EUROGRAPHICS 2002 / G. Drettakis and H.-P. Seidel
(Guest Editors)

Volume 21 (2002), Number 3

A Biologically-Parameterized Feather Model

L. Streit and W. Heidrich

{streit, heidrich}@cs.ubc.ca
Imager Computer Graphics Lab
The University of British Columbia, Vancouver, Canada

Abstract
Feathers, unlike other cutaneous appendages such as hair, fur, or scales have a definite structure. Variation in feather structure creates a wide range of resulting appearances. Collectively, feather structure determines the appearance of the feather coat, which can largely affect the resulting look of a feathered object (bird). In this paper we define the structure of individual feathers using a parameterization based on biological structure and substructures of actual feathers. We show that our parameterization can generate a large variety of feathers at multiple levels of detail and provide an initial step to semi-automatically generating a wide range of feather coats. This is achieved by specifying an intuitive interpolation between different structures and ages of feathers.

1. Introduction

Modelling of natural phenomena typically involves attempting to simulate very complex surfaces, structures, or processes. However, it is desirable to have control over a wide variety of phenomena with a limited set of parameters—a process known as *database amplification*.[23] Feathers have complex structure at many levels of detail, and the structure can vary widely across different types and ages of feathers. It is desirable to be able to create a wide range of results without having to model each substructure on every feather.

Feathers have similar scale and purpose as hair and fur. Various methods for modelling of hair and fur have been proposed recently. Some methods are based on creating the illusion of geometry through rendering, and others model a coarse view of the geometry. However, feathers are structurally more complex than hair strands: not only do they have components that are coarser than hair and thus are more apparent, but the variety of feather colours and patterns immediately reveals approximations in the underlying structure.

Structurally, feathers can be compared to plants in the context of branching patterns. Both have levels of branching substructures, however with plants the location and type of the branch varies much more than with feathers. Also, a collection of plants is not as specific an arrangement as feathers where a specific *look* to the feather coat is desired.

We propose a method of modelling feather coats by designing a small group of *key feathers* using a very limited number of parameters. Our model is based on the smooth aerodynamic nature of the surface of the feather. A Bezier curve of varying degree is used to approximate the curvature of the substructures in the feathers. Interpolation between the parameters and control points of the curve allows for smooth changes in the feather structure, and the tesselation of the Bezier curve allows us to control the level of detail.

2. Related Work

An extensive research area in computer graphics specializes in developing models of natural phenomena. Biological phenomena, a subclass of this area, have been modelled with both biologically and non-biologically based methods. The non-biologically based models typically use fractals, particle systems or stochastic systems and have been used for modelling plants[19, 23, 17] and terrain.[6] Since these models are complex, typically not interactive, and hence difficult to control, we have chosen to develop a feather model that is intuitive (based on feather structure) and interactive. We now review models of this nature.

Mimicking the development of the arrangement of biological parts (*biological patterns*) can be used to achieve *realistic-looking* models. Biologically-based models can be loosely categorized as structural or spatial models. Spatial models have generally been based on reaction-diffusion or cellular automata and have been fairly limited to constructing cellular size structures[24, 26] and plant shapes.[1, 8] Most structural models have focused on modelling botanical entities. These models have been based on L-systems to describe the development of the entity in an environment.[17, 16] Other structural models attempt to model *botanical rules* or geometric relationships.[4, 9, 13] Many models based on biological development involve simulation, restricting control.

Other work indirectly related to feather modelling includes biological models for phenomena that are similar in structure, scale, and complexity. Related work in modelling plants, hair and fur and feathers is presented below.

2.1. Plants

Plants and feathers are both branching structures. However, plants have more diverse branching types and locations than feathers. Plant branches have two degrees of freedom whereas the branching structures of feathers are very regular and always attached in the plane of the feather. Not only must plant models determine placement of organs, but plants are also generally larger in scale than feathers with a lower branching density, and thus have different geometric considerations. Also, in order to model an arrangement of plants, individual plants are modelled and placed together, whereas when modelling feathers, typically a specific feather coat pattern is desired. So, while some of the existing plant models are capable of modelling feathers (i.e. L-systems[17]), they are generally far more complicated than needed and they are not designed for specific feather attributes.

Models also exist for designing collections of plants by simulating the parameters of an entire ecosystem,[5] or by using instancing to design specific plant models and random perturbation of the model's parameters to achieve variation. This is less than desirable with feathers, since small variations in feather type and colour can affect the overall desired pattern of the feather coat. Recently a plant specific model was proposed that allowed for control over both the design of individual plants and the parameters used in simulation of an ecosystem.[18] The method we propose for feathers is similar in that it allows specific control of the feather structure and the design of the coat, without having to model each substructure in each feather of the coat.

2.2. Hair or Fur

Hair/fur are similar to feathers in scale and collectively form a coat. The strands are structurally simple and have consequently been modelled with polylines or NURBS curves,[10, 7] cylinders,[15] cylinder segments,[21, 3] and trigonal prisms.[25] Due to the structural simplicity, recent work has focused on rendering and dynamics of strands and design of hairstyles. Some methods model *key* or *guide* hairs[11] for which full dynamics and illumination are computed. The *in-between* hairs are interpolated variations of these attributes. The method we propose for modelling feathers uses the same notion of key attributes, but specific to feather modelling.

2.3. Feathers

The only existing feather model, and hence the closest to our work, was proposed by Dai et al.[2] for modelling the structure of feathers in Galliformes (a particular order of birds). Their model is based on the specification of an initial set of parameters; then evaluation of functions based on these initial parameters determines the curvature and location of the feather substructures, such as barbs (see Figure 1).

An interactively specified initial barb angle defines the direction of the first segment of the barb. Then with a user defined set of coefficients for a propagation function, the orientation of the next segment of the barb is computed from the function. An interpolation of the function coefficients defines the propagating orientation for all barbs. The barb length and distance between barbs is specified by a user defined function. There does not seem to be any way of specifying shaft length or curvature, and it appears they assume a closed, planar vane. They model the rachis as a generalized cylinder and the barbs as polylines and construct a triangular geometric representation of the vane. Since the propagation function is polygonal and the interpolation of the coefficients is continuous, there is no way to specify discontinuities in the feathers or to intuitively specify shape. They do not address interpolation of feather types and interpolation between the propagating functions is not possible.

Our parameters are similar to Dai et al.'s initial parameters without using a functional representation to model the feathers. With our model, rachis curvature, overall feather shape, and discontinuities in the vane are specified directly on the feather rather than by designing a function. Our method supports interpolation of feather shapes by design.

3. Ornithology Background

3.1. Feather Structure

A feather is a structure with a main center shaft and a hierarchy of fine branching structures extending from either side. The main shaft is called the *calamus* at the base, where there are no branching structures. The remaining portion is called the *rachis* (see Figure 1). Barbs branch from the main shaft, and extend angularly toward the tip of the feather. From the barbs branch *barbules* and from these possibly *barbicles*. Collectively, the barbs on one side of the rachis are called the *vane*. The calamus can be considered to branch into the rachis, and the *afterfeather* (a structure similar to the main feather). The feather appearance is defined by the number of levels of branching micro-structure.[14]

3.2. Feather Types

Feather taxonmies – based exclusively on feather structure, exist to classify the wide variety of feather sizes, shapes, and types. The most common type of feather is the *contour feather* (see Figure 1) which are found on the outer surface (contour) of the feather coat. The most defining feather

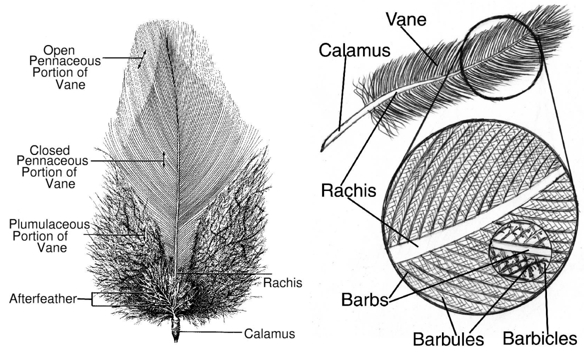

Figure 1: *Left: A Typical contour feather (from Lucas and Stettenheim,[14] p. 236). Right: Typical flight feather with branching substructures.*

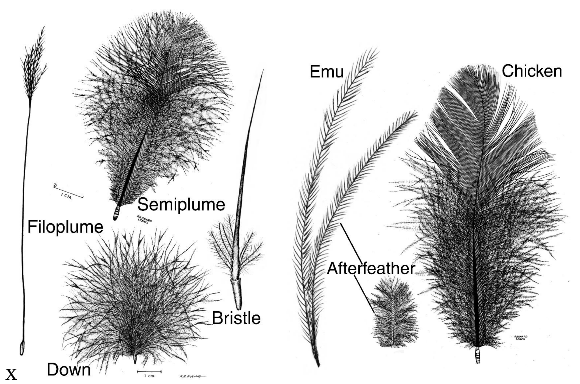

Figure 2: *Right: Examples of non-contour feathers (from Lucas and Stettenheim,[14] pp. 271,274,310,312). Left: Afterfeathers: Emu (attached) and Chicken (in center) (from Lucas and Stettenheim,[14] p.290)*

characteristic is type of vane—ranging from *plumulaceous* (fuzzy) to *pennaceous* (firm and stiff). The variation is due to the presence or absence of barbicles which aid in interlocking adjacent barbs. *Down* feathers (undercoat) and *semiplumes* (see Figure 2) are entirely plumulaceous, while *flight feathers* (wing and tail feathers—see Figure 1) are entirely pennaceous. Contour feathers are pennaceous at the tip and plumulaceous at the base. A second defining feather characteristic is the ratio between barb length and rachis length. All feather types except down feathers have a rachis longer than any barb. Specialized and less common feathers include *filoplumes* and *bristles*. Finally, the presence or absence of an *afterfeather* (see Figure 2) is another defining feather characteristic. Afterfeathers are always plumulaceous and resemble the main feather in shape but can range in size from nonexistent to the size of the main feather.

3.3. Feather Shape and Size

Feathers vary greatly in size and shape. Contour feathers can vary from a 1–2mm to 2m in length even on a single bird.

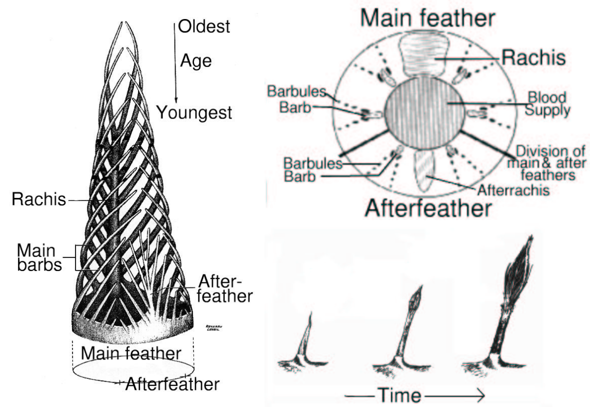

Figure 3: *Contour feather growth with cross-section (from Lucas and Stettenheim,[14] p. 200, 370).*

The curvature can vary both across and along the rachis and vane. The length and angle of the barbs can also vary, usually leading to ovate (egg-shaped) feathers, but may also result in obovate (upside-down egg), spatulate (short broad tips), or pointed. Overall curvature is almost always into the surface of the bird, but in very rare cases can curve away.

3.4. Feather Growth

Feathers are cyclically lost and regrown in a process called *moult*. They are usually lost/regrown sequentially as opposed to all at once. Feathers grow with the barbs curled inwards toward the underside of the feather (see Figure 3) and encased in a protective sheath. Cell division occurs only at the base of the feather making the tip always the oldest and consequently the first part to break out of the sheath.

4. Modelling Feathers

A feather model is needed that has an intuitive set of biologically-based parameters with both, flexibility to create a wide range of feather types and simplicity to generate thousands of specifically structured feathers. The proposed model is based on the notion of key representations at multiple levels of detail. An individual feather is designed using a set of parameters describing the length and curvature of the rachis, the length and angle of the barbs, and a set of key barbs for the vane curvature. The curvature of intermediate barbs is specified by interpolating the key barbs. These individual feathers are then used as key feathers in the design of the feather coat, by interpolating this parameterization to create intermediate feathers. The parameterization and interpolation of the parameters are discussed below.

4.1. Feather Parameterization

Our parameterization is based on the biological structure of the feather as outlined in Section 3. The two main structures we model are the shaft and the barbs. The after-feather is

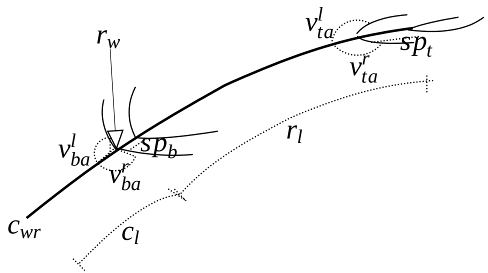

Figure 4: *Illustration of the parameters of our model.*

Structure	Parameter	Variable
calamus	length	c_l
	width-ratio	c_{wr}
rachis	length	r_l
	base width	r_w
	curvatures	r_{lc}, r_{vdc}
barb	length	l
	left vane angle	vl_{ba}, vl_{ta}
	right vane angle	vr_{ba}, vr_{ta}
	spacing	sp_b, sp_t
	# key barbs	m
	set of key barbs	$B^j, \{\forall j \vert 0..m\}$
$\forall$ Key barb	# control vertices	n
	control vertices	$cv_i, \{\forall i \vert 0..n\}$
	position on rachis	rp

Table 1: *List of parameters in our model*

modelled as a secondary feather attached to the main feather and has the same parameters as the main feather. The parameters we use are shown in Figure 4 and are found in Table 1. The calamus has length c_l and width-ratio c_{wr}. The rachis has length r_l, base width r_w, and two curvatures (side-to-side r_{lc}, and front-to-back r_{vdc}). The rachis tip is assumed to form a point and the calamus width is computed so that there is no discontinuity where the rachis and calamus meet.

The vane is modelled by a collective set of barbs. A select set of key barbs are specified and from these the rest of the barbs are generated by interpolation. Each key barb $B^j, j = 0,1,\ldots,m$ has a set of control points $cv_i, i = 0,1,\ldots,n$ and a location along the rachis $rp \in [0,1]$. The Euclidean distance between cv_0 and cv_n is clamped to $[0,1]$ and only the control vertices need to be stored, as the refined curve is computed at run-time. These remarks will be explained further in Section 4.2.1. A second barb parameter is the base angle *ba* and tip angle *ta* which define the angles of the barbs to the rachis at the rachis base and tip in the plane of the vane. There are two sets of these angles defining the left (vl_{ba}, vl_{ta}) and right (vr_{ba}, vr_{ta}) vane as shown in Figure 4. The barb angle is then linearly interpolated at run-time. A third barb parameter is the spacing along the rachis. There are both a base spacing sp_b and a tip spacing sp_t with a linear interpolation inbetween. The spacing is symmetric on both vanes.

The final parameter in determining the shape of the feather is the length of the barb. This can be constant or defined by a set of silhouette boundary points.

4.2. Interpolation

Our feather model uses interpolation at three levels of detail. Interpolation is used to generate hundreds of barbs on a feather in real-time while only storing the control vertices for a few key barbs. Interpolation is also used to generate a smooth transition between feather types and between feather ages. Finally, using these smooth transitions, a collection of feathers with realistic variations can be created from only a few key feathers. Below, the details of the interpolation to create the intermediate barbs and the variation in feather types and ages is presented.

4.2.1. Generating the barbs

When designing the feather, only key barbs are specified to define the curvature of the vane. Any barb branching from the rachis between key barb locations must be interpolated from the key barbs found immediately above and below the branch location. Thus, if B^a is the key barb found above and B^b is the key barb found below a barb b^k, then b^k would be a Bezier curve defined by the set of n control vertices $b^k_{cv_i}$:

$$\forall i \in \{0,1,...,n\} : b^k_{cv_i} = (1-t)\cdot B^a_{cv_i} + t\cdot B^b_{cv_i} \qquad (1)$$

where, $t = (b^k_{rp} - B^b_{rp})/(B^a_{rp} - B^b_{rp})$. This is a linear interpolation between each of the key barbs control vertices, where the interpolation parameter t is normalized to the distance between the two key barbs.

In the absence of a key barb below, a default key barb B^{df1} is used and is defined as four randomly generated control vertices. If there is no key barb defined above, a default key barb B^{df2} is used and is defined as a curve with slight front-to-back curvature. These two default barbs were chosen since many feather types have a plumulaceous base and pennaceous tip. If there are no key barbs defined at all, then B^{df1} is used giving the impression of an entirely plumulaceous feather (i.e. down feather).

Finally if the number of control vertices of the two barbs to be interpolated do not match, then the curve with the lesser number is simply elevated using the fact[20] that a Bezier curve of degree n with control vertices $cv_0, cv_1, \ldots, cv_n$ can be elevated to degree $n+1$ by fixing the end-points ($cv'_0 = cv_0$, and $cv'_{n+1} = cv_n$) and computing the rest as:

$$cv'_i = \frac{i}{n+1} cv_{i-1} + \left(1 - \frac{i}{n+1}\right) cv_i, \text{ for } i = 1,\ldots,n.$$

Once the barb control vertices are obtained, they are scaled by the length of the barb. With a Euclidean distance

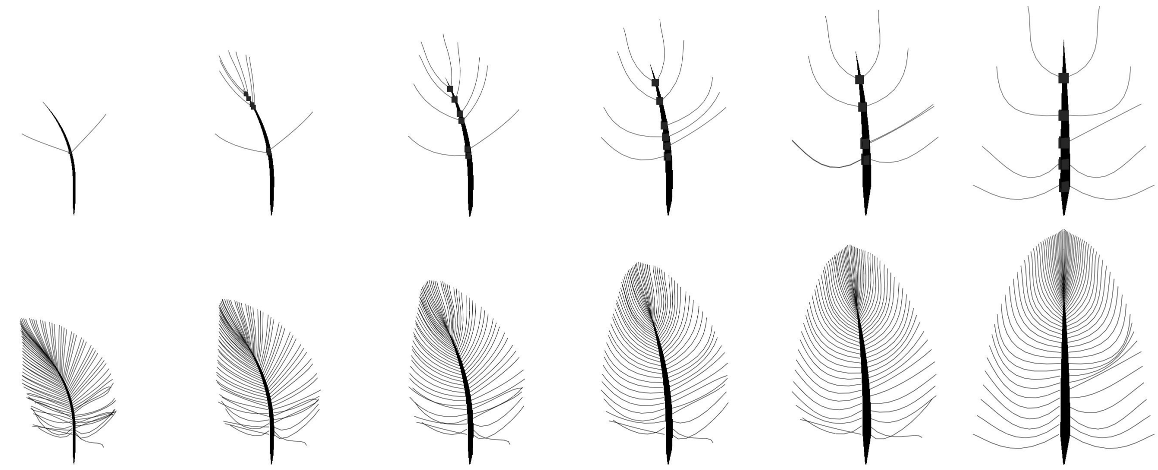

Figure 5: *Interpolation between leftmost and rightmost feathers showing key barbs (top) and feather structure (bottom).*

of 1 between cv_0 and cv_n the relative curvature of the vane is maintained through the scale, but when the distance is less than 1 the barbs scale non-linearly with respect to each other. This can be used to create discontinuities in the feather vane.

4.2.2. Acquiring the Parameters

A set of silhouette boundary points for determination of the barb length can be artificially constructed or determined from a flatbed scan of an actual feather. In fact, using image processing algorithms, not only can we extract barb length from the scan, but also the placement and control points of key barbs, barb spacing and barb angle. However, the front-to-back curvature will have to be added manually.

4.2.3. Modelling Feather Types

As outlined in Section 3, a wide variety of feather types exist. Taxonomies coarsely categorize feather types; however, generally there is a continuum between these categories. For instance, feathers can be found ranging continuously from down to semi-plume to contour. Also feather types in different regions are often blended where the regions join. Thus, it is desirable that our parameterization enables blending.

Our parameterization is designed to easily generate a continuum of feather types simply by linearly interpolating the rachis and calamus lengths and widths, the rachis curvatures, the barb angles and spacing, in addition to the key barb interpolation outlined in Section 4.2.1. The key barb interpolation across feathers is very similar to the interpolation used to generate the barbs within a single feather. First the key barbs are linearly interpolated across feathers then the remaining barbs are generated as before. When interpolating between two feathers with varying numbers of key barbs, the number of key barbs on the feather with fewer is artificially inflated until they are equal as shown in Figure 5.

4.2.4. Modelling Feather Growth

Due to the sequential loss/regrowth of feathers and the cyclical nature of moulting, a feather coat very rarely consists of entirely full developed feathers and in some moult stages only a few feathers are fully developed. Thus, it is important to model feathers at various developmental stages.

Feather growth can be modelled by interpolating the same parameters found in the previous section. However, instead of linearly interpolating all parameters simultaneously, their interpolation is staggered to simulate growth. The feather grows encased in a sheath until all substructures are nearly fully developed. The sheath is shed from the tip to the base. As the sheath sheds growth ceases in that region. From this information an interpolation schedule found in Figure 6 was constructed. The shaft length and width increase during the first 30% of the feather development. Next, the barbs lengthen to the midpoint of development. When the barbs start lengthening three extra key barbs are added to the feather. Two of these are straight barbs with an angle of 180°—one at the base of rachis and one at the tip. The third is a straight key barb with an angle of 180° that starts out located at the tip of the rachis and migrates to the base, interpolating its angle between the tip and base angles. Finally in the last 20% of the development, the extra barbs at the tip and base are removed and the rachis curvature and actual key barb angle and curvature increase. An example of some of these feather growth stages are shown in Figure 7, and in the video sequences.

4.3. Implementation

Once the feather structure is specified we must generate the geometry at a particular level of detail and then possibly add a texture map. The barbs are alpha blended to simulate the presence of barbules. OpenGL was used for rendering.

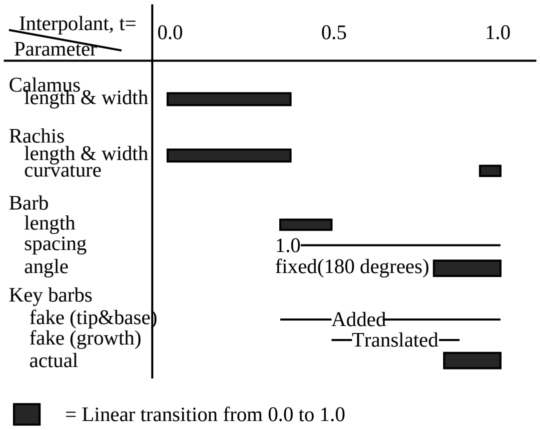

Figure 6: *Sequencing of parameter interpolation for growth.*

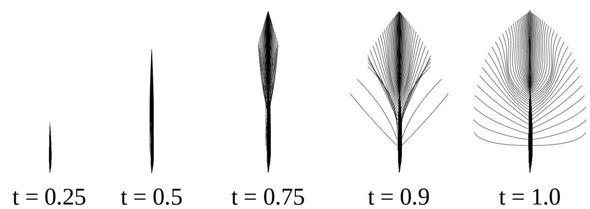

Figure 7: *Simulated feather growth from interpolation sequence in Figure 6. Time steps at 25%, 50%, 75%, 90% and 100% of development.*

4.3.1. Geometry and Alpha Blending

The feather is drawn using a generalized cylinder for the geometry of the shaft, and polylines or triangle strips for the geometry of the barbs. The rachis is also a Bezier curve scaled by the rachis length. The four control vertices used for the curve are defined by the two curvatures r_{lc} and r_{vdc} $\in [0..1]$. The control vertices are then computed as follows:

$$\begin{array}{lcccc} cv_0 & = & 0.0 & 0.0 & c_l \\ cv_1 & = & r_{vdc} \cdot r_l & r_{lc} \cdot r_l & c_l + r_l \cdot 1.0/3.0 \\ cv_2 & = & r_{vdc} \cdot r_l & r_{lc} \cdot r_l & c_l + r_l \cdot 2.0/3.0 \\ cv_3 & = & 0.0 & 0.0 & c_l + r_l \end{array}$$

The number and location of barbs along the rachis are computed as a function of rachis length r_l and the barb spacing sp_b, sp_t. A barb b^j at each location rp is generated using the interpolation between key barbs (Section 4.2.1). The barb b^j is then rotated by the angle to the rachis about the vector perpendicular to the plane such that the vector from b_{cv_0} to $b_{cv_{n/2}}$ lies within the plane of the vane. This rotation ensures continuity across the rachis.

Randomly generated control vertices are used for key barb B^{df1} to simulate a plumulaceous vane. In order to have frame to frame consistency, a table of random vertices is precomputed and reused in every frame.

Once the barb is generated as specified in Section 4.2.1, it is drawn either as a polyline or two triangle strips. The two triangle strips are drawn on either side of the polyline with constant width in the plane of the vane. They are alpha blended toward the tip of the barb and away from the center similar to Koh and Huang's[12] approach for hair. In order to ensure proper blending, not only is the shaft and barb drawing order important, but the drawing order of the left v.s. right vane is dependent on the camera location. The barb triangle strips are drawn from barb tip inwards for the vane closest to the camera and in the opposite direction for the vane further away.

4.3.2. Tessellation

Depending on the desired level of detail the amount of geometry can be adjusted by adjusting the shaft and barb tessellations. The same technique for elevating the degree of the Bezier curve specified in Section 4.2.1 can be used to generate the appropriate refinement of the curve. This approach is used to adjust the tessellation of both the rachis and barbs curves. The tessellation of the generalized cylinder for the rachis is also varied. The new level of detail is never stored, but simply recomputed for rendering. Figure 8 shows a feather with low and high tessellation.

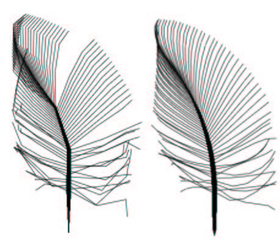

Figure 8: *Left: Four cvs per barb/rachis and triangle cross-section for shaft. Right: Much higher tessellation.*

4.3.3. Texture

The barbs may be texture mapped by generating texture coordinates ignoring rachis curvature to make the texture map generation simple and reusable. The texture map can be artificially generated as shown in Figure 9 or taken from the scan of an actual feather as shown in Figure 14. The texture can be specified in three ways: on the barb, on the vane, or at the base of the rachis during feather growth. If the texture coordinates are generated on the barb, the *u*-coordinate is specified relative to the length of the barb and the *v*-coordinate is the attachment point of the barb to the rachis. Regardless of changes in barb angle or curvature the texture remains attached to the barb. This is useful in creating different feather patterns given the locations of pigment along the barb.

If the texture coordinates are generated on the vane, they are relative to the projected location of each control vertex on each barb. Consequently as the barb curvature or angle to rachis change, the texture swims on top of the barbs. This method is useful for generating feathers with particular markings in specific locations, while still being able to alter the barb structure.

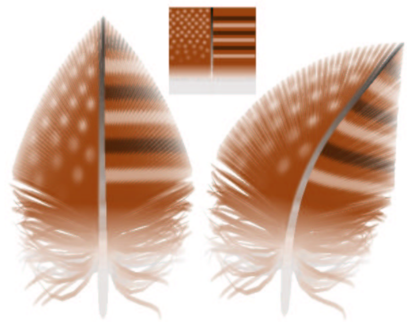

Figure 9: *Feathers texture mapped with middle texture.*

The third texture coordinate generation method is relative to growth. The *v*-coordinate is varied with time while the *u*-coordinate is mapped to the circular cross-section of the emerging feather (see Figure 3). This method can be used to approximate the natural of pattern formation of feathers, but is not biologically precise since pigments can migrate slightly after deposition.

4.4. Modelling the Feather Coat

Generating a collection of feathers from our parameterization is natural. The feathers within the convex hull of the *key feathers* are automatically generated using bilinear interpolation to blend the closest key feathers. Finally, since alpha blending is used for rendering the feathers, they must be drawn from the bottom feather to the top.

5. Results

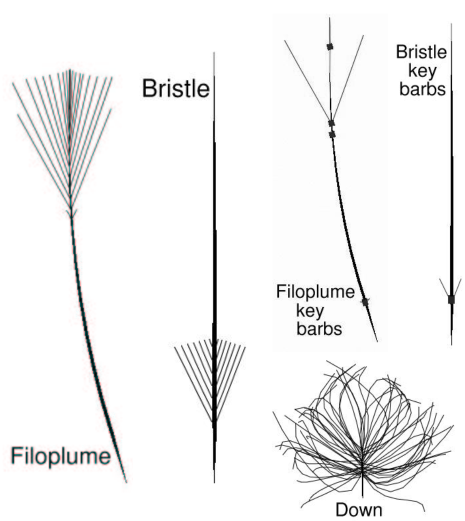

Figure 10: *Bottom: Feather structure generated with our parameterization. Top Right: key barbs used for feather generation (none for down). Barb length was specified with a silhouette shape. Compare these with the feathers in Figure 2.*

Figure 11: *Semi-plume(black) with afterfeather(blue) Left: Polyline version. Center and Right: Triangle strips version. The center feather has less barbs for illustration.*

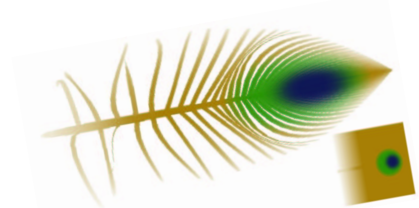

Figure 12: *Generated Peacock feather.*

Using the relationship between barb description and barb length, shape discontinuities in the vane can be created (although are rare) using a silhouette shape or by placing two key barbs directly adjacent to each other to drastically alter the length of the barb. In Figure 10 the filoplume's sharp discontinuity was created by specifying key barbs that had all four control vertices at the same location and using a square shape silhouette to tailor the length for the top barbs. Alternately the bristle uses silhouette information to form the sharp discontinuity in the vanes.

Figure 11 shows a generated semi-plume with an afterfeather. Generated shape data was used for barb length. One straight key barb was positioned at the rachis tip. The right two feathers' barbs are drawn with triangle strips; the main feather in blue and the after feather in black. Figure 12 shows a generated peacock feather, using the texture shown and three sets of key barbs: one straight set at the tip of the rachis, another below the *eye* and a third curved set at the base of the *eye*. Figure 13 shows the textured growth sequence of the feather from Figure 8.

Figure 14 shows the comparison of our generated feathers

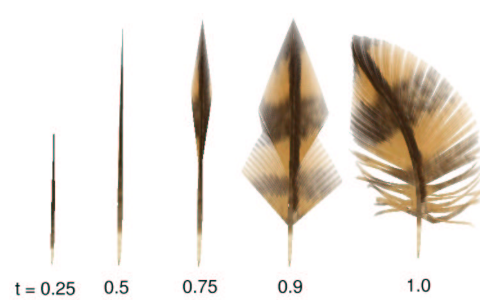

Figure 13: *Growth sequence of feather from Figure 8.*

Figure 14: *Left: Generated Hawk feather. Center: Flatbed scan of an actual Hawk feather. Right top: Generated Budgie feather. Right bottom: Flatbed scan of actual Budgie feather. Left/center use barb length from scan. Right uses barb length from shape data shown. Both textures are from the scans.*

on the left (top) with a flatbed scan of the real feather in the center (bottom). The feather on the left uses barb length extracted from the center image. The feather on the right uses barb length specified by generated shape data. Both figures use textures from the scans. The only adjustment was to the rachis placement and to resample the texture (power of two).

Figures 15 and 16 show the use of feather interpolation for creating a feather coat. The two specified feather types are shown on the bottom left and right. In each of these figures key feathers were specified to form the convex hull of the follicles. The rest of the feathers were automatically generated. All the feathers have default constant orientation to the surface. Figure 15 uses a regular hexagonal follicle distribution, where Figure 16 has a pseudo-random distribution.

Finally, individual feathers can be computed and rendered in real time using basic OpenGL rendering. The collection of feathers in Figure 15 with 64 feathers can also be computed in real time, though rendering drops to approx. 15 fps on a PIII 800 GeForce2 PC due to the thousands of barbs, each containing numerous polygons.

6. Conclusions and Future Work

In this paper we have presented a novel parameterization for modelling feathers based on the notion of key attributes at various levels of detail. The two most important attributes of our parameterization are the *key feathers* and *key barbs*. Our parameterization can create a wide variety of feather types and naturally interpolates between ages, types and levels of detail to quickly and easily generate a collection of feathers. Finally, the interpolation allows for automatic generation of spatially varying feather types over a surface.

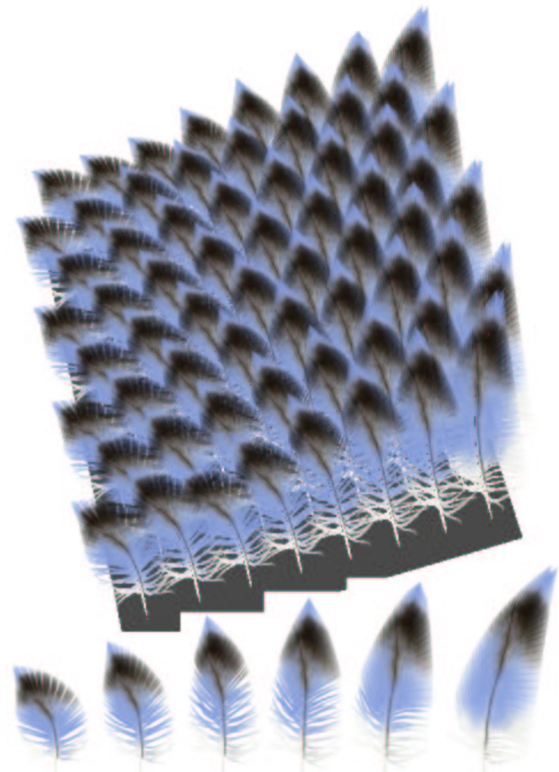

Figure 15: *Top: Bilinear interpolation of four key feathers generated from scans of Budgie feathers. Bottom: an interpolation between two of the key feathers.*

The model does not allow for specification of more than four control vertices for the rachis and does not enable direct control over the placement of these vertices. This could be easily adapted to a method similar to the barb specification.

Our model uses only basic OpenGL rendering and leaves a lot of room for incorporating more sophisticated rendering techniques such as [22]. In the future we would like to work on rendering issues such as self-shadowing, use of BRDFs for distant views, illumination attributes of feathers as well as possibly exploring non-photorealistic rendering styles similar to Figures 1 to 3.

Our model also does not address many of the complications that arise in modelling a feather coat. In the future we would like to explore issues such as follicle distribution, feather orientation, dynamics and animation.

Acknowledgements

Thank you to members of Imager for support and inspiration, particularly Alain and Roger. Thank you to NSERC for funding and to Wes for understanding so much insanity.

References

1. Peter Chambers and Alyn Rockwood. Visualization of solid reaction-diffusion systems. *IEEE Computer Graphics and Ap-*

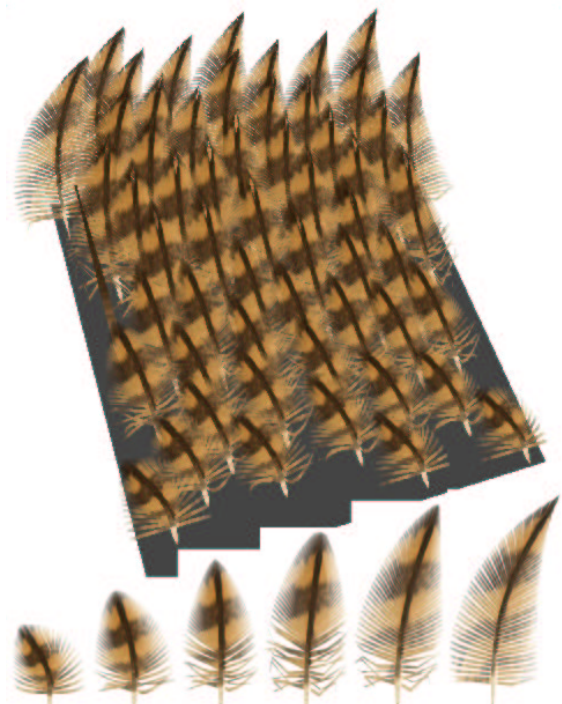

Figure 16: *Bilinear interpolation of four key feathers generated from scans of Hawk feathers.*

plications, pages 7–11, 1995.

2. Wen-Kai Dai, Zen-Chung Shih, and Ruei-Chuan Chang. Synthesizing feather textures in galliformes. *Computer Graphics Forum*, 14(3):407–420, August 1995.

3. Agnes Daldegan, Nadia Magnenat Thalmann, Tsuneya Kurihara, and Daniel Thalmann. An integrated system for modeling, animating and rendering hair. *Computer Graphics Forum (Eurographics '93)*, 12(3):211–221, 1993.

4. Phillippe de Reffye, Claude Edelin, Jean Francon, Marc Jaeger, and Claude Puech. Plant models faithful to botanical structure and development. *Computer Graphics (Proceedings of SIGGRAPH 88)*, 22(4), August 1988.

5. Oliver Deussen, Patrick Hanrahan, Bernd Lintermann, Radomír Mech, Matt Pharr, and Przemyslaw Prusinkiewicz. Realistic modeling and rendering of plant ecosystems. *Proceedings of SIGGRAPH 98*, pages 275–286, July 1998.

6. A. Fournier, D. Fussell, and L. Carpenter. Computer rendering of stochastic models. *Communications of the ACM*, 25(6):371–384, June 1982.

7. Allen Van Gelder and Jane Wilhelms. An interactive fur modeling technique. *Graphics Interface '97*, pages 181–188, May 1997.

8. Ned Greene. Voxel space automata: Modeling with stochastic growth processes in voxel space. *Computer Graphics (Proceedings of SIGGRAPH 89)*, 23(3):175–184, July 1989.

9. M. Holton. Strands, gravity, and botanical tree imagery. *Computer Graphics Forum*, 13(1):57–67, January 1994.

10. Ken ichi Anjyo, Yoshiaki Usami, and Tsuneya Kurihara. A simple method for extracting the natural beauty of hair. *Computer Graphics (Proceedings of SIGGRAPH 92)*, 26(2):111–120, July 1992.

11. A. Iones, A. Krupkin, S. Volodarsky, and S. Zhukov. Fur and hair: practical modeling and rendering techniques. *The Proceedings of the IEEE International Conference on Information Visualization (IV'00)*, 2000.

12. C.K. Koh and Z. Huang. Real-time animation of human hair modeled in strips. *Eurographics Computer Animation and Simulation Workshop*, 2000.

13. Bernd Lintermann and Oliver Deussen. Interactive modelling and animation of branching botanical structures. *Eurographics Computer Animation and Simulation Workshop*, 1996.

14. Alfred M. Lucas and Peter R. Stettenheim. *Avian Anatomy Integument*. U.S. Goverment Printing Office, Washington D.C., 1972.

15. Gavin S. P. Miller. From wire-frames to furry animals. *Graphics Interface '88*, pages 138–145, June 1988.

16. Przemyslaw Prusinkiewicz, Mark James, and Radomír Mech. Synthetic topiary. *Proceedings of SIGGRAPH 94*, pages 351–358, July 1994.

17. Przemyslaw Prusinkiewicz and Aristid Lindenmayer. *The Algorithmic Beauty of Plants*. Springer-Verlag, 1990.

18. Przemyslaw Prusinkiewicz, Lars Mundermann, Radoslaw Karwowski, and Brendan Lane. The use of positional information in the modeling of plants. *Proceedings of SIGGRAPH 2001*, pages 289–300, August 2001.

19. William T. Reeves and Ricki Blau. Approximate and probabilistic algorithms for shading and rendering structured particle systems. *Computer Graphics (Proceedings of SIGGRAPH 85)*, 19(3):313–322, July 1985.

20. Alyn Rockwood and Peter Chambers. *Interactive Curves and Surfaces*. Morgan Kaufmann Publishers Inc., California, 1996.

21. Robert Rosenblum, Wayne Carlson, and Edwin Tripp. Simulating the structure and dynamics of human hair: Modelling, rendering and animation. *The Journal of Visualization and Computer Animation*, 2(4):141–148, 1991.

22. Morgan Schramm, Jay Gondek, and Gary Meyer. Light scattering simulations using complex subsurface models. *Graphics Interface '97*, pages 56–67, May 1997. ISBN 0-9695338-6-1 ISSN 0713-5424.

23. Alvy Ray Smith. Plants, fractals and formal languages. *Computer Graphics (Proceedings of SIGGRAPH 84)*, 18(3):1–10, July 1984.

24. Greg Turk. Generating textures for arbitrary surfaces using reaction-diffusion. *Computer Graphics (Proceedings of SIGGRAPH 91)*, 25(4):289–298, July 1991.

25. Yasuhiko Watanabe and Yasuhito Suenaga. A trigonal prism-based method for hair image generation. *IEEE Computer Graphics & Applications*, 12(1):47–53, January 1992.

26. Andrew Witkin and Michael Kass. Reaction-diffusion textures. *Computer Graphics (Proceedings of SIGGRAPH 91)*, 25(4):299–308, July 1991.