

The Trees of *The Jungle Book*

Alexander Schwank*
MPC

Callum James James†
MPC

Tony Micilotta‡
MPC



Figure 1: Typical jungle scene - ©2016 Walt Disney Pictures. All rights reserved.

Abstract

For Disney’s live action remake of *The Jungle Book*, we were required to build a jungle comprised of photorealistic trees, vines and other plantlife. We leveraged SpeedTree [IDV 2016] and Fabric Engine [Fabric Engine 2016] to create more than two hundred unique trees and vines that were used in the mid- and background of the jungle. To maximize the tree look quality, we created high resolution trunks and branches in Maya and included these during the node-based configuration stage in SpeedTree. Moreover, with each tree having approximately one million leaves, the leaves were approximated with cards and later replaced by high resolution leaf model configurations during the render process; this reduced model complexity, facilitated technical animation, and ultimately enabled us to handle the vastness of the jungle.

Keywords: trees, leaves, pipeline, SpeedTree

Concepts: •Computing methodologies → Computer graphics; Shape modeling;

1 Creating Highly Complex Trees

SpeedTree enabled artists to procedurally model complex trees of various species with different styles. To increase the detail of certain trees we modeled trunks and branches in Maya and incorporated them into the SpeedTree modeling process. This allowed us to use high resolution models with our own UV layout while benefiting from SpeedTree’s procedural tree generation.

All leaves were represented as simple cards, making it possible to keep the tree models lightweight in SpeedTree. With the use of

*e-mail:alexander-sc@moving-picture.com

†e-mail:callum-j@moving-picture.com

‡e-mail:tony-mi@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 ’16 Talks, July 24-28, 2016, Anaheim, CA,

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

DOI: <http://dx.doi.org/10.1145/2897839.2927428>

leaf textures with alpha masks we were still able to review the tree model directly in SpeedTree.

All trees were built with the highest required level of detail (LOD) and subsequently simplified to generate lower LODs. This was achieved by two processes in Maya. The first deleted fine detail such as little twigs, randomly culled a percentage of leaves, and up-scaled the remaining leaves to maintain the volume of the tree. The second process reduced and simplified the model irrespective of the tree hierarchy using Maya’s mesh reduce feature. A representative tree LOD comparison is shown in figure 2.



Figure 2: LOD comparison - left to right: LOD B with 657K polygons, LOD C with 49K polygons, LOD D with 7K polygons. - ©2016 Walt Disney Pictures. All rights reserved.

2 Pipeline

We implemented a pipeline based around Fabric Engine, a digital content creation platform that provides the foundation for the various components required for building 3D content creation tools [Fabric Engine 2016]. Our tree pipeline extracted trees from SpeedTree and imported them into Maya, where we were then able to use established functions to make the trees available in MPC’s general pipeline.

The Fabric pipeline took the SpeedTree hierarchy and converted it into a hierarchy of Maya meshes. With access to the original SpeedTree hierarchy, the pipeline was able to differentiate tree parts and their importance for the look, and use it for the LOD creation. In the resulting structure, meshes were grouped by branch and element type allowing for easy manipulation of the model and isolation of elements in Katana for lookdev. This was impossible with standard export file formats due to the missing understanding of

the tree hierarchy. Ultimately, this improved expansion and render performance.

In addition to the meshes, the pipeline was able to use the tree hierarchy information from SpeedTree to generate Maya skeletons and skinClusters for the TechAnim department. Further to this, the pipeline converted and prepared SpeedTree's generated textures for MPC's Texture department. At this point, the model, skeleton and texture were released into our pipeline in the usual manner. The export from SpeedTree and the packaging were distributed on the farm to reduce the artist machine load.



Figure 3: Full LOD A tree model example (leaf cards with texture and alpha mask) - ©2016 Walt Disney Pictures. All rights reserved.

3 Animating and Rendering Complex Leaves

Representing leaves as cards gave artists the freedom to swap out and scale these cards with any leaf model further down the pipeline without needing to recache any animation (see figure 4). Moreover, this configurability allowed us to drastically change the tree appearance near the end of the pipeline. Bending leaf cards in Katana instead of replacing them was an additional option, however we found that mesh replacement produced more realistic results.



Figure 4: Tree with leaf cards compared to the final render with replaced leaves. - ©2016 Walt Disney Pictures. All rights reserved.

Depending on the tree's LOD, the leaf cards were replaced by leaf meshes with a suitable polygon count for LOD A and LOD B. Background trees with a lower LOD typically used textures with alpha masks on the simple cards without the need to use actual leaf models to create a realistic look.

Furthermore, artists were able to add a second leaf texture layer to create realistic color variations, similar to the way fur color variations are done at MPC. Since tree leaves have overlapping UVs in a few UDIMS (depending on the number of leaf variations), we unwrapped all leaves into a single UDIM. In MARI, artists were then able to paint a variation map that affected the entire canopy without having overlapping textures. This variation map was then multiplied with the color texture in Katana. In addition, each leaf had a unique ID assigned which could be used for further, shader based color variations.

Ultimately, the tree vertex count was much lower compared to trees with full leaf meshes. Benefits included a smaller data footprint, simplification of the tree simulation process and significantly improved responsiveness in Katana. Compared to the approach of instancing leaves, this approach had the advantage of being more efficient due to the low vertex count per leaf.

4 Future Work

We intend to extend the use of Fabric for the SpeedTree export and tree preview to remove the reliance on Maya. Moreover, we plan to use SpeedTree's internal wind animation for background trees.

Acknowledgements

To Fanny Chal on, Greg Croft, Jonathan Attenborough, Peter Pearson, Rui Padinha, and Yoran Radecker, for all the help, input and enthusiasm.

References

FABRIC ENGINE, 2016. <http://www.fabricengine.com/>.
IDV, 2016. Speedtree. <http://www.speedtree.com/>.