Artistic Rendering of Feathers for Animated Films

Feng Xie Wes Burian Andy Harbeck Ben Kwa *
Dreamworks Animation Inc

1 Introduction

Feathered creatures have become prevalent in our films since Kung Fu Panda, where we first developed a proprietary feather modeling, animation and rendering system for Crane. Since then, we have had Fifi the goose in Shrek Forever After, and now Lord Shen, the majestic white peacock in Kung Fu Panda 2. To enable maximum artist efficiency in different departments, we have found it's important to separate styling and rendering of feathers from modeling, grooming and lamination, so that different artists can iterate over their part of the feather system without requiring significant changes or rebuilds to other parts of the setup. In this talk, we focus on the feather plugins that enable our surfacing artists to style and vary the microstructure of individual feathers across different parts of a bird or across many birds, we also talk about some features developed to enhance the final rendering and lighting of feathers.

Figure 1: *Lord Shen at sunset*

2 Barb Generation and Styling

Each feather consists of 3 surfaces: a shaft and 2 vanes. Each vane is formed of thin hair-like barbs growing outwards from the shaft, tangential to the vane surface. The barbs vary in softness and density based on their location on the vane, soft and downy toward the root and firm and strong towards the tip. Individual feathers differ in their barb style based on the type of feather on each bird and vary also across different species of birds. Our feather plugin starts by generating straight barbs growing orthogonal to the shaft, tangent to the vane surface, then artists can style the straight barbs with these controls: density, width factor, split force, turn, curl, and bend, as illustrated in figure 2a. Texture (procedural or painted) maps can be used to control these factors. Variation between feather types can be achieved by changing the control maps or scaling them with noise or maps that vary across the skin of the birds. The feathers' surface details and barb styles are quite different between Fifi and Crane and across their different body parts yet they are all generated and rendered with the same geometry plugin and shaders.

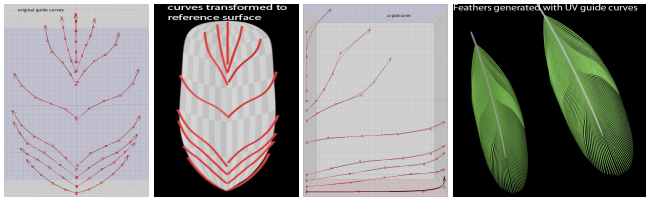
(a) *style controls* **(b)** *Fifi* **(c)** *Crane*

Figure 2: *Map controled feathers*

ISBN 978-1-4503-0921-9/11/0008.

3 Guide Curve Based Barbs

Control maps are powerful but they are not very intuitive. Furthermore, extreme distortions at vane tips can cause rendering or buzzing artifacts because the geometry can self intersect. To overcome these challenges, we developed guide curve based barb styling for Lord Shen. Curves are intuitive to use but modeling with curves needs to happen in reference space, prior to dynamics and deformation. This conflicts with our desire to separate barb styling from feather modeling and simulation. Our key insight is to transform curves modeled in reference space into uv curves which are not affected by deformation so we can apply styling curves to simulated feathers and across different types of feathers (which facilitates instancing and style sharing).



(a) *original* **(b)** *tranformed* **(c)** *uv curves* **(d)** *generated feathers*

Figure 3: *Guide Curve Based Feathers*

Algorithm: Transform guide curves onto the reference surface and use closest point search to find the uv of each control point to convert position guide curves to UV guide curves. To style a feather vane S using these guide curves, we generate a barb at each step v_j as a bezier strip bounded by two curves $C(v_j)$ and $C(v_j + dv_j)$ on S. A curve $C(v)$ on a vane surface S is computed by locating the guide curve interval $[v_i, v_k]$ that contains v, then linearly interpolating UV_i and UV_k to compute $UV(v)$ and evaluating S at $UV(v)$. dv_j is computed based on the density and curvature at v_j. To change the style of the feathers, one need only make simple edits to the original position guide curves and re-render.

4 Shading and Lighting

The map shader networks used to control the barb styles are also used to add shading and color variation to feathers, as shown by the images of Fifi, Crane and Lord Shen. Fire lighting was widely used in Kung Fu Panda 2 which made translucency crucial for Lord Shen's close ups. We added to the feather plugin: adaptive density controls and the ability to generate particles using two sided vane surfaces instead of thin hair-like barbs, which led to significant improvements in memory and computation efficiency for point based global illumination and translucency on feathers; and resulted in the pervasive use of translucency as a defining look for Lord Shen. By giving the artists a high degree of control and flexibility, the feather rendering system proved indespensible to the creation of feathered creatures, including the magnificent supervillain, Lord Shen.