

The Jungle Book: Management, Caching and Preview of Many Animals

Marco Romeo Ryan Chan Je-Ren Chen Greg Fisher
MPC*

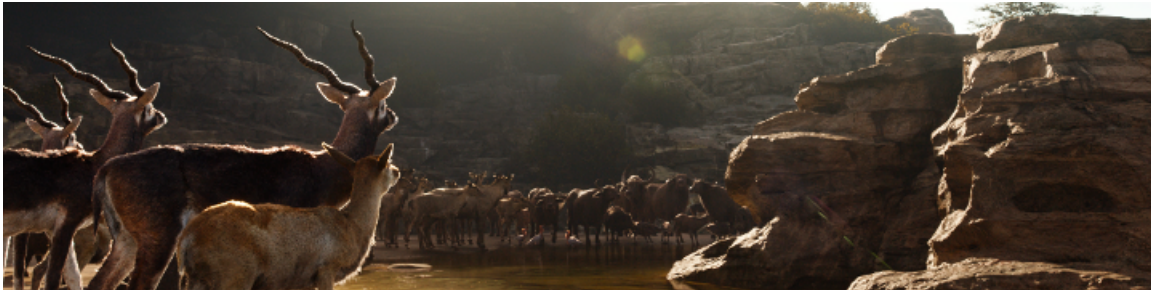


Figure 1: A shot from *The Jungle Book* with 47 characters. ©2016 Walt Disney Pictures. All rights reserved.

Abstract

The creation of the visual effects for *The Jungle Book* was characterized by shots with many different animals (the largest shot had 207 creatures). These animals interacted with each other and the movements of some were related to or affected by the movements of others. Because of this, each character needed to be animated with a notion of the movement of the surrounding animals. Unfortunately, given the complexity of each character mesh and rig, it was not possible to load, visualize or animate large numbers of characters at once.

At MPC, we developed a system that integrates our Rigging, Animation and Crowd pipeline and technology to enable the caching and fast previewing of animated characters in complex shots within Maya. This way artists could animate one animal, while the other animals were displayed at a reasonable playback speed. Furthermore, the system enabled artists to easily switch across different levels of detail (LOD) of the characters. We provide performance evaluation from production data for loading and playback of different amounts of characters.

Keywords: visual effects, automation, crowds, pipeline

Concepts: •Computing methodologies → Computer graphics; Animation;

1 A Crowd Solution to Crowded Shots

Animating many characters in a single scene for *The Jungle Book* was challenging for several reasons. As characters had a dense mesh and a complex deformation rig, the playback performance was good enough to interact with one animal but too low when displaying multiple characters. Furthermore, the complexity of the characters also meant that the loading times were too long to be

*e-mail:(marco-ro, ryan-ch, jeren-c, greg-f)@moving-picture.com

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). © 2016 Copyright held by the owner/author(s).

SIGGRAPH 2016 Talks, July 24-28, 2016, Anaheim, CA

ISBN: 978-1-4503-4282-7/16/07

DOI: <http://dx.doi.org/10.1145/2897839.2927414>

managed efficiently and that the graphic workstations would run out of memory when too many characters were loaded. To reduce the issue, we used lower resolutions of the characters, but the impact on playback performance and memory was still too big to allow multiple characters in the scene.

One possible solution was to solve the most complex shots as crowds. Unfortunately, given the creative control required and the different unique characters in the scene, this was not enough. Thus, we designed a system that combined the efficiency of the Crowd pipeline with the flexibility required by the Animation department. This fulfilled the need to produce hand-crafted animation for unique characters and efficiently preview them as crowd agents.

2 Crowd GPU Rendering

Over the years digital crowds have been an integral part of MPC's production pipeline. As the demand for crowd animations grows in visual effects, we are required to simulate complex crowd behaviors in massive numbers, often in the size of hundreds or even hundreds of thousands. To effectively visualize the result of the simulated crowds, we utilize GPU-accelerated hardware in our crowd cache viewer to enable fast rendering of crowd animations.

Our crowd cache viewer uses GPU shader programs to speed up crowd rendering in two stages: skin deformation and skin shading. In the skin deformation stage, an OpenGL vertex shader is used to apply the transformation of the skeleton cache to deform the geometries of each character accordingly. Each skeleton cache file contains all the transformations for each agent of the crowd at each frame. For skin shading, a fragment shader is applied to simulate the lighting and color of the characters.

For *The Jungle Book* we extended our cache viewer to work with skeleton caches produced by the Animation pipeline considering each character as a unique crowd agent.

3 Caching

Before *The Jungle Book* animation skeleton caches were created only when a new animation version was delivered. Unfortunately, the skeleton cache was not made available until other relevant data (e.g. geometry cache) was ready.

To overcome this issue and enable artists to immediately preview their animations using the cache viewer, we enabled the pipeline to

generate skeleton caches directly within the active Maya scene and automatically store and retrieve them on the artist’s local storage. By storing the caches at locations accessible to animators, we allowed them to interchange caches and update their scene to display the progresses of their colleagues.

4 Managing Characters

Through the animation of a shot, artists needed to use different versions of the characters, depending on their tasks (Figure 2). For the film we animated characters using high resolution body rigs (detailed geometry), low resolution body rigs (simplified geometry) and facial rigs. Furthermore, this development added the possibility to view the animated character as a GPU-skinned mesh. We designed a simple interface that provided a way to easily switch across different animatable character resolutions and generate fast cached previews.

To enable animators to rapidly switch between preview and animation modes we had to keep rigs hidden in the scene, while a preview was displayed. To improve the performance in scenes with hidden characters we worked with Autodesk to prevent Maya from traversing hidden hierarchies. Thanks to this, we were able to avoid performance loss caused by unnecessary evaluation of dependency nodes.

At MPC animated characters are assetized as hierarchical structures which define their geometry, rig and per-shot animation [Butler et al. 2008]. This enables us to properly manage animated assets through the pipeline but it also requires a complex offline process to update them through the pipeline; this might make latest rigs updates not immediately available. To help the Animation department to reduce the waiting times, the system enabled artists to safely access different versions of the rigs at any moment, without having to wait for the full pipeline update. To properly control this, the system provided feedback about the used version and informed the artist about other available versions. Finally, the system stored crucial information about the interaction between the user and the pipeline to ensure consistency with the pipeline and persistence of data when saving and opening scenes.



Figure 2: Different levels of detail for characters geometries. From left to right: high resolution (707216 triangles), low resolution (44162 triangles) and preview (14052 triangles). The quality of the preview version is comparable to the low resolution geometry. ©2016 Walt Disney Pictures. All rights reserved.

5 Evaluation

We ran a simple evaluation to assess how the animation previews performed, compared to rigged characters. We tried to load and playback different amounts of characters for the shot in Figure 1. The characters used in the test were comparable in terms of mesh resolution and complexity of their rig. Figure 3 shows the results for loading time (seconds) and playback speed (frames per seconds) for one, two, five, ten and twenty characters in rigged (high and low)

and preview versions. For more than twenty characters the performance of rigged characters was too low to be worth considering.

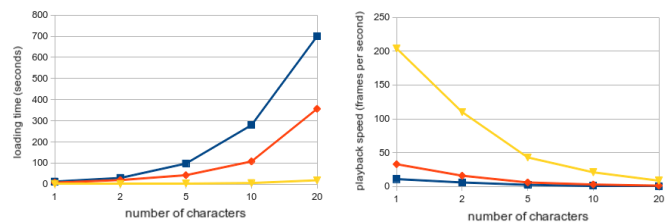


Figure 3: Loading times (left) and playback speed (right) for high quality rigs (blue), low quality rigs (red) and animation previews (yellow).

In the tests, our solution for animation previews outperformed rigged characters. On average, animation previews loaded 26 times faster than high quality characters and 12 times faster than low quality. Furthermore, the average playback speed of cached previews was 17 times faster than high quality rigs and 7 times faster than low quality rigs.

With 20 characters it took 19 seconds to load the cached previews, which compares well with high (700 seconds) and low (357 seconds) quality rigs. Playback performance was 9 frames per second for 20 cached previews, 0.5 frames per second for high quality rigs and 1.3 for low quality rigs. Finally, we performed some tests with large amounts of characters on two shots from the film: 47 characters were previewed at 4.5 frames per second and 207 characters were previewed at 1.5 frames per second.

6 Conclusions and Future Work

The system enabled animators to be more efficient when working with many characters in shot. It proved to load large sets of animation previews within reasonable time and enabled animators to animate required characters in context with others.

Future work will focus on improving the performance of the cache viewer by implementing the parallel buffering of cached frames. We also plan to implement a tool to support and foster the interchange of caches across animators.

Acknowledgements

We would like to thank the Crowd and Rigging departments at MPC, especially Adam Davis, Tom Reed, Marco Carboni and Ben Jones. We would also like to thank Gabriele Zucchelli, Marlène Chazot, Thomas Stevenson and all the artists from the Animation department. Finally, we thank Mila Grigorova for developing the first prototype.

References

BUTLER, G., LANGLANDS, A., AND RICKLEFS, H. 2008. A pipeline for 800+ shots. In *ACM SIGGRAPH 2008 Talks*, ACM, New York, NY, USA, SIGGRAPH '08, 72:1–72:1.