Rendertime Procedural Feathers Through Blended Guide Meshes
Daniel Seddon  Martin Auffinger David Mellor
Framestore

1 Introduction

In Monster ‘Stork’ the challenge was to create a system for efficiently creating and grooming photo realistic feathers for a CG bird. Feather grooming can be difficult, as there are a number of interdependent factors to be adjusted and controlled in order to achieve the final look. The appearance of the individual feathers is largely determined by both continuous and localised variations in their size and shape across the body. At the same time, each feather can have an individual textural quality, in terms of colour and barb shaping. We present a method which involves creating groomed feathers that are then blended by painting a weighting function on to a base mesh. This gives a continuous variation in groom across the bird’s body, allowing for control over the appearance of specific feathers as well as overall groom.

2 Basic Approach

The basic idea for the system is that the grooming process is broken down in to two stages.

In the first stage, a number of key feathers are chosen from the bird to be groomed. These feathers are then recreated with a procedural feather grooming system to give a matching appearance in terms of a number of pre-defined groom filter attributes. These attributes are:

- Splitting – imitates the splitting seen on feathers
- Scraggle - random noise used to displace the barbs
- Tangle – a scraggle that accumulates down the barb
- Clipping - takes random cuts along the length of the barbs

Real scanned feathers\(^1\) and their procedurally generated copies\(^2\). The smaller feathers have more scraggle / tangle and clipping, whereas the larger feathers have only splitting.

This procedural process takes place at rendertime using a Renderman DSO. A representative Bézier grid is passed to Renderman in order to give the shape and size, and also the painted groom attributes. Because of the shape of the Bézier, a feather is easily built – by first evenly distributing the barbs along the “\(v\)” parameter of the Bézier grid, and then applying the filters to give the look. The number of barbs on the feathers was determined by how many barbs of a given width and separation could fit on to the grid, which meant that the granularity could be controlled directly by these values. In all about 15 feathers were groomed for our bird.

In the second stage, once these filters have been used to match the appearance of the feathers on our stork, the grooming process takes place using these feathers as a template, with the groomed feathers determining the size and shape of the feathers on the whole of the bird. A base mesh is used as a grooming surface, upon which a weighting function for each feather is painted to define what fraction of each of the groom feathers is present on each section of the bird. After this the size of the groom feathers determines the feather density in a scatter function that distributes points across the mesh. These scattered points then use the weighting function, along with other grooming attributes such as orientation and lift, to build feathers at each point from the guide feathers. Following this, there are a number of ‘cleanup’ stages designed to deal with feather collisions and interpenetration. All of which gives a final output to the Renderman DSO which is a mesh of Bézier grids, enabling it to build smoothly interpolated but unique feathers to a variable degree of granularity for different sizes in frame.

3 Discussion

The method used in our commercial gave us great flexibility in changing what the feathers looked like. This was necessary in the case of a stork, as we didn't have the option of using recognisable texturing or colour, and the shaping/groom of a stork's feathers is one of its characteristics, making it very sensitive to the size and shape of the feathers. Also, several of the shots in the commercial needed the bird to have an aged appearance, and being able to resize and shape the feathers with simple modelling and displacements meant that this was easily accomplished over a few days at the end of the project.

The system could easily be adapted to do more colourful birds. This could be achieved partly through texture mapping for gross colour, but also using procedural textures for patterning, as such an approach would work well in tandem with our generated feathers.

We did have some problems with this approach, most of which could be solved for future projects. The main difficulty was that the amount of groom data was very high – so much so that we had to write a caching system for the DSO. This slowed down the work flow. In retrospect, the blending of the groom filter attributes could have been carried out in the DSO, removing the need to store much more than a few attributes per feather. Future work would look at optimising this part of the process.