geometry and lights, we have traditionally relied on all lights being created ahead of all geometry by the means of a global light list. Unfortunately, when the lights are siblings of geometry, accessing the lights upfront can lead to cooking parts of the scene redundantly. Moreover, instancing of lights was not supported because it required a data-amplification stage to build extra lights. When combined with the desire to use light linking, this approach quickly became impractical.

The solution was to discard the concept of a global light list and change the way our renderer plugin exports the scene to the renderer at scene build time. This did not present any major technical difficulty as our renderer does not rely on delayed loading of geometry, nor does it impose any particular ordering to how lights must be created. Only light linking required special handling. We redesigned the attribute convention by which light links are expressed in Katana. We recorded exceptions to the default state of the light in a compact table based on the observation that the number shapes that want to differ from the default state of the light (on or off) is small. When the entire scene graph has been traversed, we can quickly loop over all created shapes and simply generate the final list of lights that should affect each object in the scene. Because the recorded information is in terms of scene graph paths as opposed to renderer objects, features like instancing work efficiently as well as many renderer objects come from a small set of paths.

With this feature in place, lookdev artists created two variations for each type of plant: "night" and "day". We built a tool to mark plants that are in the cone of the sunlight as "day" and all of the others as "night". With the lookfile of the correct "day/night" variation being loaded automatically, and the on-demand-light feature enabled, shot lighters were relieved from creating or managing millions of environment lights and were able to focus more on the actual work of character lighting, with the environments working out of the box. Also, scene optimization became easier, as we prune the environment by camera frustum and clipping planes, instanced lights will be gone naturally at the same time. Some beautiful shots set in the enchanted forest ended up having a few hundred lights after optimization

3 CONCLUSION

The challenges that we faced in creating the enchanted forest motivated us to revisit the fundamentals of how our lighting tool thinks about lights. By generating lights on demand, we have raised the bar on the number of lights that can be efficiently supported and freed the artists from tedious technical steps that existed purely as historical limitations. We could take advantage of this to support new workflows and bring this stunning environment to life. This new workflow has become the default at our facility, allowing all future shows to make much more liberal use of per-asset lights when setting up large environments.

REFERENCES

Brian Hall, Jeremy Selan, and Steve LaVietes. 2014. Katana's Geolib: Behind the Scenes. In *ACM SIGGRAPH 2014 Talks (SIGGRAPH '14)*. ACM, New York, NY, USA, Article 60, 1 pages. https://doi.org/10.1145/2614106.2614143

The Foundry. 2011. KATANA. https://www.thefoundry.co.uk/products/katana/. (2011).