Termite: DreamWorks Procedural Environment Rigging Tool

Chris Michael
DreamWorks Animation
chris.michael@dreamworks.com

Arunachalam Somasundaram
DreamWorks Animation
arun.somasundaram@dreamworks.com

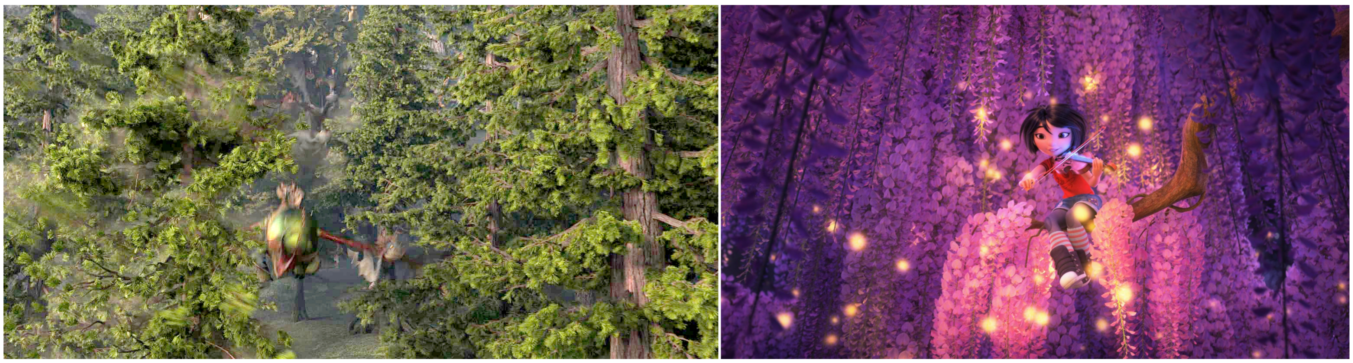

Figure 1: Example environments in film shots that have been rigged and simulated using Termite

ABSTRACT

This talk presents DreamWorks' *Termite*, an environmental rigging and simulation utility used by the Character Effects Department (CFX) for rapidly creating simulation setups for environment assets without pipeline dependent complexity requirements or additional data carried by the geometry itself, such as hierarchy attributes. With minimal artist input, high quality environment effects can be achieved quickly and easily. The system was used on DreamWorks' films *How to Train Your Dragon: The Hidden World*, *Abominable*, and *Trolls World Tour*.

CCS CONCEPTS

• **Computing methodologies → Physical simulation**; **Procedural animation**; **Shape modeling**; **Collision detection**.

KEYWORDS

environment, rig, automation, motion, simulation, effects

ACM Reference Format:
Chris Michael and Arunachalam Somasundaram. 2020. Termite: DreamWorks Procedural Environment Rigging Tool. In *Proceedings of SIGGRAPH '20 Talks.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3388767.3407384

1 INTRODUCTION

During the production of *How to Train Your Dragon: The Hidden World*, CFX became tasked with the majority of each location's plant environment behaviors. With additional productions interested in these results, CFX needed a way to better automate the creation and deployment of simulation setups for environment assets. However, as these assets do not carry the same conventions as a character or animated structure in regards to motion, a number of issues arose.

- Many of the environment assets had no clear naming convention across parts or components, and those that did were inconsistent from asset to asset.
- Most assets were designed with no true hierarchy in mind during the creation process, rather they were constructed per art-direction without a true structure taken into consideration.
- No existing workflow required these assets to carry attributes or stored information detailing their construction or hierarchy, even if that asset was constructed in a procedural way.

All of this meant that a solution to automate the creation of the environment simulation rigs could depend only on the nature and design of the geometry. Because these environment designs were specific to the styles of each film and created by studio specific processes, which also did not include control systems at creation, commercially available packages were not an option.

2 ENVIRONMENT RIGS

Termite takes a multi-stage approach to evaluate an environment asset to infer an asset's hierarchy using the structure of the geometry itself combined with the geometry's relative position to world origin at creation time.

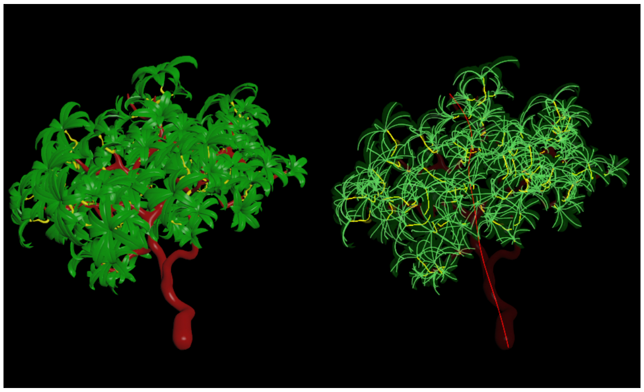

Figure 2: An example tree and its corresponding hierarchical curve simulation rig (with 3 levels) created by Termite.

2.1 Connected Mesh Assignments

During the first stage, *Termite* takes any part names that may have been provided to the asset from the modeling department, evaluates naming similarities between those parts, and provides the artist that information. If needed, the artist can guide the hierarchy creation based on those names. New component names are assigned to each connected mesh, and these names are used for binding the render mesh to the simulation curves.

2.2 Root Part Assessment

Termite then evaluates the new meshes, assessing which meshes have components nearest the world origin point, presuming those to be likely candidates for the base of the assets hierarchy. These results can be refined by specifying whether an asset has single or multiple root objects, and how liberal the tolerance of this evaluation is.

2.3 Hierarchy Evaluation

With root objects established, *Termite* begins evaluating geometry outward from these objects. A proximity based evaluation is performed on the components of each subsequent mesh to infer its relationship to the root objects. Each mesh is then assigned as a child of those root objects, or a child of those children, and outward until the intended hierarchy of the asset is established. Any geometry that is still orphaned is assigned as a child to the closest object in the outmost level of the hierarchy.

2.4 Curve Rig Creation

With this information, *Termite* converts the meshes to volumes and extracts volume medial curves associated with each mesh component for simulation. Bifurcation of split objects is currently disregarded for speed purposes and so far has not been necessary for our desired results. These control curves are given information about their parent curves, allowing for a hierarchical simulation setup where children inherit the motion of their parents.

These simulation rigs and binding information can then be written out for use in complex large scale environment simulations via DreamWorks' *Grasshopper* [1] system. Alternatively, *Termite* carries built-in utilities to allow the setups to be used immediately for environment motion, as a one-off setup for a complex environment asset.

3 ENVIRONMENT MOTION

Using *Termite's* deformation tools, dynamic motion can be applied directly to an asset after creating a simulation setup. *Termite* uses a layered motion approach to independently control the various contributors to the final result.

- Velocities sampled from fluid fields can be used to bend control curves, by using the custom *CurveBendByVelocity* node, for indirect interaction such as reacting to forces from wing flapping or blasts.
- An art directable procedural wind solution with controls for noise function types, direction, magnitude, frequency, and speed can be applied for ambient wind effects.
- Geometry contact collisions and subsequent recovery behaviors were handled by *FurCollide* [2].
- A fast custom *CurveJiggle* node using a spring simulation produces secondary motion.

These effects take into account the motion of the entire hierarchy. Artists can moderate and control levels of the hierarchy independently, allowing more restricted motion to the root objects if desired while outer thinner areas move more freely.

4 PIPELINE

Termite has the ability to manage pipeline compliant overrides, prepping animated assets for delivery to downstream departments. Overrides enable a full takeover for the animated asset including specific consideration to bind surfacing assets to the deforming geometry. Point instancer data used for leaves or other surface growth objects, are processed and attached as animated point information tracking the movement of the underlying surface. Wrapped procedural objects, like fuzzy moss or exposed threads, are also given overrides to inherit the parent asset motion. This functionality allows *Termite* to act as an authoring tool for animated geometry as well as a utility for the creation of simulation setups.

5 CONCLUSION AND RESULTS

Termite simplifies the time consuming process of manually creating simulation setups for environment assets, reducing a process that can take hours or days to minutes. Environment rigs can be created solely based on the nature and design of the geometry. This level of automation also expanded the development process to a wider scope of artists, while still providing enough input to make directed decisions about the setup. Consequently, the process to generate new setups is efficient, substantially reducing downtime when new assets are included late into a production. With its included deformation suite, *Termite* is flexible enough to fulfill a variety of roles, on a wide range of assets. On released films, *Termite* has been used to configure 188 unique environment assets and has been used to directly deform animated assets in 59 shots across three films.

REFERENCES

[1] Chris De St. Jeor, Chris Michael, Kurt Phillips, and Arunachalam Somasundaram. 2019. Grasshopper: Dreamworks Environmental Motion System. In *ACM SIGGRAPH 2019 Talks*. Article 62, 2 pages. https://doi.org/10.1145/3306307.3328174

[2] Arunachalam Somasundaram. 2017. FurCollide: Fast, Robust, and Controllable Fur Collisions with Meshes. In *ACM SIGGRAPH 2017 Talks (SIGGRAPH '17)*. ACM, New York, NY, USA, Article 55, 2 pages. https://doi.org/10.1145/3084363.3085051