Furtility: Robust Hair Styling

Curtis Andrus* MPC

Mark Manca† MPC

Abstract

MPC's proprietary fur creation system, Furtility, is used by our Groom department for all the fur, hair and feathers on our characters. Furtility is a procedural tool, sampling the character's mesh and growing fur at each sample on-demand. This is done by applying a sequence of *GOP*s (Geometry Operators) to curve geometry spawned at each sample location. This allows the artists to give the fur its desired characteristics (curls, clumps, frizz, etc.). The Mesh Style GOP, first created for *The Chronicles of Narnia: Prince Caspian*, allows artists to control the overall shape of the groom from a sparse set of manually groomed *guide curves*. Nowadays it's used in nearly every groom, and the number of guide curves can get into the thousands for a creature covered in fur (roughly 5000 for a character on an upcoming project). To handle the increasing number of furry characters in our projects, we re-worked a core component of this GOP to be much more robust and usable by the artists.

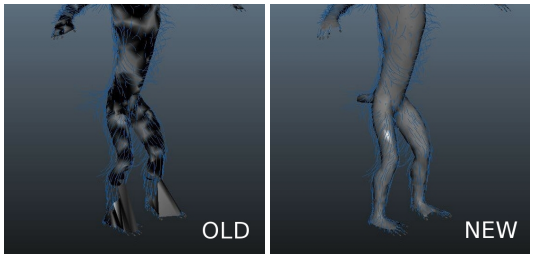

Figure 1: *Old vs New Style Mesh Generation - ©2014 Twentieth Century Fox. All rights reserved.*

1 Motivation

The Mesh Style GOP determines the shape of a fur curve by interpolating between the nearest guide curves. The interpolation weights are computed by first building a triangulated mesh from the root points of each guide curve. We refer to this mesh as the *style mesh*. Next, each fur curve finds the closest point on the style mesh. Since each vertex of the style mesh corresponds to a guide curve, the barycentric coordinates of the closest point can be used as weights. This gives an interpolation that is smooth and will exactly match the guide curves for close enough hairs.

The problem, however, lies in the generation of the style mesh. Our current tool attempts to "fit" a watertight mesh to the guide curves, ignoring the original surface mesh. Too few curves, or even a slightly misplaced curve, can cause the algorithm to fail unexpectedly. Additionally, there's no guarantee that the style mesh will look anything like the original (see Figure 1), since it only takes the root points as input. This can cause confusing interpolation results on thin parts of the mesh, such as ears and fingers. Until recently, artists worked around it through a process of trial-and-error. Unfortunately, they aren't always able to correct the issue, leading to a large amount of frustration.

2 Improving Style Mesh Generation

To alleviate the artists' pain, we developed a new method for building the style mesh. Unlike the old method (which effectively treats the guide curves as a point cloud), our new method relies heavily on the surface mesh, using it as a guide for the style mesh's topology. This allows us to nicely handle thin parts of the character. Finally, the failure conditions of this algorithm are easier to understand and fix.

The algorithm works by "coloring" each mesh vertex with the ID of the nearest guide curve. Distance on the mesh is approximated by using edge lengths (which is reasonable, considering our hero meshes are quite dense relative to the guide curves).

We start by iterating over the guide curves and marking the vertex nearest to the guide curve. From there, a vertex's color gets passed to its neighbors if the associated guide curve is the closest seen by that vertex. If we see a vertex that's been marked by a closer guide curve, the traversal stops.

Now, each group of identically colored vertices forms a cell. Since neighboring cells correspond to guide curves that are nearby on the surface, we construct the style mesh by building edges between any cells sharing a border. The GOP then proceeds as before, using this new style mesh.

3 Results

We tested this on several production assets. Artists found it to be far more stable with a better visual appearance. Model components that were difficult for the previous method (limbs, ears, fingers, etc.) are handled cleanly by the new method. It also handles large numbers of guide curves, running at interactive rates on thousands of guide curves.

The GOP can succeed as long as the style mesh has a well-defined "closest point" function, so it is reasonably robust against non-manifold or degenerate bits in the style mesh. In a few cases where a guide curve needs to be thrown out, the reasoning is understood and helpful feedback can be provided to the artist.

Since the algorithm is heavily reliant on the mesh vertex density, there is an issue with guide curves getting ignored when they are too dense (i.e. more than 3 guide curves inside a single triangle). This can be fixed by implementing a subdivision scheme that ensures each guide curve gets its own unique vertex. In practice, this isn't a problem on hero models (vertices are typically more dense than the guide curves).

Overall, our Groom department has found this new method extremely useful. While it was initially meant for a single asset, it soon found use in all of our current shows.

*e-mail:curtis-a@moving-picture.com

†e-mail:mark-ma@moving-picture.com

SIGGRAPH 2015 Talks, August 09 – 13, 2015, Los Angeles, CA.
ACM 978-1-4503-3636-9/15/08.
http://dx.doi.org/10.1145/2775280.2792564