Light Pruning on Toy Story 4

Vaibhav Vavilala
vibe@pixar.com
Pixar Animation Studios

Culling lights in this shot dropped the light count from 8400 to 260, and render time from 98 hours to 48. ©Disney/Pixar.

ABSTRACT

Pixar films have recently seen drastically rising light counts via procedural generation, resulting in longer render times and slower interactive workflows. Here we present a fully automated, scalable, and error-free light pruning pipeline deployed on Toy Story 4 that reduces final render times by 15-50% in challenging cases, accelerates interactive lighting, and automates a previously manual and error-prone task.

CCS CONCEPTS

• **Computing methodologies → Rendering**.

KEYWORDS

Rendering optimization, RenderMan, Universal Scene Description, Katana, Light Path Expressions

ACM Reference Format:
Vaibhav Vavilala. 2019. Light Pruning on Toy Story 4. In *Proceedings of SIGGRAPH '19 Talks.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3306307.3328161

1 BACKGROUND & RELATED WORK

Pixar previously employed an automated light culling pipeline for the REYES renderer on Brave [Tam et al. 2016]. The system first constructed an object-to-light linking table in a prepass render, and then used this table in-render to exclude certain lights from being considered for a given piece of geometry. However, no automated light culling system has existed in the RIS-era until now.

With regards to geometry culling, Pixar employs *metrics pruning*, that removes out-of-frustum, occluded, or tiny objects in a scene to reduce memory usage and speed up renders. However, lights that meet the criteria for metrics pruning can easily contribute meaningful illumination, so such geometry-based techniques do not directly transfer to light culling.

The Disney Hyperion renderer executes a pre-process before each render called *Cache Points* that locally computes important lights for thousands of regions in a scene, reducing light selection cost and wasted light samples [Burley et al. 2018]. Our approach instead caches lighting data in a cheap, low-resolution probe render, such that subsequent renders do not pay the cost of a pre-process.

2 PIPELINE

The way Pixar makes films has changed over recent years, with tighter schedules and several departments working on shots in parallel. Most notably, nearly all departments view shots in the context of lighting (as opposed to cheaper GL renders that convey character performance but not shading & lighting). This means that lighting is placed in shots earlier than before. Coupled with the fact that more departments are requesting renders with lighting and the general rise of lighting complexity, the burden on the render farm is immense. Further, rising light counts slow down interactive workflows, with lighters having to wait on average 6 minutes every time they start a new render before seeing pixels back.

Light Pruning was created to address these problems. As soon as a shot has lighting in it, a probe render is spooled (either by a cron

SIGGRAPH '19 Talks, July 28 - August 01, 2019, Los Angeles, CA, USA

ACM ISBN 978-1-4503-6317-4/19/07.
https://doi.org/10.1145/3306307.3328161

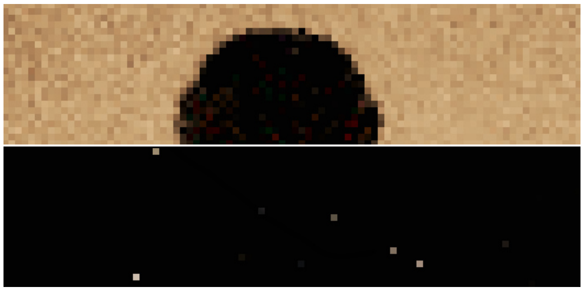

Figure 1: Light pruning aims to preserve useful lights (top) while removing non-contributing and spike-noise producing lights (bottom), which are common in ray-traced rendering. *Exposure raised for clarity.* ©Disney/Pixar.

job or on demand). This special render includes every five frames at a quarter of the usual resolution. Accompanying each beauty pass are Arbitrary Output Variables (AOVs) generated by light path expressions (LPEs). All lights are spatially clustered such that each AOV contains the contribution of a group of nearby lights. We set the number of clusters to be 125, as above this number the CPU utilization of rendering declines. Upon completion of the probe render, a post-render script computes per-cluster metrics including the max (R,G,B) value of each AOV. These per-cluster metrics are translated into per-light metrics and written to a USD file for consumption by Katana. In addition to writing out the max (R,G,B) of the raw AOV, we also firefly-kill the image (using median filtering) and then recompute the max (R,G,B), which allows us to prune lights that contribute nothing but spike noise [Fig. 1]. Fireflies increase the variance at a pixel, requiring more samples for convergence. By pruning firefly-producing lights, we reduce both rendertime as well as one-off shot fixes.

Inside Katana, per-light metrics from the USD file are written as attributes onto each light, and a Lua Op iterates over each light, queries its firefly-killed max (R,G,B), and prunes that light if its contribution lies below a threshold. We carefully tune system parameters such that small but useful light sources (like eye highlights) do not get pruned. We observe through perceptual studies that human eyes are more sensitive in dark scenes and therefore we must be less aggressive with pruning lights in such situations, whereas in bright scenes we can be more aggressive. After broad tests, dim frames get a threshold of 0.00065, and bright frames are thresholded at 0.005. Whether the max of the median (R,G,B) value of the beauty pass exceeds 0.05 determines whether a scene is dim or not (a heuristic we developed through testing). While default parameters generally avoid changing the image, the tunable pruning threshold enables rendering optimization artists to interactively reduce light counts (and render times) in exchange for subtle look differences in the render. To deal with edge cases, we create a Katana whitelist macro for artists to protect lights from pruning. Roughly 3% of shots have required manual whitelisting, and this fraction has dropped as production progressed and we improved system design and parameter tuning to accommodate more challenging cases and new light types. We note that each shot consists of several passes rendered (and light pruned) independently and later composited together, such as solid elements (containing characters & sets), fog/atmosphere, particulate, and sky. Since lights are pruned per-shot and not per-frame, we need not worry about temporal coherence issues that could produce flickering artifacts. Optionally accompanying each probe render are verification renders of the shot - one with light pruning enabled, and another with light pruning disabled, to visually confirm the absence of changes to the image and to measure render performance improvements.

To avoid unintentional pruning due to changes in the lighting setup (i.e. a lighter choosing to brighten lights previously deemed prunable), the Katana Op detects lights that have changed since the most recent probe render and does not prune them. To do so, a per-light hash is created and stored in the USD file at the end of the probe render. This hash takes into account several lighting attributes including position, intensity, and color. The Katana Op compares the current hash with the cached hash before considering the light for pruning. Whitelisted lights due to changes in their attributes again become eligible for light pruning when the probe render is re-run.

Our pipeline utilizes a C++ AttributeFunction for K-means clustering of lights, Python for custom render farm job preparation and processing EXR images, the USD Python API for storing light metrics, and Lua Ops for in-Katana pruning. Although Pixar uses RenderMan, any rendering software with the capability to define LPEs to generate AOVs can support our light culling strategy.

3 RESULTS

In broad tests, light pruning reduces global render times by 16%. In challenging cases with high lighting complexity, the savings are 20-50% and several GB. Interactive workflows also see a benefit: the typical Time-To-First-Judgment (TTFJ) measuring time between an artist clicking render and seeing pixels back to evaluate the image is 2-4 minutes for a moderately sized shot and 10-20 minutes for heavy shots. Roughly 6 minutes is the global average. After light pruning, this figure drops to 4.5 minutes. If a lighting artist launches an interactive render 20 times per day, light pruning can save the artist 30 minutes per day in TTFJ costs.

4 FUTURE WORK

Any optimizations that make the probe render cheaper or more aggressive in pruning non-contributing lights would enhance the net benefits of the system, and we are actively improving the project from both angles for future shows.

ACKNOWLEDGMENTS

The author thanks Jeremy Newlin, Steve LaVietes, Emily Weihrich, Joao Montenegro, Alexander Kolliopoulos, Tomas Nettleship, Yasmin Khan, James Jackson, Reid Sandros, Brett Warne, Mark Vandewettering and Fernando de Goes for contributing to the project.

REFERENCES

Brent Burley, David Adler, Matt Jen-Yuan Chiang, Hank Driskill, Ralf Habel, Patrick Kelly, Peter Kutz, Yining Karl Li, and Daniel Teece. 2018. The Design and Evolution of Disney's Hyperion Renderer. *ACM Trans. Graph.* 37, 3, Article 33 (July 2018), 22 pages. https://doi.org/10.1145/3182159

Renee Tam, Bill Reeves, and Justin Ritter. 2016. Efficient feedback-based illumination and scatter culling. (Jan. 2016).