Artist Friendly Level-of-Detail in a Fur-filled World

Sean Palmer* and Kendall Litaker†
Walt Disney Animation Studios

Figure 1: *(a) A final render of hundreds of furry characters in Zootopia, (b) A preview of fur with LOD applied (right) and without (left).*

Abstract

The geometric complexity required to create a city filled with furry animals in Walt Disney Animation Studios' Zootopia necessitated a new approach to level-of-detail for our in-house primitive generator, XGen. The characters spanned the breadth of the animal kingdom, from a mouse to an elephant, and each presented challenges with scale and fur quality. Early test scenes proved unmanageable to render with even a few characters and we knew some sequences called for thousands of them. To address this, we updated XGen's underlying pruning algorithm, refreshed the user interface, and developed a new wedge rendering tool designed to help streamline the fine-tuning of level-of-detail settings. These updates provided quick and informative visual feedback to the artists and aided in optimizing scene generation time and memory use.

Keywords: level of detail, animation, rendering

Concepts: **•Computing methodologies → Rendering;**

1 Improved Pruning

Level-of-detail is a procedural and temporally coherent method by which geometric density is reduced; in the case of fur, this is achieved by simplifying or eliminating primitives as the overall groom moves further from camera. Before Zootopia, XGen used a stochastic algorithm to control pruning: primitives in the groom were randomly chosen to be scaled down and removed, while the remaining ones were widened to fill in the gaps. The random nature of this culling could create an uneven distribution of the remaining primitives, resulting in patchy grooms at farther distances. This patchiness became noticeable even at a moderate distance from the camera owing to the discrepancy between the fur and skin colors, which was often done for physical correctness. This limited the amount of pruning artists could use, especially at closer camera distances, which in turn limited the algorithm's computational savings.

*e-mail:sean.palmer@disneyanimation.com
†e-mail:kendall.litaker@disneyanimation.com

SIGGRAPH '16, July 24-28, 2016, Anaheim, CA,
ISBN: 978-1-4503-4282-7/16/07
DOI: http://dx.doi.org/10.1145/2897839.2927466

To improve this, we took advantage of the fact that XGen already has a precomputed uniform sample distribution to ensure primitives are spread out evenly as they are emitted on each face of the mesh. We kept the same basic pruning behavior (choosing one set of hairs to shrink and remove and the rest to retain and widen) but instead of randomly selecting which group a primitive belongs to, we simply dialed back the number of primitives emitted in the first place. This greatly improved the distribution of the retained primitives and allowed artists to prune characters much closer to camera (see Fig. 2). As XGen emits primitives per face, not mesh, it is possible to lose this even distribution when only a few hairs remain per face, but in practice, we were able cull upwards of 99% of the grooms without encountering issues.

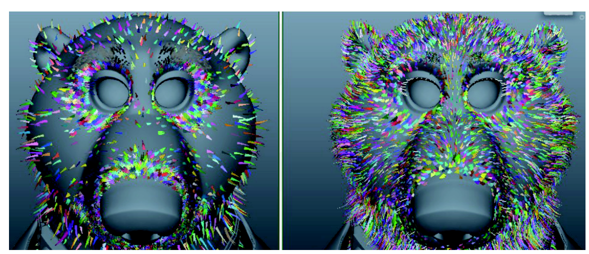

Figure 2: *Spotty primitive distribution when using a pure stochastic pruning (left) versus reducing the build amount per-face (right)*

2 Improved Artist Interaction

Understanding and controlling how the level-of-detail algorithm operates can be challenging for the uninitiated, and due to the variety of grooms, reusing settings for other characters was often impossible. The previous XGen user interface had limited and unintuitive controls. The algorithm computes level-of-detail from the render-time camera data, using artist-defined near and far pixel width values to indicate where the transition zones begin and end. Artists had to try to estimate settings based purely on the visual result, which was difficult to do reliably without human error. We streamlined the user interface, and added a UI button to automatically compute the near and far pixel width values based on a user-specified camera view. The simple boundary controls were extended to include optional intermediate steps as way-points for

Sheep	No LOD	LOD	LOD + CV
Memory Usage	2.47GB	0.18GB	0.062GB
Scene Conversion	142.9s	18.0s	19.2s
Render Time	296.0s	148.0s	45.0s

Table 1: *Memory use, scene conversion time, and render time for the sheep in Zootopia with and without CV reduction.*

more nuanced dynamic pruning. Overall settings could be made for all of the grooms of a character, and there are options to allow for further refinement where necessary. The algorithm is designed to produce a smoothly transitioning result across frames, but for rare cases where the pruning change was noticeable, artists had the option to lock-off a static camera to compute level-of-detail consistently. We also support per-shot modifications of level-of-detail settings to fine-tune the pruning based on a specific shot layout.

In addition to culling entire hairs, we added the ability to dynamically reduce the control vertices per primitive by resampling them to a lower point count. This allowed for an even greater reduction in geometric weight than pruning full primitives alone could provide, and both techniques could be used in concert for maximum efficiency (see Table 1). This incurred a small increase in groom generation time, but was minor compared to the reduction in memory use that resulted.

3 Automation

While having these interactive setup controls made dialing level-of-detail more intuitive for artists, rendered images were still needed for aesthetic approval. We developed a wedging tool to automate the setup and renders of different values for the major controls (see Fig. 3). Results were collected and composited so that artists could easily flip between before and after images. Statistics on render time, memory, and the number of primitives reduced were embedded in the rendered images. The standardization of this process made it possible to compare performance across characters and define reasonable time and memory saving goals for fur generation.

Figure 3: *Example output from the wedging tool showing the character at 4 canonical distances with pruning stats.*

4 Expanding to Environments

Due to the success of the process with characters, we extended the level-of-detail workflow for environments. As the XGen pruning algorithm was at that time restricted to curve primitives such as grass, its use in environment assets was limited. A simpler approach of culling all primitives outside of the camera frustum often afforded the most savings. There were, however, cases where level-of-detail still allowed for significant savings. Due to the much larger area of coverage for grass on ground planes compared to a single character, the wedge tool's initial camera setup based on the element's bounding box was too limited for artists' needs. Instead, pixel values for the transition zones were calculated from a top-down view of the element, which ensured all the curves would have roughly the same pixel width for the entire geometry. Once those values were calculated, the artist was able to render using actual shot cameras to see how the algorithm performed (see Fig. 4). This workflow provided the context that environment artists needed to determine whether or not their settings would hold up when used in actual shots.

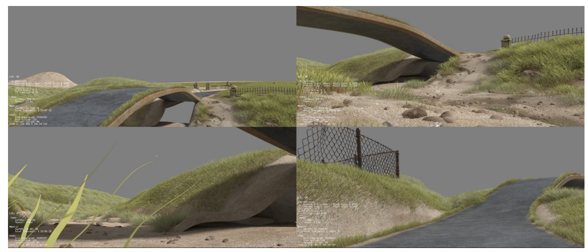

Figure 4: *Another render wedge showing multiple artist-selected cameras to judge the effect of LOD in context.*

5 Results

Our updates to the level-of-detail process allowed for major reductions in memory use and scene generation times, to the point where hundreds of furry characters could be rendered together manageably (see Table 2). With the new user interface and automated wedge tools, artists were able to iterate quickly and confidently over the settings and verify the visual results before releasing optimizations. For most cases, the artists were able to cull up to 95% of the primitives at their maximum prune thresholds without losing integrity. There are many avenues to extend this work. The level-of-detail algorithm in XGen has since been updated to operate on any type of geometry, which will make this system more effective for environments. Extending the render wedging tool to tune attributes outside of XGen, such as shader or tessellation values, would prove useful for other departments to optimize a broader range of assets.

Character	Statistics Type	No LOD	LOD
Fox	Memory Usage	1.09GB	0.044GB
	Scene Conversion	81.2s	26.8s
Sheep	Memory Usage	2.47GB	0.062GB
	Scene Conversion	98.2s	19.2s
Bear	Memory Usage	0.67GB	0.043GB
	Scene Conversion	53.6s	14.2s
Ground Plane	Memory Usage	58.88GB	14.37GB
	Scene Conversion	4549.4s	2996.0s

Table 2: *Scene generation time and memory use for various assets using XGen LOD on Zootopia.*