Mesh-Driven Generation and Animation of Groomed Feathers

Rasmus Haapaoja
MPC R&D
rasmus-h@mpcfilm.com

Christoph Genzwürker
MPC R&D
christoph-ge@mpcfilm.com

Figure 1: Groomed feather driven by a static (left) and deforming (right) mesh.

ABSTRACT

MPC's proprietary grooming software - *Furtility* - has been used to create all hair, fur and feathers on our characters with great success for over a decade. However, the creation of feather grooms has always been a time consuming and technically challenging task for our Groom department, often keeping a senior artist occupied for months. Due to narrower deadlines and a constant push for higher quality, we recently extended our feather tool set, which has allowed our artists to significantly streamline their feather workflow. After adopting our new geometry-based feather system in production, our Groom artists have been able to reduce the time frame for finalizing a hero character from months to weeks.

CCS CONCEPTS

• **Computing methodologies → Geometry models**.

KEYWORDS

procedural modelling, deformation

ACM Reference Format:
Rasmus Haapaoja and Christoph Genzwürker. 2019. Mesh-Driven Generation and Animation of Groomed Feathers. In *Proceedings of SIGGRAPH '19 Talks.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3306307.3328178

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

ACM ISBN 978-1-4503-6317-4/19/07.
https://doi.org/10.1145/3306307.3328178

1 BACKGROUND

A feather in Furtility is represented as two separate components: quills and barbs, essentially a top level groom with a sub-groom attached to it. The artist starts the grooming process by first creating a set of barb designs – feather templates – on a straight quill. A template defines the look of a single feather, which can then be applied to a selected set of quills distributed across the skin of the character.

Allowing our artists to animate and texture feathers comes with its own set of challenges compared to fur, where animation and texturing is performed on the skin mesh of the character. In our previous feather workflow, this was achieved by *wrapping* the groom to a set of feather geometries which would then drive the deformation and allow textures to be projected onto the curves. In practice, this process tends to be very time consuming for our Groom artists, constantly trying to find the sweet spot between following the animation and retaining the shape of the designed feather. Additionally, the artist is often required to maintain several variations of the same template, due to the lack of a natural way of adapting the design to any shape differences among the input geometries.

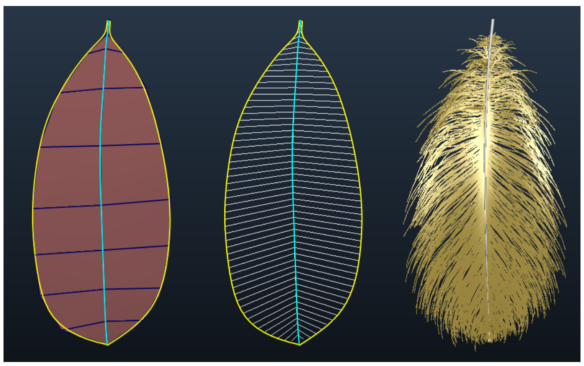
Over the past year, MPC's Software team has put in a substantial effort to revamp the core structure of our latest major iteration of Furtility [Andrus 2018]. The new, more flexible design has played a key part in our recent feature extensions, allowing us to construct our grooms from small building blocks without any hard-coded functionality, improving both code maintenance and efficiency.

2 FEATHER GENERATION

Considering the drawbacks of the previous feather wrapping solution described above, we decided to approach the wrapping-problem the other way around. Adopting the same standardized topology used for feather meshes by our Modelling and Rigging departments,

we can easily extract one central quill curve and two border curves, as illustrated in Figure 2a. Since the central curve directly represents the feather quill, we have already eliminated the step of having to place and groom these separately.

As shown in Figure 2b roots are then distributed along the quill and barbs grown to the corresponding parametric value along the border curves, giving us a perfectly wrapped feather from the get-go. This also guarantees that the profile of the groom will respond properly to any shape variations across the meshes, without any additional manual work. Once the initial shape has been generated, our standard set of grooming operations, as clumping, scraggles and curls etc., can be applied as usual. An example of this is provided in Figure 2c.



(a) Generated curves (b) Distributed barbs (c) Groomed feather

Figure 2: Curves (cyan and yellow) generated from a feather model, the initial barb distribution and the final groomed feather.

3 PROCEDURAL GROOMING

In addition to defining the initial profile of the feathers, the extracted curves also unlock new possibilities to how we allow our artists to perform the actual grooming.

We utilize the same parametric connection between the quill and the borders as mentioned above, and expose the ability to re-shape and bias this parametrization with e.g. ramps and noise textures. The influence of these textures can also be tweaked further, both along and across the feather to achieve the desired look.

Another feature that essentially comes for free is the ability to achieve the look of fluffy feathers, by shortening the length of the quill and letting the barbs wrap around the tip. Previously, this required a very precise fine tuning of inclinations and curls in order to connect the two sides seamlessly.

Due to the limited resolution of our models, the transition between the two sides might appear sharp and the overall look somewhat flat. We prevent this by generating a NURBS surface from our extracted curves, onto which the barbs are mapped. Making sure they always follow a *smooth* representation of the mesh.

As deformers are applied, the feather will naturally start to diverge from its original shape. Hence, we also expose the ability to project the barb tips back onto the border curves, in order to "reset" the contour at any stage in the deformer stack.

4 SIMULATION & ANIMATION

Due to the limitations of our previous feather system, simulation of feathers was somewhat limited to the quills. However, it's now possible for our Technical Animation artists to apply both cloth simulations and to solve collisions with the meshes directly. Our new system has the ability to follow the animation perfectly while still retaining the look created by the groom artist, without any trade-off between the two.

In order to guarantee stable results, the groom is first generated completely in the reference pose and then transferred to the animated counterpart. Barycentric look-ups and local offsets along the mesh normals are then used to retain the groomed shape during animation, of which an example is provided in Figure 3.

(a) Compression (b) Bending (c) Twisting

Figure 3: Feather groom driven by animated mesh.

5 CONCLUSIONS

Our new geometry based feather system has proven to be a significant addition to our Furtility tool-set, with our Groom department adopting it immediately in production. It has already proven its value in terms of turn-around times, with the first iteration of a full hero groom being created in days instead of weeks, and finalized in a couple of weeks instead of months. Additionally, we were able to transfer a character from the old to the new system, achieving the same look and reducing the time for generating the full groom by 30% at render time.

ACKNOWLEDGMENTS

The authors would like to thank Ben Jones for the initial concept and ideas, along with the London Groom department, especially David Mayhew, Gabriel Arnold, Fabio Messina and Michael Cauchi for providing invaluable feedback during the testing phase and for quickly adopting the tool in production.

REFERENCES

Curtis Andrus. 2018. Layering Changes in a Procedural Grooming Pipeline. In *Proceedings of the 8th Annual Digital Production Symposium (DigiPro '18)*. ACM, New York, NY, USA, Article 4, 3 pages. https://doi.org/10.1145/3233085.3233094