Grasshopper: DreamWorks Environmental Motion System

Chris De St. Jeor
DreamWorks Animation

Chris Michael
DreamWorks Animation

Kurt Phillips
DreamWorks Animation

Arunachalam Somasundaram
DreamWorks Animation

Figure 1: Various environments simulated using Grasshopper

ABSTRACT

This talk presents DreamWorks' *Grasshopper*, an environmental motion system for creating believable animation for grass, plants, and other vegetation. The system was used extensively on the complex and expansive sets in *How to Train Your Dragon: The Hidden World* and is currently being used on more productions at DreamWorks.

CCS CONCEPTS

• **Computing methodologies → Physical simulation**; **Procedural animation**.

KEYWORDS

environment, foliage, vegetation, grass, plants, field, motion, simulation, procedural animation

ACM Reference Format:

Chris De St. Jeor, Chris Michael, Kurt Phillips, and Arunachalam Somasundaram. 2019. Grasshopper: DreamWorks Environmental Motion System. In *Proceedings of SIGGRAPH '19 Talks.* ACM, Los Angeles, CA, USA, 2 pages. https://doi.org/10.1145/3306307.3328174

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

ACM ISBN 978-1-4503-6317-4/19/07.
https://doi.org/10.1145/3306307.3328174

1 INTRODUCTION

While approaching the work required for *How To Train Your Dragon: The Hidden World [THW]*, it became clear that a significant amount of character interaction would occur with the environment. On previous shows, it was a common practice for set dressing to design specific acting areas with sparse vegetation as to avoid any character interaction. If interaction was desired, the motion effects were limited and data-intensive. Grass could be moved with a sparse set of motion guide curves, and plants could be manually and individually deformed through a technical animation workflow. *THW*'s broad scope of environmental interaction far outreached that level of complexity, thus a new system needed to be created to handle large quantities of data, produce aesthetically pleasing results in a timely manner, and be easily accessible by a wide range of artists.

2 EXTENSIBLE WORKFLOW

From the outset, *Grasshopper* needed to be flexible enough to accommodate a wide variety of new and different asset types. *THW* used asset instancing to efficiently set dress plants, while grass and fields of tall stalks were defined using render-time procedurals. The layout department could also break out any of these as hero assets at any time. Since many of the environmental set dressing techniques were being developed during production of *THW*, *Grasshopper* needed to be modular to adapt to these new asset types as they were created, without disrupting the artist experience. Each type of environmental element (grass, plant, field, etc.) was represented as an abstract node with the parent *Grasshopper* system

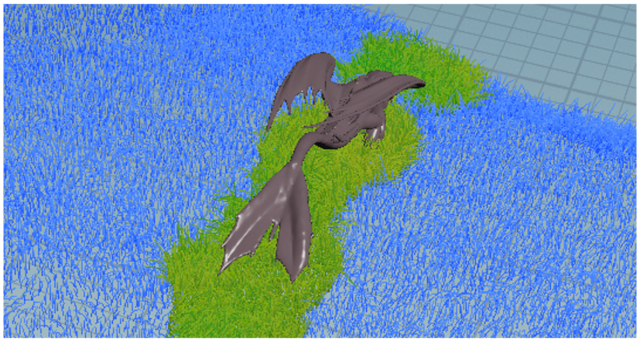

Figure 2: Automatic culling of grass field curves based on the path of the character in the entire shot.

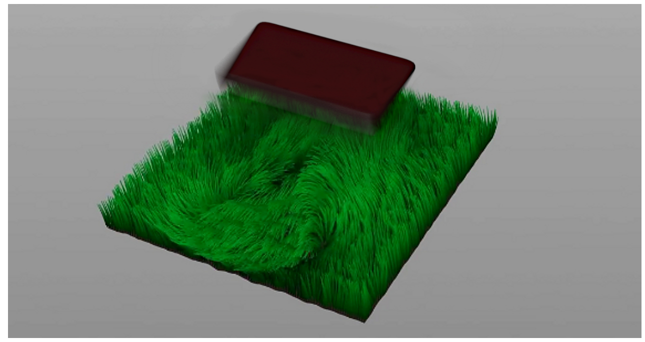

Figure 3: Organic swirl motion in a grass field obtained from a fluid simulation driven by flapping of a winglike object.

feeding these nodes common inputs to influence the motion. Additionally, each plant type had unique settings and custom control curves stored in a library, allowing for a wide variety of behaviors within the same framework. Thanks to this design, artists could execute a full environment simulation without needing to be trained to use multiple tools, or work in multiple files.

3 AUTOMATIC CULLING

Since a large amount of environmental elements needed motion for a single shot, *Grasshopper* was built to automate as many aspects as possible. This allowed the artists to make artistic decisions for their shot without concern for data management. For example, *Grasshopper* would automatically detect the characters' paths as well as the camera frustum to takeover only the needed environmental pieces in the shot. This could be further controlled to expand or reduce the desired regions to affect.

4 LAYERED MOTION EFFECTS

A layered motion approach was used to independently control the various contributors to the final result, including fluid simulations, wind, collisions, and secondary motion effects.

4.1 Indirect Interaction

For each shot, a fluid simulation was first run of all the characters moving through an air field. The more extreme the characters' motion, the stronger and wider spread their effect would be in the air field. Velocities were point sampled from this air field and transferred to the control curves. The deformation was achieved using a custom node that bends the curves based on these velocities. The effect could be further controlled after the velocities were cached using a custom region of influence node to scale the velocity around and away from the characters. These fluid dynamics also provided organic interactions with the environments, such as swirls of wind and the pull of air from a dragon's wings. This method provided artistically controlled indirect motion, providing results which could not have been produced otherwise.

4.2 Atmospheric Wind

Wind played an important role in the film, and was used to set the mood for many sequences. For this, *Grasshopper* utilizes a procedural solution with controls for direction, magnitude, frequency, and speed to design each location's wind effects. The wind settings were stored in a library so that all artists could produce consistent results across an entire sequence.

4.3 Contact and Secondary Motion Control

Collisions and subsequent recovery behaviors were handled by *FurCollide* [Somasundaram 2017]. A custom jiggle node was built using a spring simulation to produce secondary motion. This node also helped with softening the motion, especially when characters interacted with a dense environment. The jiggle node's parameters, such as mass, spring constant, and spring damping could be controlled per curve and was setup procedurally to organically vary across the elements to produce visually rich motion. Finally, a custom mesh contact node was built to procedurally resolve interpenetrations between collision geometry and ground surfaces to provide a cleaner scenario for the plant and grass interaction.

5 IMPLEMENTATION

Grasshopper was implemented in a node-based procedural commercial package. Custom multi-threaded nodes were built using C++ and the commercial package's expression language, and were optimized to handle large quantities (tens of thousands) of curves. *USD* overrides were used to wholly replace plant instance data in affected areas with animated *Alembic* caches. Plants that did not require complex deformation simply received overrides to animate the transforms of those instances. All overrides were referenced into a single *USD* layer for delivery to other production departments.

6 CONCLUSION AND RESULTS

Grasshopper allowed the *How to Train Your Dragon : The Hidden World* production to design environments and sets unlike any DreamWorks production that came before it. In total, *Grasshopper* handled 102 different plant types and 52 unique grass sets, across 13 different locations. These locations were populated by as many as 4.5 million grass curves and thousands of instanced plants. A single shot could result in an artist simulating tens of thousands of environmental elements. Since *Grasshopper* could integrate these large numbers of assets and simulate them with the same approach, the results are consistent across whole locations and sequences.

REFERENCES

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