

Modelling of Feather Coat Morphogenesis for Computer Graphics

by

Lisa Marie Streit

B.Sc., The University of Alberta, 1996

M.Sc., The University of Alberta, 1998

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

THE FACULTY OF GRADUATE STUDIES

(Department of Computer Science)

We accept this thesis as conforming
to the required standard



The University of British Columbia

November 2003

© Lisa Marie Streit, 2003

Abstract

This thesis addresses the problem of modelling realistic looking feather coats and their morphogenesis over the lifespan of the bird, for computer graphics. Modelling of feather coats is interesting aesthetically simply because of the prominence of birds in our daily 'lives', as well as the texture, pattern and shape diversities of feathers and feather coats. From a computer graphics point of view it becomes an interesting problem due to the sheer detail and complexity of the model that must be managed in a changing environment. Not only is it desirable to simulate changes in the feather coat, but there is also a need for the feathers to respond appropriately when the skin surface moves or is animated.

In this thesis three aspects which adversely affect the feather coat's appearance are explored and modelled: feather structure, follicle distribution and feather arrangement. Each aspect takes its basis from attributes of real feather coats, and the results created are compared with real feather coats. In addition to the static creation of feather coats, their change over time (morphogenesis) is also explored and shown to have an adverse affect on the resulting appearance of the coat. The results of feather moulting are included in the morphogenesis and the methodology is both based on real feather coats and the results compared with real feather coats.

The focus of this thesis is on modelling the structure of the feather coat, rather than rendering. However, since an adequate rendering methodology is required to assess to results of the structural model a rendering methodology is also investigated. Synthesis of the patterns in feathers is also an interesting problem and is an entire area of research in areas such as Chemistry, Mathematics and Mathematical Biology [Har93]. Since pattern synthesis in feathers is currently an active area of research in these fields, it is deemed outside the scope of this thesis. In order to preserve the realism of our results, acquired patterns are used in this work.

Contents

Abstract	i
Contents	iii
List of Tables	vii
List of Figures	ix
Dedication	xiii
Acknowledgements	xv
1 Introduction	1
1.1 Why Feathers?	2
1.2 Why Morphogenesis?	3
1.3 Modelling Natural Phenomena	4
1.4 Problem Scope and Goals	5
1.5 Overview of Thesis	7
2 Feather Morphology	9
2.1 Introduction	9
2.2 General Anatomical and Avian Terminology	9
2.3 Individual Feather Structure	10
2.3.1 The Contour Feather	11
2.3.2 Flight Feathers	14
2.3.3 Down Feathers	14
2.3.4 Contour Feather Variations, Filoplumes and Bristles	14
2.3.5 Feather Shape and Curvature	15
2.4 Feather Coats	16
2.4.1 Follicle Distribution	16
2.4.2 Feather Orientation and Shape	20
2.4.3 Number and Size of Contour Feathers	21
2.5 Feather Patterns	21

2.5.1	Pigments or Biochromes	21
2.5.2	Structural Colours	22
2.5.3	Colour Variation and Changes	23
2.5.4	Pattern and Pigment Location	24
2.6	Morphogenesis	24
2.6.1	Feather Growth	25
2.6.2	Moulting	26
2.7	Summary	27
3	Computer Graphics and Feathers	29
3.1	The Natural Phenomena Challenge	29
3.2	Level of Detail Modelling	30
3.2.1	Texture Maps	31
3.2.2	Texture Mapping Problems	32
3.2.3	Reflection Models	33
3.3	Mathematical-Biological Models	35
3.4	Relevant Computer Graphics Models	38
3.4.1	Plant Models	39
3.4.2	Hair and Fur Models	40
3.5	Feathers	42
3.5.1	Commercial Productions	42
3.5.2	Dai et al.	42
3.5.3	Non-photorealistic Feathers	43
3.5.4	Concurrent Photorealistic Feathers	44
3.6	Summary	47
4	Problem Scope, Goals and Approach	49
4.1	Model Expectations and Goals	49
4.2	Assumptions and Limitations	51
4.2.1	Level of Detail	51
4.2.2	Object-Specific Models	52
4.2.3	Emergence, Control and Efficiency	52
4.2.4	Input and Output of the System	53
4.2.5	Other Necessary Model Attributes	53
4.3	Problem Segmentation	54
4.3.1	Feather Structure	54
4.3.2	Feather Arrangement	55
4.3.3	Feather Alterations	56
4.4	Main Contributions	56
4.5	Summary	57

5	Modelling Feather Structure	59
5.1	Basis for Feather parameterization	59
5.2	Our Parameterization of Individual Feathers	61
5.2.1	Interpolation	63
5.2.2	Interpolation Within a Feather - Barb Generation	63
5.2.3	Acquiring the Parameters	65
5.2.4	Feather Types and Interpolations	66
5.2.5	Modelling Feather Growth and Change	69
5.3	Surface Generation	70
5.4	Feather Colouration	72
5.5	Summary	73
6	Modelling Feather Arrangement	75
6.1	Step One: Follicle Distribution	77
6.1.1	Lloyd's Algorithm	79
6.1.2	Delaunay Triangulation On a Surface	80
6.1.3	Applying Lloyd's Algorithm	81
6.1.4	Density Biasing	83
6.1.5	Manual Follicle Placement	83
6.2	Step Two: Feather Generation	84
6.2.1	Imposed Parameterization	84
6.3	Step Three: Feather Direction and Orientation	85
6.3.1	Surface Directions and Coordinate Frame	88
6.3.2	Shaft Rotation	88
6.3.3	Barb Elevation	89
6.3.4	Packing Summary and Result	91
6.4	Analysis of the Algorithm	92
6.5	Summary	92
7	Level of Detail and Rendering	95
7.1	Feather Rendering	95
7.1.1	Textures	96
7.1.2	Alpha Blending	98
7.1.3	Levels of Detail and Reflection Model	98
7.1.4	Other Levels of Detail	102
7.2	Rendering Feather Coats	103
7.3	Alternative Rendering Styles	103
7.4	Summary	103

8	Simulating Feather Alteration	105
8.1	Feather Growth	105
8.2	Feather Replacement	107
8.3	Feather Rearrangement	111
8.4	Summary	115
9	Results and Analysis	117
9.1	Components of the Model	117
9.1.1	Feather Structure Component	118
9.1.2	Feather Arrangement	126
9.1.3	Alteration Component	136
9.2	Discussion and Analysis	138
9.2.1	Integration Challenges	138
9.2.2	Rendering and LOD Challenges	141
9.2.3	Comparison with Other Models	142
9.2.4	Contribution and Validation	143
9.3	Summary	144
10	Conclusions and Future Work	145
10.1	Main Contribution	148
10.2	Future Work	148
10.3	Conclusions	149
10.3.1	Supplemental Material	149
	Glossary	151
	Bibliography	154
	Non-Computer Science References	154
	Computer Science References	156
	Index	165

List of Tables

5.1 Our Model Parameters	61
------------------------------------	----

List of Figures

1.1	Feather Coat Morphogenesis of the Rose-breasted Grosbeak	3
1.2	Feather Coat Morphogenesis of the Herring Gull	8
2.1	Anatomical Terms	10
2.2	Typical Contour Feather	10
2.3	Flight Feathers	11
2.4	Other Feathers	11
2.5	Afterfeathers	11
2.6	Plumulaceous and Pennaceous Barbules	13
2.7	Main Feather Tracts	17
2.8	Primaries, Secondaries and Wing Coverts	19
2.9	Feather Wear and Colouration	23
2.10	Typical Contour Feather	23
2.11	Illustration of Feather Development	25
2.12	Plumage Succession and Moulting	27
3.1	Microsoft Research Asia Feather Images	46
4.1	Breakdown of Feather Coat Morphogenesis Problem	55
5.1	Diagram of the Model Parameters	62
5.2	Parameter Variation	62
5.3	Examples Using Key Barbs	65
5.4	Examples Using Silhouette Data	66
5.5	Examples of Feather Types	67
5.6	Matching of Key Barbs	67
5.7	Feather Interpolation	68
5.8	Level of Detail Example	71
6.1	Feather Coat Modelling Process	76
6.2	Feather Tract Division	78
6.3	Lloyd's Algorithm in 2D	79
6.4	Lloyd's Algorithm limitations on surface	79

LIST OF FIGURES

6.5	2D Delaunay Triangulation in 3D	80
6.6	Distribution Results	82
6.7	“Hi” Bias Distribution	83
6.8	Follicle Distribution	83
6.9	Key Feather Interpolation	85
6.10	Surface Direction Vectors	85
6.11	Data Point Determination for Shaft Rotation	89
6.12	Packing Planes	90
6.13	Results of Packing Algorithm	91
7.1	Texturing Examples	96
7.2	Alpha Blending	97
7.3	Anisotropic Reflection Model	99
7.4	Multi-texturing Example	102
7.5	Alternative Rendering Styles	104
8.1	Sequencing of Parameters for Growth	106
8.2	Feather Growth Example	106
8.3	Sequential Growth Replacement	108
8.4	Random Growth Replacement	109
8.5	Radial Growth Replacement	110
8.6	Bending of Wing	112
8.7	Styling Using Fixed Angles	113
8.8	Styling Using Surface Directions	114
9.1	Budgie Semiplumes and Coverts	119
9.2	Afterfeather	120
9.3	Budgie Afterfeather	120
9.4	Hawk and Budgie Feathers	120
9.5	Synthesized Peacock Feather	122
9.6	Synthesized Pheasant Feather	122
9.7	Feather Growth	123
9.8	More Feather Growth	123
9.9	Multi-Resolution Feathers	124
9.10	Follicle Distribution	127
9.11	Follicle Distribution vs Pattern	128
9.12	Packing on a Sphere	130
9.13	Packing on a Hi-Resolution Sphere	131
9.14	Multi-texturing LOD	131
9.15	Budgie Pattern Comparison	131
9.16	Space in the Packing	134
9.17	Interpenetrations with Packing	134

9.18 Parrot Moults 136
9.19 Feathered Wing 138
9.20 Feathered Bird 139
9.21 Microsoft Research Asia Feather Images 142

To *Dr. Alain Fournier*

to whom I will always be *your* student.

May your curiosity, perseverance and passion for the truth live on in *your* and all students.

and

To *Arthur Desorcy*

Who taught me that many discoveries can be made in a workshop if you try.

Acknowledgements

Ahhh... the acknowledgements. So many people helped over the years in so many ways. Where to begin?

The Imager lab - where I spent so many hours, with-standing the mockery of so many (Rob, Jimmy, Bill G., David B., Ben, Ritchie, Dave B.) nudging me ever so gently to the brink of insanity, before yanking me back for another run ;) The lab would have been lifeless without you. But most importantly, I thank Roger Tam and Rob Scharein - good friends and colleagues. Roger, together we have been through the thick and thin of it all. You have taught me (I am still practising) patience, humility, perseverance and pride. Rob, as a friend and mentor you have always been there to offer good, sound, truthful advice. Without you two, I am not sure I would be here. Thanks also to my initial mentors Dave M. (always buddies), Alex, Dima, Paul F., Oleg, Mario, John and of course Alain who encouraged me quite unwittingly to start this crazy trip - thanks - I think :). And those whom I began the journey with: Paul K., Bettina, Alex, Dima - your progress has always been an inspiration. Thanks also to those who bridged the gap: Kelly - your undaunting support of the lab, even through my demands and Wolfgang - for stepping in to fill large shoes and providing an alternate perspective and a fresh viewpoint. And thanks for feed back from my committee members - Drs. Kim Cheng, Wolfgang Heidrich, Michiel van de Panne, my university examiners - Drs. Ron Rensink and Sidney Fels and my external examiner - Dr. Oliver Deussen.

I must also thank my family for being ever so patient and supportive without question. Understanding why was never a question I could answer completely, but you still believed in me and offered comical relief and advice in tough situations (Michael, Sandy, Valerie) and words of encouragement when times got rough (Mom, Dad). I could not have completed this without your support. Michael and Valerie - watching you grow and succeed in life has been a motivation for me. You will never know how much you have truly helped. I hope one day I may be there for you as you have for me. And to my extended family - Ron, Kathy (fellow Sagg.) and William. Seeing you persevere through so much has truly been inspirational. And through it all you still managed to send a joke or a reminder of Alberta at just the right moment. And to Mary, Rudi, Marie and Ginger - I only hope to be half as great as any of you!

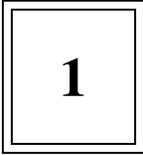
And last but never least - Wes - my best friend, my boy friend and most recently my husband. You have always encouraged me and helped set me straight when the road seemed impassable. You never let me get completely lost - continually reminding me there is life beyond my degree. You have been with me through my entire University education and I know we will continue to learn and grow together for many years. Your entrepreneurship is both head spinning and motivational. Thanks for not letting life pass me by.

And of course, thanks to my fine feathered friends - Chippy, Taz, Trouble, the finch family and all those passed on - for original inspiration. Finally, thanks to funding agencies: NSERC, ASI, UBC and the MumandDad fund.

Thank you all, I am ever so grateful.

LISA MARIE STREIT

*The University of British Columbia
November 2003*



Introduction

Of all of earth's creatures birds have the most variable appearance - largely due to their feather coat. Mammals with their fur and hair have a coat of similar complexity and scale, but lack the variation in natural pigments needed to create widely varying colours and patterns. Tropical fish and butterflies have similar pigment spectra, but lack the ability to drastically alter the orientation and positioning of the scales to achieve wide variations in shape and size. The variability of feather coats is not only diverse across species of birds but also over time. A bird's appearance can change drastically over a span of a year and more so over the lifetime of the bird. With this variability in shape, colour and texture within and across species of birds the simulation of changes in feather coats is an interesting and challenging task.

A long addressed problem in computer graphics is the desire for visual realism. In the realm of nature this generally means addressing complexity. A sub-area of computer graphics called natural phenomena has targeted realism of nature including modelling of complex shapes, patterns, processes and dynamics, and rendering of complex lighting situations and materials. Often this complexity is targeted by devising a specific model for a particular class of objects.

Birds are interesting creatures and from a computer graphics/simulation viewpoint, there are many interesting problems that can be addressed including, simulating flight, feather structure, feather patterns, feather illumination and growth, bird growth, modelling of beak, feet eyes and movement/behaviour. This thesis concentrates on development of the feather coat, which involves modelling feather structure, placement and growth as well as the collective arrangement and replacement of feathers.

This chapter will start by first motivating why there is a desire to model feathers and feather coat development (morphogenesis). Next, the principles considered in creating the feather coat morphogene-

sis methodology are outlined along with the scope of both the problem and model. Finally, this chapter concludes with an outline of the rest of the thesis.

1.1 Why Feathers?

“Feathers are probably the most complex derivatives of the integument to be found in a vertebrate animal. Their many parts display enormous variety of modifications”. (Lucas and Stettenheim [LS72a, p. 235])

Feathers are interesting to study for three reasons: they are prevalent in our world, they are biologically interesting, and little work has been done in this area of computer graphics. Birds are a part of our everyday lives and as such simulating feathers and hence birds would enhance the realism of computer graphics scenes and environments. As well feathers have been used in various man-made items for fashion, writing and decoration. Biologically, feathers are similar to hair in that they are both cutaneous appendages¹, have similar distribution densities and scale, and both greatly affect the resulting appearance of the object. However, feathers are structurally more complex and diverse than hair, thus resulting in a wide variety of feather types, since structurally hairs vary only in thickness and curliness. Also, feather colouration is more diverse than the limited variation of melanin of mammalian hair, resulting in many unique feather and feather coat patterns.

From a computer graphics point of view, feathers are comparable in complexity and scale to hair and plants. There has been a lot of work in both of these areas of modelling. Hair has typically been modelled as individual strands or groups of strands (lofts) using polylines [iAUK92], cylindrical segments [RCT91] and trigonal prisms [WS92] with hinges and mass-spring systems for dynamics. Often hair-to-hair collisions have been ignored [DTKT93] since specific ordering between strands is usually not noticeable. This is not the case with feathers, as overlapping feathers form a specific pattern and altering the order is immediately noticeable. Orientation in feathers is also very critical, where hair orientation is negligible. The dynamics of hair tend to be more complex, since hair is generally more flexible, however it is uniformly flexible where feathers have variation in flexibility within a single feather.

Plants are structurally similar to feathers, since they both have basic branching structures. Modelling of large collections of plants is comparable to modelling feather coats, both in quantity of substructures and

¹appendages attached to the cutaneous layer of the skin

the development or morphogenesis of the structures. However, the arrangement and orientation of individual plants in an ecosystem does not have to be completely accurate, since they do not form any specific collective patterns. The branching structures of plants have typically been modelled using L-systems [PL90], however models made of parameters specific to their structure have also been proposed [DHL⁺98].

Not only are feathers interesting to model but the notion of modelling a changing feather coat is intriguing due to the complexity of this problem. Although the structure of a single feather is diverse and intricate, the specific structure and arrangement of the thousands of feathers a bird may possess further complicates the problem. Adding change to the feather coat and animation of the bird additionally complicates the approach taken, since not only must the feathers react with each other properly due to movement of the bird's skin, but they have to interact naturally to simulate the development of the feather coat. Managing this complexity is both the challenge and contribution of this work.

1.2 Why Morphogenesis?

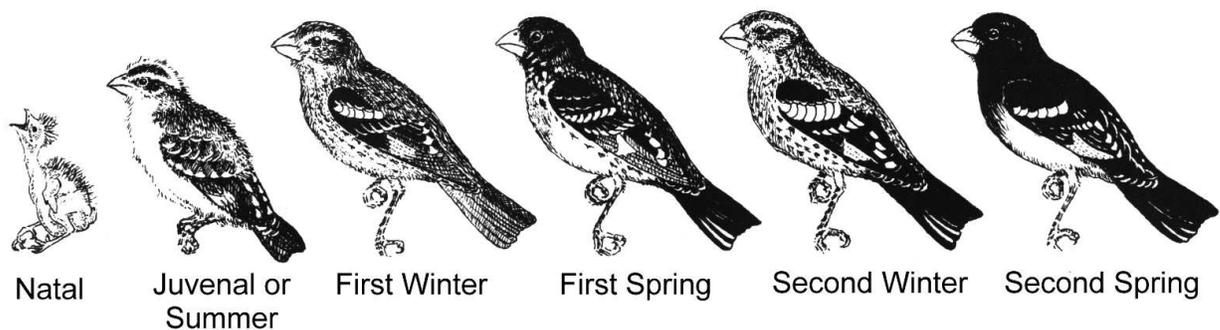


Figure 1.1: Successive plumages in the first two years of a Rose-breasted Grosbeak. Adapted from [WB88].

While modelling feathers and collections of feathers is interesting, to accurately model a feather coat integration of change or development is required. Within a single year most birds lose and replace (moult) all their feathers at least once and as many as 5-6 times. Since the moulting of the feathers is staggered, in any natural feather coat at any particular time feathers at various stages of growth or development can typically be found. Depending on the quantity and density of the immature feathers, they too can have a large effect on the appearance of the feather coat. This growth, development and change of the feather coat is known as *morphogenesis* and is an integral part of the feather coat model. Two examples of feather coat morphogenesis are shown illustratively in Figures 1.1 and 1.2.

1.3 Modelling Natural Phenomena

From a computer graphics perspective, devising a model for a particular phenomena involves considering many principles. Four basic principles are consider in this work; tailourability, level of detail, stochastic elements and dynamics. Since many of the scenes and objects are so complex, it is naturally desirable to have a fairly automatic method of creating these models. However, often methods that automate a process reduce the flexibility of the results. Thus, the model chosen, must not only be automatic, but must not limit the ability to create a wide variety of results - tailourability. Fournier [Fou89] outlined the other three principles for natural phenomena models: levels of detail, the use of stochastic elements and the role of time. In order to create a scene or object that appears realistic, it is often desirable to add as much detail as possible. However, this complexity increases rendering time² which is problematic when many frames need to be rendered (e.g. animation). Proper use of level of detail allows for the best balance between complexity and realism, minimizing complexity where and when it is not necessary. The use of stochastic elements adds to the realism of the model. Often many of these modelling techniques are far too regular to simulate “realistic” looking natural objects. Introducing random elements into the model (e.g.. height fields, branching angle etc.) increases realism. Fournier and Fussel [FFC82] defined stochastic modelling to be the combination of an ordinary model with a stochastic process³. Finally, considering the element of time in a model is essential. Fournier emphasizes that “if a model is intended for animation, time should be a parameter of the model” [Fou89, p. 197] and that if this is not considered and “there is a problem with the model, it **will** manifest itself sooner or later in an animation” [Fou89, p. 197].

The final element that must be considered in choosing the best model is how specific the model is to the object. Modelling of complex surfaces in computer graphics has often been done by using textures, bump maps or displacement maps to achieve a realistic look without having to model the actual surface or subsurface properties. However, there are some materials in nature, that when viewed from certain distances, are not adequately modelled using texture or displacement maps. And yet, modelling these materials using geometry exponentially increases the complexity of the modelling problem requiring special methods for modelling these types of materials. Thus, the development of special, object-specific, modelling methods may be needed. An example of the development of unique methods for models specific to objects, is the

²processing time required to convert the mathematical model or representation of model into an image

³processes that generate random variables

recent work of modelling the placement, structure and dynamics of both fur and human hair [Gol97, GW97, iAUK92, WS92].

Tailourability, level of detail, use of stochastic elements, inclusion of time and the use of object specific models are all common principles seen in the modelling and rendering of natural phenomena and computer graphics in general. These common principles will form the basis of an approach to the modelling and rendering of feather coat development.

1.4 Problem Scope and Goals

The ultimate goal of this project is to semi-automatically⁴ model the feather coat over a bird's lifetime (e.g. see Figure 1.1) such that the model can be animated and the feather coat responds acceptably. Due to the variability and morphogenesis of the feather coat appearance, a methodology that is driven by specific feather related parameters is needed. Also, given the complexity of feather coat structure and the desire for efficiency in animation there is a need for a multi-resolution model. The claim is that *modelling feather coat morphogenesis using a multi-resolution methodology that is feather parameter specific will best facilitate the creation of "realistic looking" feather coats for the purposes of animating the follicle surface or "skin"*.

Feathers, and consequently feather coats, could obviously be modelled manually. However due to the complexity of feathers and variability found on a bird's body, this could take the average modeller an extremely long time to construct. Also since the appearance of the feather coat changes over the lifespan of the bird and it must respond appropriately when the "skin" is animated, continuous alterations by the modeller would be required. Thus, it is desirable to have a method that could model most of the details automatically given some initial information from the modeller and respond completely automatically when the skin is moved, without losing the ability to create a wide variety of resulting "looks".

In order to preserve the ability to create a variety of results the approach taken must be tailorable and flexible. The user must have the ability to provide various initial parameters. The initial information that is needed must accurately describe the look of the feather coat as it develops. This would involve modelling the evolution of the placement of the feathers, the type of feathers and the colour of the feathers over the lifespan of the bird. The user will need to initially specify a model to be *feathered*, an indication of how various types of feathers are distributed, an indication of the colouration of the feathers and how these three

⁴semi-automatically - with a limited amount of user intervention to set the initial modelling parameters

1.4. PROBLEM SCOPE AND GOALS

parameters evolve over the lifespan of the bird. For the approach to be flexible, it is important to make no assumptions about the underlying model, in order to facilitate the placement of feathers on any model including inanimate objects. Even if the system was limited to feathering birds, this flexibility is needed because the bird's body can vary in their shape, structure and size. Creating such a versatile system with a minimal set of a priori parameters, would facilitate the placement of an eagle's feathers on any model (i.e. human or airplane) and accommodate simulating the growth of the model showing all the natural changes in appearance caused by the development of the eagle's feathers.

The focus of this research is the modelling of feather coat morphogenesis. This involves devising models for individual feathers and their growth, the placement of the feathers on the bird and a model for the order and timeliness of feather replacement. A secondary objective is the desire to animate the feather coat or to have it respond appropriately when the model is animated. This requires that the feathers be modelled in 3D, that the placement and orientation/direction of the feathers be consistent across movement of the underlying structure and that there be some mechanism for dealing with movements of the feathers caused by skeletal movement.

Obviously, modelling the feather structure and feather coat morphogenesis will not produce effective realistic results unless the model is also rendered realistically. For the necessity of completion, a method for modelling pattern and colour must also be devised. This incorporates the problem of rendering with the problem of modelling the feather coat. To realistically render the feather coat, feather details such as subsurface properties may need to be known to produce realistic effects such as the iridescence found in hummingbird feathers. Rendering of subsurface effects, is an entire area of active research in computer graphics. Hence the construction of an accurate renderer will be left for future research and this current research will focus mainly on the modelling of structural, pattern and movement aspects. Alternately, the goal of this work is to produce a model that is versatile enough to produce realistic results with an "off-the-shelf" renderer as well as produce non-photorealistic results using standard NPR shaders or filters. An example of a desired NPR rendering is given in Figure 1.2 which shows a 2D hand drawn illustration of feather coat development or morphogenesis.

One problem not addressed in this work is the various behavioural movements and flight patterns of birds. Motion and movement of the bird and feathers combined is an issue that must be considered in order to attain realistic results. This may involve simulating the motion of take off and landing, soaring, or flying - perhaps also the behaviours of sleeping, preening and bathing. Simulation of flight may involve not only the

motion of feathers during flight and aerodynamics [RW99], but also the various flight patterns for different families of birds. Exciting as these problems are, they are left for future researchers to consider.

1.5 Overview of Thesis

This chapter provided a general overview and some motivation for the problem and approach of modelling feather coat morphogenesis. Chapter 2 will overview the structural properties of feathers and characteristics of feather coats. Chapter 3 will review the current work in computer graphics that is applicable to the problem of modelling feather coat morphogenesis. Chapters 4-8 specify the modelling and rendering details of the feather coat morphogenesis methodology. Finally Chapter 9 provides an analysis of the results with conclusions and future work in Chapter 10.

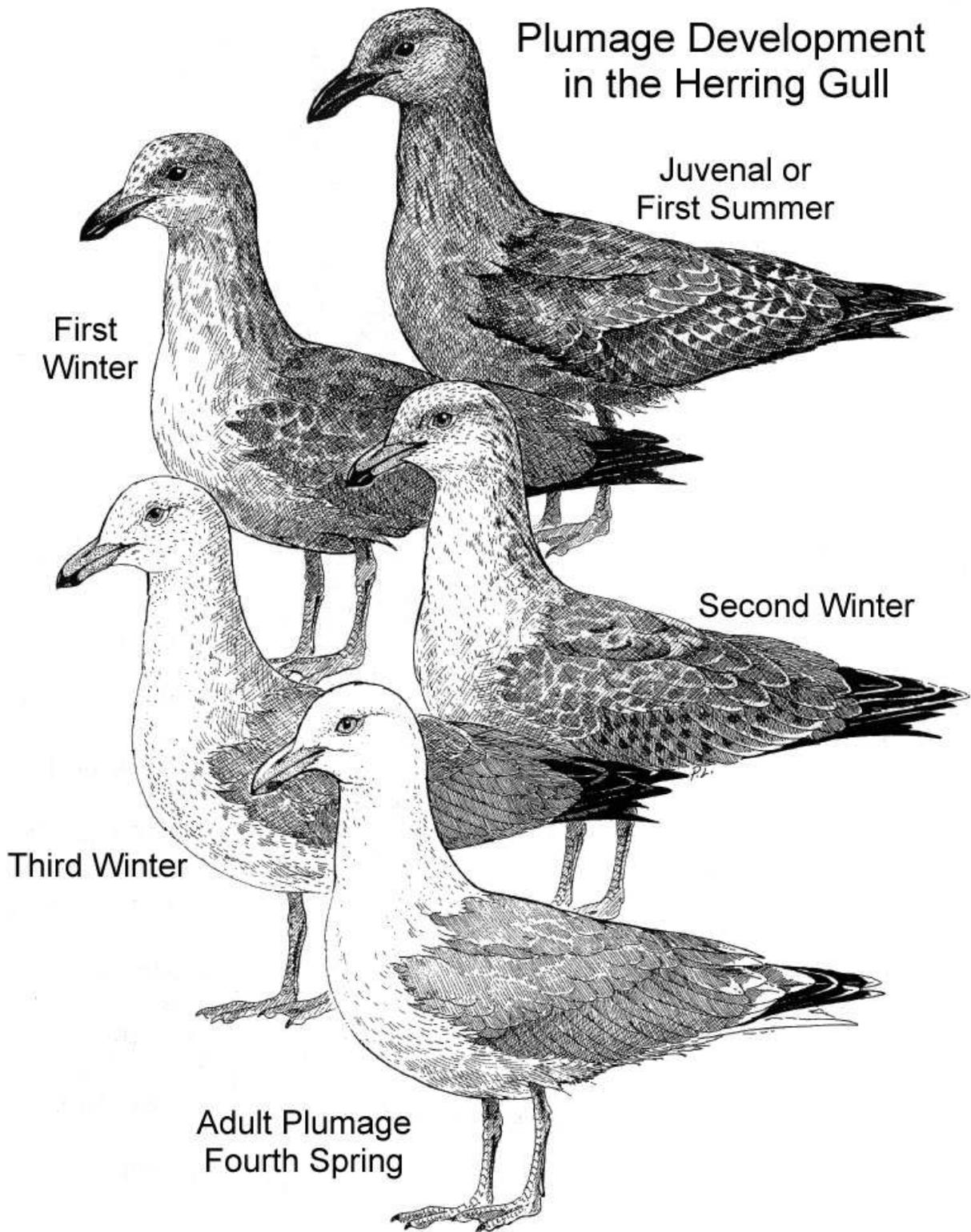


Figure 1.2: Feather coat development/morphogenesis from approximately 6 months to 5yrs of a Herring Gull as illustrated in pen-and-ink. Adapted from [PL93, p. 107].



Feather Morphology

2.1 Introduction

A bird's appearance is largely a function of its feathers. Together a bird's feathers weigh typically two and a half times the weight of their skeleton and are essential for sustaining life. Feathers aid in body temperature regulation, skin protection, camouflage, mate attraction, swimming, sound production, cleanliness, water repellency, water transport, tactile sensation, hearing and balance [Gil95, p. 65]. Feathers alter the appearance of the bird by forming both individual and collective (feather coat) patterns. The structural and chemical make-up of individual feathers create various pigmentations which contribute to the bird's overall appearance. As well, slight alterations in feather arrangement create a variety of contour shapes and patterns. This chapter describes the various key aspects of feathers that contribute to the overall appearance of the feather coat, including feather structure, position, and arrangement.

2.2 General Anatomical and Avian Terminology

In order to discuss the placement and structure of feathers a set of terms is needed to describe feather locations and directions on a bird's body. A *feather tract* or *region* is an area of the bird's body containing a specific arrangement of contour feathers (see Section 2.3.1). An *apteria region* is a region of either no feathers or no contour feathers and any feathers that are present are unorganized with variable distributions. Feathers in the feather tracts are cyclically lost (*moulted*) and replenished. The set of feathers grown after

2.3. INDIVIDUAL FEATHER STRUCTURE

a full moult is collectively known as *plumage*. However, since various regions may be at different stages of moult, the collective set of feathers worn by a bird at any particular time is known as the *feather coat*.

A set of terms is used to describe locations on a bird's body in the flight position as shown in Figure 2.1. *Ventral* and *dorsal* refer to the underside and topside of the bird. *Anterior* and *posterior*¹ describe locations toward the head and tail, respectively. Three perpendicular planes divide the body: *sagittal* dividing left and right (midline at centre), *transverse* dividing anterior and posterior and *frontal* dividing dorsal and ventral. *Proximal* means toward the midline, while *distal* is away. Within the frontal plane *medial* and *lateral*² describe a comparative relationship between objects closer to the midline and objects farther away. Finally orientation is described, as with aircraft using *roll*, *pitch* and *yaw* as shown in Figure 2.1.

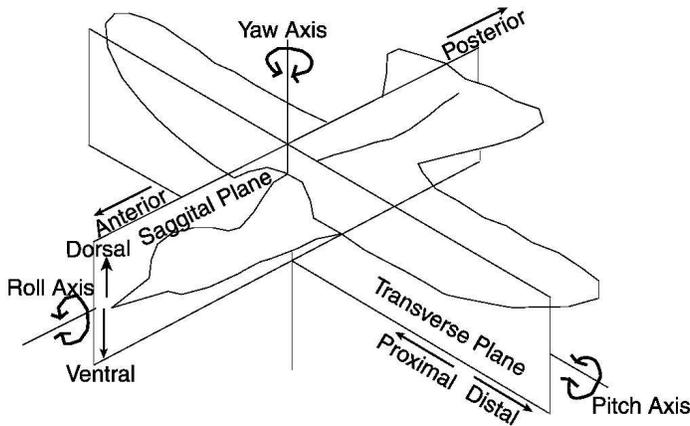


Figure 2.1: Common avian anatomical terminology (after [PL93, p. 9]).

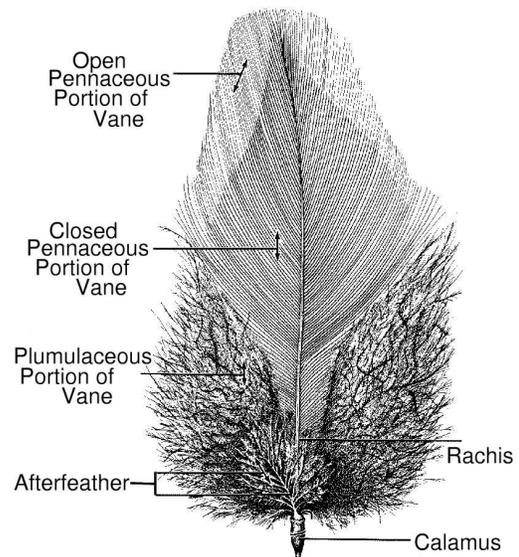


Figure 2.2: Main parts of a typical contour feather. Taken from the dorsal tract of a Single Comb White Leghorn Chicken [LS72a, p. 236].

2.3 Individual Feather Structure

A feather can be described as a structure with a main centre shaft and a hierarchy of fine branching structures extending from either side. The main shaft is called the *rachis* (distally) and *calamus* (proximally) and

¹Also called *cephalad* and *caudal* respectively

²Outside the frontal plane, the terms *mesial* and *peripheral* are used similarly.

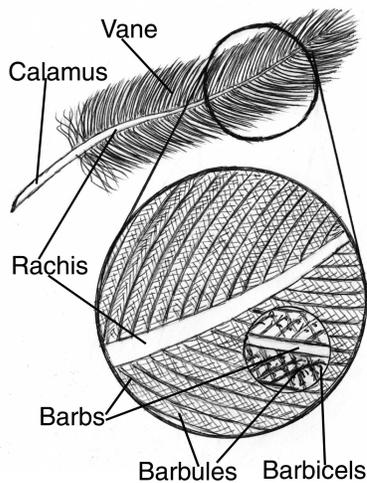


Figure 2.3: Typical flight feather with branching substructures.

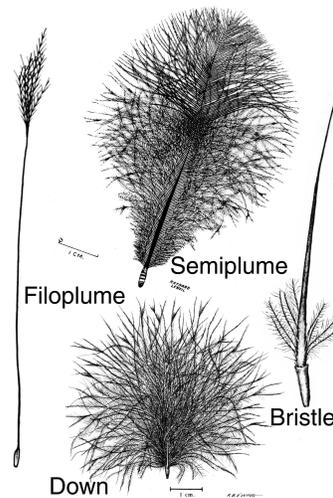


Figure 2.4: Examples of non-contour feathers (from Lucas and Stettenheim [LS72a, pp. 271,274,310,312].

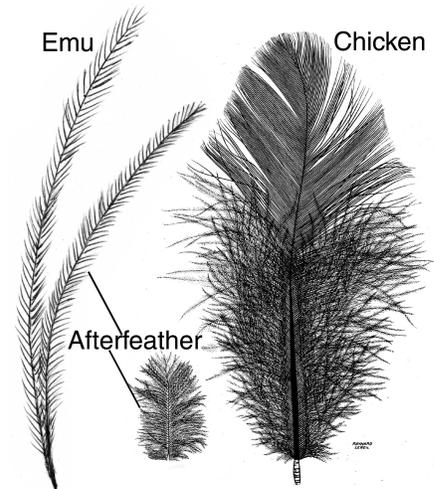


Figure 2.5: Afterfeathers: Emu (attached) and Chicken (in centre) (from Lucas and Stettenheim [LS72a, p. 290].

provides most of the mechanical strength of the feather (see Figure 2.2). From the main shaft branch *barbs* (*rami*) that extend angularly toward the tip of the feather. From the barbs branch *barbules* and from these possibly branch *barbicels* (see Figure 2.3). Collectively, the barbs (barbules) on one side of the rachis (barb) are called a *vane* (*vanule*). Finally, the main shaft can be thought of as a branch of the calamus, with the other branch containing the *afterfeather* (a structure similar to the main feather). The number of levels of branching micro-structure define the appearance of the feather.

There are multiple feather taxonomies. This thesis will use Lucas and Stettenheim's [LS72a] classification categories of feathers: *down*, *contour*, *flight*, *filoplumes* and *bristles*. The *afterfeather*, *semiplume* and *ear coverts* are considered special cases of the contour feather and the *powderdown* a special case of the down feather. Most feathers are either *plumulaceous* (non-interlocking barbules/no barbicels) or *pennaceous* (barbicels). The *contour feather* is most common and incorporates many feather substructures.

2.3.1 The Contour Feather

Contour feathers (see Figure 2.2) typically have a large, firm vane and a plumulaceous base. They are composed of the shaft, the left and right vanes (barbs and barbules) and an afterfeather.

2.3. INDIVIDUAL FEATHER STRUCTURE

Shaft

The shaft is composed of the calamus and rachis. The calamus (rachis) tapers distally and has a circular or oval (rectangular) cross-section [Voi66, p. 13]. The calamus is largely embedded in the follicle and has a hole (*inferior umbilicus*) through which pulp (blood) enters during feather growth. On the ventral side, where the calamus and the rachis meet, is an opening (*superior umbilicus*) from which the afterfeather grows. The calamus wall has two layers: an outer which extends beyond the calamus to the rachis tip, forming the *feather sheath* within which the feather grows and a thinner inner layer which acts as a barrier to the pulp. The calamus of a grown feather is hollow and transparent, but the rachis contains *pith*³ and is opaque.

Vane, Barbs, Barbules and Barbicels

The vane of the contour feather is composed of barbs and barbules, forms the surface or airfoil and defines the overall look of the feather. Vanes typically have a width-length ratio of 1.2 to 4 and are symmetric. The barbs vary in quantity, distribution, location, length, angle, and composition. The barb quantity is the same in both vanes but may range from a few dozen to several hundred, depending on the type and size of the feather. In contour feathers barb distribution is dense proximally and sparse distally and barb length increases distally from the superior umbilicus to approximately 1/4 the length of the rachis where it stabilizes and then decreases to the tip. The angle of the barbs is greatest at the base and most acute at the tip⁴. The width of the vane is dependent on both the length of the barbs and their angle. Barbs are composed of two layers, similar to the main shaft. The inner region of pith usually consists of many irregularly fused cells enclosing single cavities or small spheroidal vacuoles of various sizes [LS72a]. These various size vacuoles and the flattening of the barb to form a ventral or dorsal crest contribute to the various colours seen in feathers.

Extending from the barbs are barbules which are categorized into three groups: *plumulaceous*, *pennaceous* and *simplified* with practically a continuum of variation (see Figure 2.6). Plumulaceous barbules are symmetric, have notches or joints and can be constant width or taper distally. Pennaceous barbules make up the flat, closely knit portion of the vane of the contour and flight feather by having interlocking

³Pith is a spongy tissue (air-filled cells) formed from the keratinization of long epithelial cells.

⁴The change in barb angle is very subtle causing adjacent barbs to appear parallel

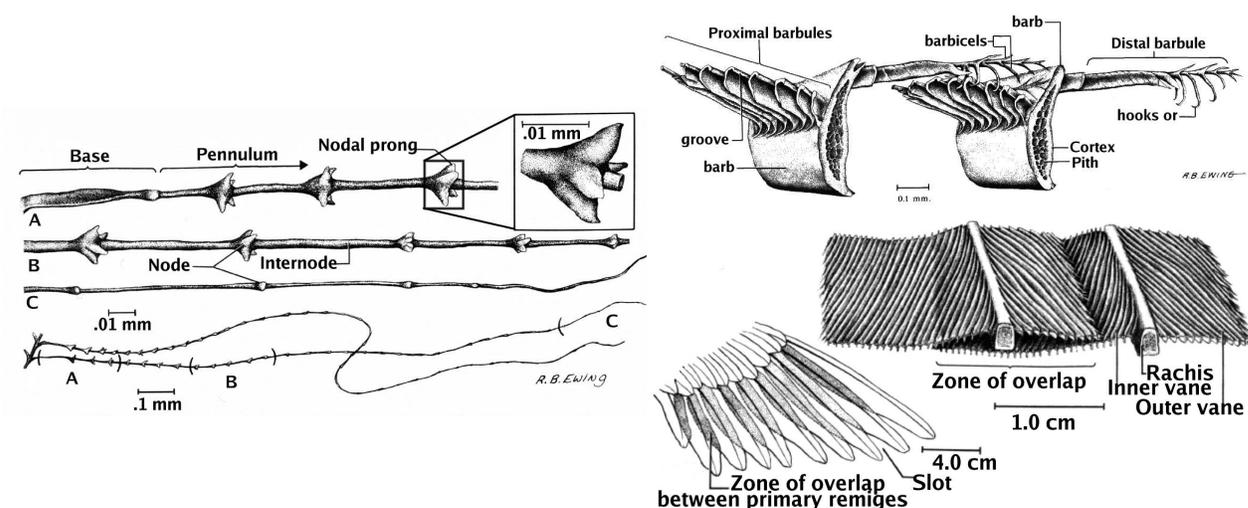


Figure 2.6: Barbs of common pigeon. **Left:** Plumulaceous **Right Top:** Pennaceous **Right Bottom:** Zone of overlap in primaries. All modified from [LS72a, pp. 244,247,261].

grooves on the proximal side and *hooks*⁵ on the distal side and are typically angled from the barb at 20–60 degrees and are asymmetric. Finally, simplified barbules possess characteristics of both plumulaceous and pennaceous barbules and are subdivided into *stylet*, *reduced pennaceous*, *reduced plumulaceous*, and *rachidal* [LS72a, p. 251], but are always simplified from the two extremes (e.g. no hooklets or notches, smaller, stiffer, more pointed etc.). All barbule lengths taper distally along the barb and both hold barbs apart to aid in parallelism (notches restrict movement) and hold adjacent feathers together using friction. A single feather may possess all types of barbules with a gradation from plumulaceous to pennaceous distally along the feather. Stylet barbules can be found at the tips of barbs forming the open pennaceous portion and rachidal barbules may be found on the rachis. The ratio of plumulaceous to pennaceous barbs varies and is one of the criteria for defining the type of feather.

Afterfeather

The afterfeather⁶ (see Figure 2.2) is always plumulaceous, has shorter barbs, and barbules than the main feather and can range from a fusion of a couple of barbs to a complete feather. The afterfeather's size and existence is not reliable, but is associated with the feather tract, type, species, and season [LS72a, p. 253]. Emus and Cassowaries have afterfeathers that can be as long as the main feather (see Figure 2.5).

⁵also called *hamuli*, *hooklets* or *barbicels* - range from 2 to 4 in body contour feathers and 3 to 7 in flight feathers

⁶also called hyporachis, hypopennae and aftershaft, however hyporachis also refers to the rachis of the afterfeather

2.3. INDIVIDUAL FEATHER STRUCTURE

2.3.2 Flight Feathers

Flight (large wing, *remiges* and tail, *rectrices*) feathers⁷ are typically larger and stiffer than other feathers and have asymmetric and almost entirely pennaceous vanes (see Figure 2.3). The outer vane is always narrower than the inner and usually overlaps the inner vane of adjacent feathers (see Figure 2.6). Specialized barbules create friction in the overlap region. This overlap causes the inner secondaries (see Section 2.4.1) and the central rectrices to be the uppermost feathers when the wing and tail are folded. Barb length in flight feathers is more constant than other contour feathers. Barb spacing changes inversely to angle along the rachis which is equal in both vanes for the secondaries and rectrices, but is more acute in the outer, stiffer vane than the inner for the primaries. The pennaceous barbules are uniform size and are spaced 30-40 (proximal) and 20-30 (distal) μm apart for all birds.

2.3.3 Down Feathers

Down feathers are entirely plumulaceous and have a rachis shorter than any of the barbs, distinguishing them from semiplumes (see Figure 2.4). *Natal (immature) down* is lost and replaced by a different type of feather, while *definitive (adult) down* is continuously present in a fixed follicle. Natal down have no afterfeathers and possibly no rachis, while definitive down have longer calami, always have a rachis and generally mimic the characteristics of the bird's contour feathers (i.e. afterfeathers are only present in down when they are present in contour feathers). Definitive down barbs are much narrower than the contour feather's pennaceous barbs and the barbules tend to be longer and less variable than the contour feather's plumulaceous barbules. *Powder down* feathers continuously grow and wear at the tips forming a talc-like powder⁸ that coats the feathers. These feathers can be various colours, although the powder is always transparent.

2.3.4 Contour Feather Variations, Filoplumes and Bristles

Semiplumes have characteristics of both down and contour feathers and form a continuous gradation from semiplume to contour feather. They are entirely plumulaceous and the rachis is longer than any barb (see Figure 2.4). Tyne [vTB76, p. 120] described them as "loose-webbed plumulaceous contour feathers". *Ear coverts*, also are a form of contour feather, are quite small, arranged in circular rows, all point in the same direction (posteriorly), and have liberally spaced pennaceous barbs.

⁷Can be up to six feet in length (tail feathers of Crested Argus).

⁸Some powder is also produced from the sheath of other feathers, but most comes from powder down.

Filoplumes (see Figure 2.4) have long hair-like rachis with a few (2 to 12) flexible barbs near the tip and are the only feathers that do not have muscles attached to the follicle. The calamus is less than 2 mm long while the rachis can range from 1 mm to the length of the host feather, is filled more solidly than other feathers and is thickest at the distal end. The barbs are similar to downy barbs, but tend to be shorter and stiffer. Long downy-like barbs grow at the base, but are lost as the feather develops.

Bristles have a stiff, thick rachis at the base tapering to a fine point. Barbs are restricted to the proximal end and barbules are simplified and may diminish or disappear near the tips of the barbs. Bristles typically do not have afterfeathers. Semi-bristles may have barbs along most of a slightly longer rachis (see Figure 2.4). The gradation between contour and bristle can be described by lengthening or proximal thickening of the rachis, loss/shortening/wider spacing of distal barbs (so rachis extends beyond them), barbule simplification, loss of barbules (starting at tips of barbs), and loss of afterfeather except possibly hyporachis.

2.3.5 Feather Shape and Curvature

“Except for colour, there is probably no property of feathers as diverse as shape” [LS72a, p. 240]. Feather shape is established during growth and does not change thereafter, except for wear. Feathers are most commonly *ovate* (egg-shaped - broadest proximally). Other shapes such as *obovate* (upside down egg-shaped), *spatulate* (short broad tips, with few/no barbs proximally), *lanceolate*, and *attenuate* are less frequent. These latter two types are more narrow than long and either taper distally (*attenuate*) or both proximally and distally (*lanceolate*). Feathers can have pointed, round (contour feathers), or truncate tips. Gill gives some examples of variation in tail feather shape (see [Gil95, p. 72]). Most feathers have symmetric planar shape. However, flight feathers especially primaries and their coverts often have asymmetric, notched (abrupt narrowing) vanes which emphasizes the feather overlap and reduces drag.

Most contour feathers have distinct curvature along an axis except in rare cases (e.g. penguin - short, flat contour feathers⁹). The rachi of all contour feathers curve ventral/dorsally into the body (except some rare French Hens and Frizzled Bantams [Gil95, p. 128]) and may also curve laterally (e.g. rectrices of Honey-guide). Barbs can curve to form either a 'c' or 's' cross-section of the feather. These curvatures develop from differential growth. The consistent ventral/dorsal curvature of contour feathers may be attributed to feather growth, but afterfeathers reverse this curvature once mature (see Section 2.6).

⁹These feathers form a frictionless scale-like surface for swimming efficiency, insulation and waterproofness.

2.4 Feather Coats

A feather coat is the collection of feathers on the bird at a particular time. Due to moulting this may mean that in a particular coat there are various types of feathers from various different plumages¹⁰ at various stages of growth. No bird necessarily possesses all types of feathers, but different types of feathers tend to be found in particular locations on a bird and on birds from particular environmental climates. Also, as the bird matures, the feather coat evolves typically from mostly down to mostly contour. For instance down is used for insulation and thus is found in regions close to the body and is more prevalent on birds from arctic and water environments, while contour feathers are needed for flight and aerodynamics and are therefore present on the wings and the outer surface of the body. This section will overview the distribution of follicles which hold the feathers as well as the orientation and alignment of the feathers, with particular emphasis on contour feathers, since they have a larger impact on the bird's appearance.

2.4.1 Follicle Distribution

Feather tracts (pterylae) identify the location of contour feathers. Other types of feathers such as down, semiplume and powder down, are distributed in apteria regions and/or underneath the contour feathers. They lack a local distribution or density pattern, but can be restricted to certain tracts or species of birds. Below is a brief overview of the distribution of non-contour feathers, and then more detailed overview of contour feather distribution.

Semiplumes can be found in apteria, along the edges of pterylae as well as singularly within a pterylae. *Natal down* is generally distributed throughout the body and is visible, while *definitive down* may be evenly (e.g. penguin) or unevenly (e.g. owls and gulls) distributed throughout the body, restricted to pterylae (e.g. tinamous) or apteria (e.g. herons and most galliform species) and is generally concealed beneath the contour feathers. *Powder down* is restricted to certain groups of birds (e.g. pigeons, storks, herons, toucans, parrots and bowerbirds) and may occur in patches (e.g. herons). *Afterfeathers* are present depending on the type of bird, feather and season. They are not found on any of the larger flight feathers (see Section 2.3.2), but are consistently found among other contour feathers. *Filoplumes* grow close to a "host feather"¹¹, are present in all birds except the Ostrich, Emu and Cassowaries and are either underneath

¹⁰plumage - the feathers a bird posses at the completion of a moult

¹¹the host feather is usually a contour or semiplume and may have 2 to 10 filoplumes associated with it.

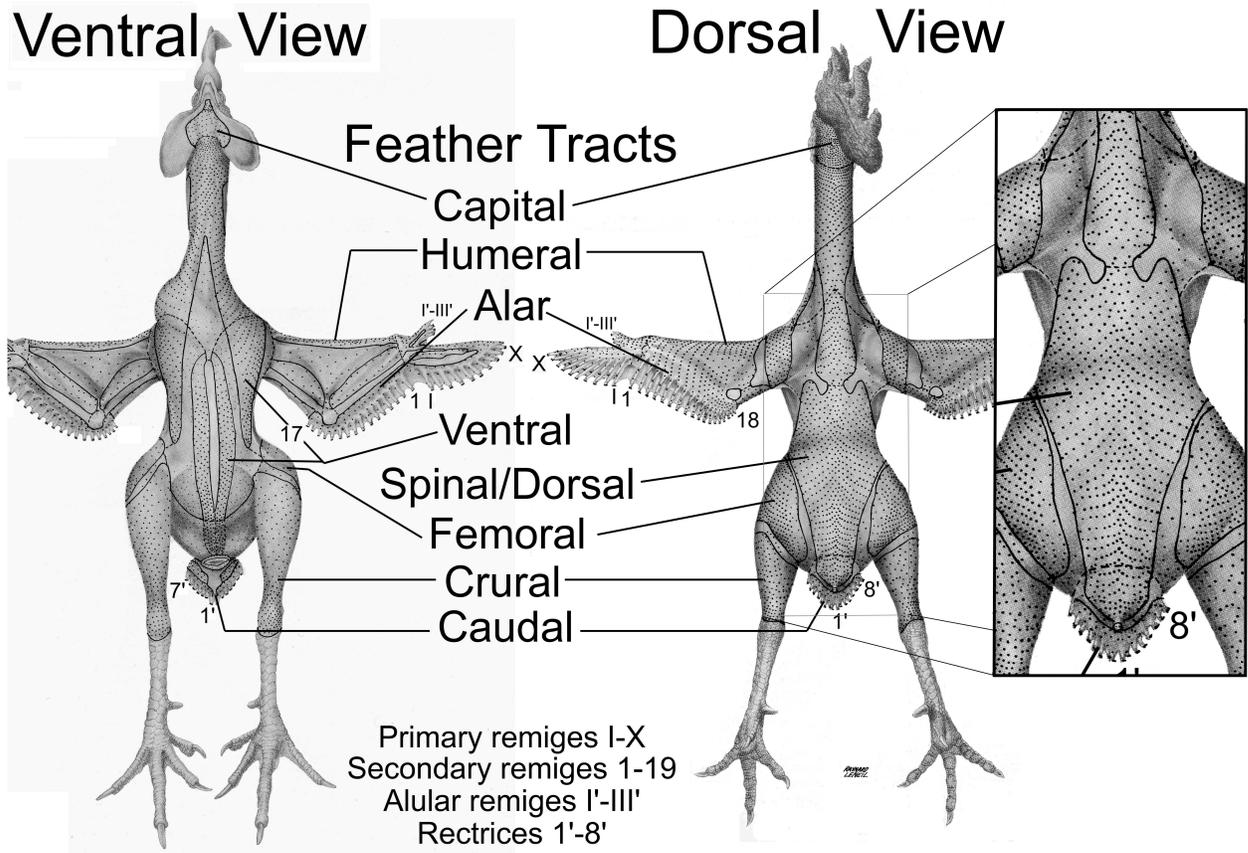


Figure 2.7: Eight main pteryalae areas on a Single Comb White Leghorn Chicken modified from [LS72a, p. 105,108]. Magnified region of the dorsal tract. Note follicle arrangement in rows.

contour feathers and thus are not visible, or between contour feathers [vTB76, p. 122]. *Bristles* are almost exclusively found on the head of birds¹² around the mouth, nostrils, eyes and crown.

Contour feathers are named from their placement - they form the surface/silhouette or contour of the bird. In most present day species contour feathers are restricted to and have a well-defined distribution within pteryalae. Some primitive birds, flightless birds (e.g. penguins, ostriches, ratite birds) and a few flying birds (e.g. screamers, mousebirds) have no apteria and thus have a hair-like distribution of feathers. This thesis discussion will concentrate on birds with both pteryalae and apteria. The feathers in pteryalae usually cover adjacent apteria. The eight main pteryalae (see Figure 2.7) are: *capital*, *spinal or dorsal*, *humeral*, *ventral*, *caudal*, *alar*, *femoral* and *crural* tracts. These regions have also been sub-divided, non-standar

¹²except on the Bristle-thighed Curlew and on the toes of some owls

(e.g. capital tract into frontal, loreal, malar, postauricular, auricular, occipital, ocular, superciliary). For more details on subregions of these eight main tracts see [Mil28].

Embryological Development of Follicles

The embryological formation of follicles assists in understanding the arrangement of feathers within the tracts. Initially, the dermal and epidermal layers of the skin differentiate and form clusters of dermal cells called (blastema) which are the start of feather rudiments¹³. A central longitudinal row of circular rudiments form, then rows of rudiments develop laterally, beginning at the centre of the row and proceeding peripherally.

Typically the first longitudinal row of rudiments is created near the centre of the tract (along the middorsal line). Before the middorsal row is finished forming, the lateral rows begin to differentiate into placodes. These rudiments are formed as closely as possible to the first row and to each other, so each new rudiment is tangent to two rudiments in the previous row forming a regular hexagonal pattern. However since these rows are formed as the bird grows they "stretch" not only longitudinally, but also laterally and obliquely as seen in Figure 2.7. All pterylae have this arrangement except the Alar and Caudal tracts where growth is axial and follicle maps are known. Holmes [LS72b, p. 347] states that the initial formation of tracts

“...have regular orientation and location in relation to the bird’s body: ... the dorsal and ventral sides of the tract are oriented longitudinally, those on the thigh, more or less dorsal-ventrally and those in the upper surface of the wing are parallel to the trailing edge.”

Capital, Humeral, Femoral and Crural Tracts

The capital tract consists of the feathers on the head and neck, ending at the ventral and dorsal tracts. The humeral tract has little variation, is narrow and extends from the shoulder across the top of the wing to the elbow, including the marginal coverts and the alula. The femoral tract is a narrow uniformly shaped tract extending from the femur region along the flank toward the tail. The crural tract is sparsely feathered and covers both the inner and outer regions of the leg. The capital tract, does not have documented placement of follicles, but is divided into many sub-regions where follicle locations are defined for particular species.

¹³blastema - a small mass of living substance capable of growth; rudiment - start of organ development

The femoral, humeral, crural tracts are less organized and while they still possess the hexagonal arrangement of follicles, this arrangement is not obvious and an axis of follicle development is not as clear as with the ventral and dorsal tracts.

Ventral and Dorsal Tracts Distribution

The ventral tract consists of all contour feathers on the underside of the bird except for the humeral and femoral tracts. It extends from the base of the neck to the cloaca and is usually forked. The dorsal tract extends down the back from the base of the skull to the base of the tail and is highly variable across species. For examples of this variation compare Passinerformes and Cululiformes order [Cle70, p. 659,665] with Falconiformes order [Com37]. The ventral and dorsal tracts are well-organized with a specific hexagonal distribution although the size of these pterylae regions and their number of feathers can differ across species.

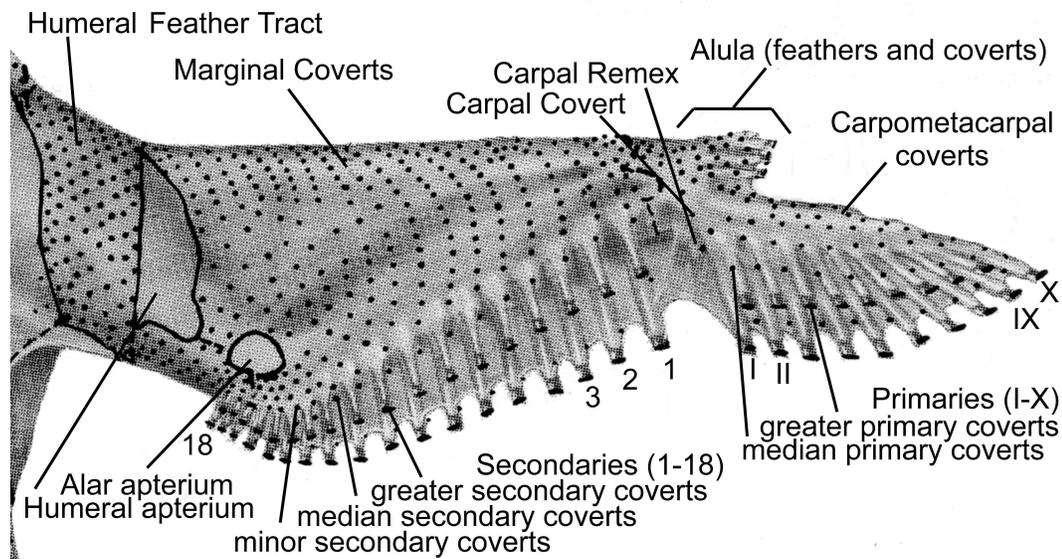


Figure 2.8: Primary, Secondary, Carpal Remex and coverts. Magnification of of dorsal view of Figure 2.7 modified from [LS72a, p .105].

Alar and Caudal Tract Distribution

The larger feathers of the alar (*remiges*) and caudal (*rectrices*) tracts¹⁴ are responsible for flight control and hence are called flight feathers (see Section 2.3.2). The Alar tract contains primary and secondary remiges

¹⁴Singular of remiges is remex and singular of rectrices is rectrix

2.4. FEATHER COATS

and coverts (see Figure 2.8). Along the wing the remiges are aligned in a row with the primaries numbered distally and secondaries numbered proximally. Most flying birds have 9–12 primaries, while non-flying birds typically have 3–16. There are typically 9 secondaries, but can be up to 14 in Passerines and 32 in other birds dependent on forearm size [vTB76, p. 129]. For recorded numbers of remiges, and rectrices see [vTB76]. The first six or seven primaries are attached to one fused bone, with a possible gap between the 5th and 6th depending on the species, where others are attached to the fingers or digits (see [vTB76, p. 127]). Each remex is covered proximally by a greater covert both ventrally and dorsally and by smaller and smaller coverts (median, minor) until the leading edge of the wing is reached. Some extra coverts (*Carpal Remex*) or extra feathers (*alula* 2–6) may exist depending on the species.

In the caudal tract, the number of rectrices is even, numbered from the centre outward. Typically there are 12 (6 pairs), but they can range from 4 to 32. They are covered dorsally by greater upper tail coverts which are covered by lesser upper tail coverts with the exception of the centre rectrices. Coverts are present ventrally, though less defined.

2.4.2 Feather Orientation and Shape

Feather Direction/Orientation

The general direction of contour feathers is always from the head to the tail in flight position. “The apical ends of the feathers [on a bird’s body] are oriented toward the end of the body” [Voi66, p. 39]. Feathers overlap each other in a shingle-like fashion with the top-most feathers around the beak. Local differences in orientation occur at the edges of pterygiae where feathers are more laterally oriented to cover adjacent apteria. Flight feathers also have a local arrangement. “All of the remiges overlap laterally in a uniform way: the inner vane of each feather is overlapped ... by the next proximal feather” [vTB76, p. 136] (see Section 2.3.2 and Figure 2.6). All greater coverts also overlap in the same consistent order, however other coverts may reverse this order. The ventral wing coverts have a consistent reverse ordering. Rectrices and their coverts are consistently layered with the centre as the top-most and the outer rectrices as the bottom-most.

Wing Shape

Wing size and shape varies across families, from long and distally pointed for migration to short and round for navigation. Body mass is also a factor in wing length (wing-loading). Longer wings have disproportion-

ally more secondaries than primaries. The relative lengths of the primaries define the shape of the wing and tip location and are “extraordinarily consistent in any given kind of bird” [vTB76, p. 127]¹⁵. A constant or distally decreasing relative length creates a round wing and distally increasing creates a pointed wing.

2.4.3 Number and Size of Contour Feathers

Feather counts vary from 940 (Ruby throated hummingbird) to more than 25,000 (Whistling Swan). Within species however, the feather tracts of adults are consistent across age, sex and plumage stage [Cle70, p. 676] causing feather counts to vary only from plumage stage (i.e. moulting or abnormal loss¹⁶) or damage. The variation in feather counts is considered statistically insignificant [vTB76] [Cle70, p. 676].

Feather count and size are physiologically and statistically related to environmental temperature, bird’s metabolic rate, body weight, and body surface area. While no mathematical relationship exists for all birds, studies across taxonomies associate a decrease in the bird size with a decrease in body weight, feather count and feather area, but at varying rates [LS72a, p. 193]. Wear alters feather length, weight and density.

2.5 Feather Patterns

Feathers combine pigment and structure to create more extraordinarily varied colours, than for any mammal and most vertebrates (except some coral reef fishes) [WB88, p. 48]. Some insects have similar colouration.

2.5.1 Pigments or Biochromes

Most feather pigments (*biochromes*) are from: *melanins*, *caroteniods* and *porphyrins* [LS72b, pp. 391-419]. Melanin is granular and has variable shape and colour, but generally produces earth tones. Caroteniods produce bright red, orange, and yellow and porphyrins produce some non-iridescent red, magenta, and green.

Melanin is the most common pigment and is found in all bird’s feathers except albinos. *Eumelanins* are large rod shaped blackish granules that produce dark browns, grays, and blacks and *phaeomelanins* are smaller oval or sphere shaped reddish or light brown granules that produce tans, reddish browns, and some yellows. These granules are deposited into developing barbs and barbules by mobile pigment cells called

¹⁵from Eisenmann 1969 - Bird Banding Vol.40 pp144-145

¹⁶abnormal or unusual circumstances include breeding, fright, sickness, or extreme sudden temperature change

2.5. FEATHER PATTERNS

melanoblasts which manufacture melanin from the amino acid tyrosine. These melanoblasts are believed to be distributed in a process similar to reaction diffusion. Periodic depositions of melanin create subtle colour pattern such as barring or speckling. Shades of various colours are a result of the granule density. Melanin (especially the darker colours) is associated with keratin creating a stiffer feather (e.g. proximally dark bristles in a light feather coat [LS72b, p. 272]).

Carotenoids and porphyrins are less common than melanins. Carotenoids (or *lipochromes*) are obtained through dietary sources (e.g. Vitamin A), are organically soluble and thus carried under hormonal control by body fluids to different feather regions. Keratinization dissipates the solvent fixing the pigment location. Porphyrins are related chemically to iron-containing hemoglobin and thus produce fluorescent red under ultraviolet light. They also produce intense red-copper, bright magenta and some olive green colours¹⁷.

2.5.2 Structural Colours

Structural colours (*schemochromes*) are caused either by a scattering or interference phenomena. The scattering phenomenon [oPA01] is view-independent and either scatters one particular wavelength (*Raleigh*) or scatters all wavelengths (*Mie*) of light. As white light enters a layer of particles only the wavelengths of light larger than the particles pass, reflecting the rest (typically blue, i.e., *The Tyndall Effect*). The passing light is often absorbed by a dark layer of melanin below the particles, purifying the reflected light. The interference phenomena is caused by the reinforcement or nullification of wavelengths due to refraction and reflection. It can produce view-dependent shimmering or iridescent colours (as in soap bubbles) and most recently has been thought to produce structural blues [Ons01]. Typically, the fine structure that causes refraction or reflection is layers of non-opaque substances or surface deformities. Peacock barbules are covered by three very thin layers of keratin. Hummingbird barbules are twisted and have multiple layers of oval platelets and air bubbles. Starling barbules have countless, precisely aligned gas-filled tubes of melanin. The colour reflected depends on the size of the gas pockets and the thickness of structures causing a range from blue-violet to coppery or greenish-yellow iridescence.

¹⁷The porphyrin turacin creates bright magenta while turacoverdin creates olive greens

2.5.3 Colour Variation and Changes

Generally, pigment is responsible for most non-iridescent colours, however some (e.g. blues, greens, and purples) are created from structure or a combination of structure and pigment. Colours can be created using multiple sources such as various structures, pigments or the use of spatial arrangement. Multiple pigments can be spatially combined (e.g. red and yellow) to create other colours (e.g. orange). Structure and pigment can be combined, such as a layer of structure for scattering (e.g. blue) above a dark absorbing layer of melanin causing bright blue, or containing a layer of yellow (red) pigment causing green (purple). Or a layer of large granule structure (reflecting all wavelengths) with a layer of pigment below which clouds or dims the result. Finally, using one structure and/or pigment on one part of the feather (e.g. blue/white barbules) and another structure and/or pigment in another area (e.g. red barbules) varies results (e.g. purple/pink feathers) from a distance. Age, genetics, and hormones affect feather coat colouration, but once a feather is

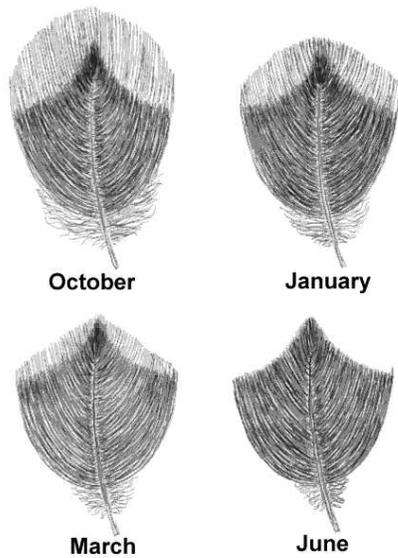


Figure 2.9: Snow Bunting changing from brown (Oct.) to black/white (Jun.) from winter tip wear (adapted from [WB88, p.50]).

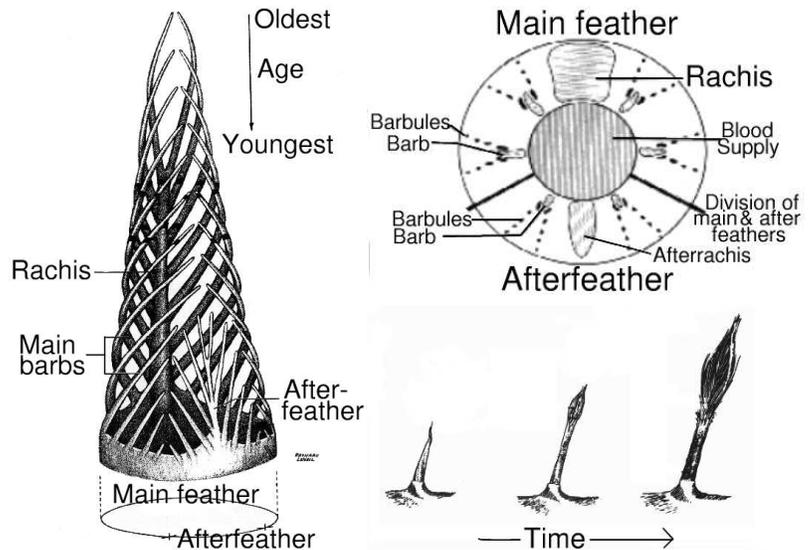


Figure 2.10: Contour feather growth with cross-section (from [LS72a, p. 200,370]).

grown it receives no new pigment, thus colour change can only occur through wear, bleaching, or external substances. Pigmented feathers are more resistant to wear due to the presence of keratin. Wear of a light tipped feather darkens the observed colour (e.g. Red Crossbill - pink to red). Sunlight sensitive pigments can be altered through bleaching (e.g. light pink/pale red-brown will fade to white/grey [WB88, p. 51]).

Colour alterations can also be caused by substances emitted and carried in the feathers, though this occurs less frequently. Two natural substances are oil gland secretions (slight yellow or pinkish pigment) and powder down fragments which although transparent cause dimming or blue-gray tones due to interference (e.g. gray/red/yellow becomes blue-gray/violet/green). Foreign substances such as dirt darken the feathers and iron bearing water or soil cause yellowing or reddening (e.g. Bald Eagle). Finally, genetic or hereditary mutations can also affect the colouration, although are rare and non-standard (e.g. Melanism, Erythrism, Schizochrism, Xanthochrism and Albinism [WB88]).

2.5.4 Pattern and Pigment Location

Bright colours and bold patterns typically only exist on visible portions of the feathers and bird (e.g. wing, breast, tail, head, neck, and distal portion of contours). Ventral or concealed body regions and proximal portions of the feather are duller (e.g. white/gray/dull brown plumulaceous and down feathers).

Melanin is present in all parts of the feather except the barbicels. Non-iridescent colours from melanin and carotenoids are found mainly in the rachis and barbs and sometimes in the barbules. Carotenoids are only deposited in early feather growth, hence are only present in the pennaceous portion of the feather and are often not found in the barbules. Neither melanin nor carotenoids are typically deposited in the final growth stage making the calamus white or clear. Melanin and carotenoid deposition within barbs can be alternative or simultaneous. The fine structure needed for iridescent colours is only present in barbules. Larger particles are found in plumulaceous feathers, producing whiteness or cloudy pigmentation.

Very little is known about individual feather pattern formation except that the presence of melanoblasts is needed for melanin production, but they can be present and inactive [LS72b]. Altering activity and inactivity over time or location can create spatial patterns such as barring or spotting.

2.6 Morphogenesis

Feather wear and damage necessitate replacement in order to preserve colour and retain airfoil and heat. Feathers fall or are pushed out of their follicle by the new growing feather (see Figure 2.11). Curvature and pattern are defined during initial growth, but is altered through moult and regrowth.

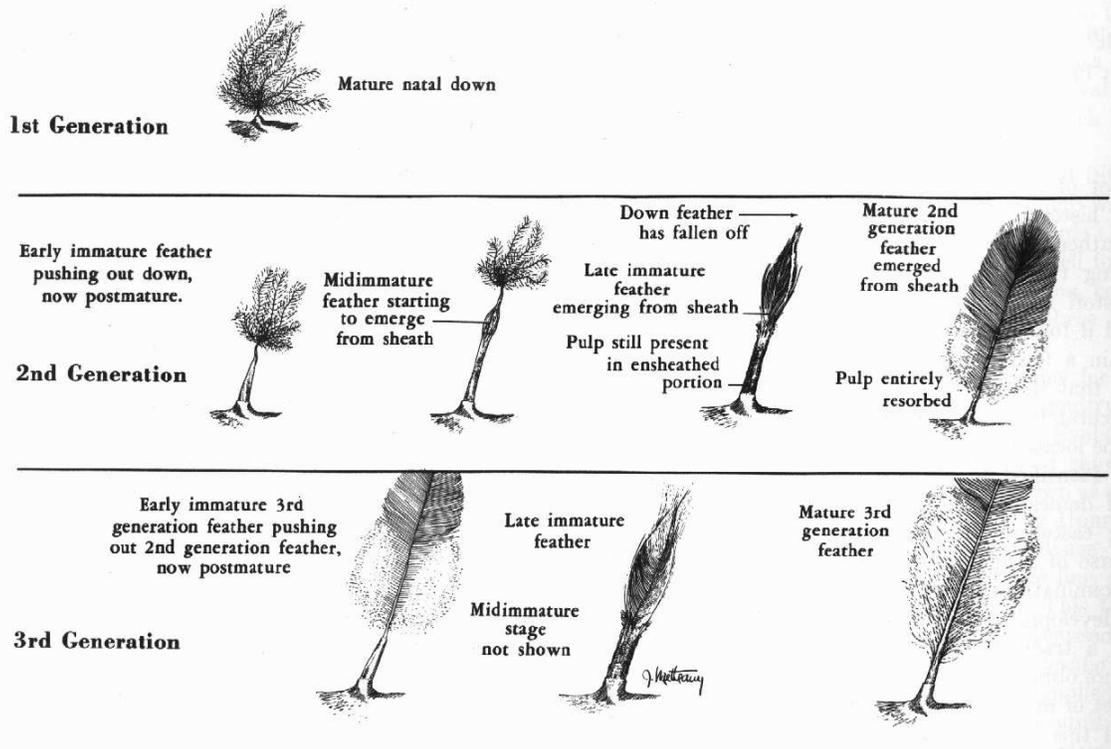


Figure 2.11: Development of feathers above the skin [LS72a, p. 200].

2.6.1 Feather Growth

Once the follicles are developed (at the embryonic stage), they do not change. A few follicles may be added to the edges of a tract during the juvenal or adult plumage stages, but this is often minimal and infrequent.

“A feather does not grow like a plant, by sending out branches and twigs mostly at the periphery. Cell division occurs mostly at the base of the feather germ and the parts differentiate as they move upward” from Lucas and Stettenheim [LS72b, p. 351].

As a feather grows, it evolves from a blastema, to a rudiment, to an immature feather (barely protruding from the follicle), to a mature fully developed pin-feather or quill and eventually it unravels from the quill to its final formation as shown in Figure 2.11. During growth, the feather is encased in a protective sheath. The feather is laterally curled toward its ventral side within the sheath, resulting in a quill-like formation, thickest at the base and tapering distally (see Figure 2.10). All cell division occurs at the base of the feather, causing the quill to age distally. A supply of blood and pulp at the centre of the quill feeds the feather as it grows. The barbs and barbules are formed at the base and are attached to the rachis systematically as the

feather grows out of the follicle where they continue to develop. As the nutrients in the pulp are absorbed by the developing barbs and barbules at the apex, the growth rate exceeds the rate of pulp production leading to a recession and capping of the pulp. As the feather reaches maturity, its structures (barbs, barbules and sheath) are keratinized¹⁸ and the sheath becomes brittle and falls off, revealing the final feather. Once the feather unravels the lateral curvature is lessened. The afterfeather also grows curled laterally, but toward the dorsal side and reverses its curvature once unravelled.

2.6.2 Moulting

Moulting replenishes various types of feathers, from various locations at different times (see Figure 2.12). Regeneration of the complete feather coat (sequence of moults) takes approximately one to two years depending on feather size. The sequences of moult and feather loss within a moult are discussed below.

Feather Coat Regeneration

In a typical year most birds have one complete moult (late summer) replacing all feathers and a second alternate moult before breeding (spring) replacing only head, neck and some body feathers [PL93, p. 106]. In the first year four or five moults lead to the final adult plumage. Once this is acquired only the alternate moult and feather wear can affect the appearance. The sequence of plumages is: natal down, juvenal, first basic, first alternate, second basic, second alternate etc., where first, second etc. is the bird's age in years. The final adult coat (second basic) is acquired at breeding age (approx. 1.5yrs¹⁹).

Newly hatched chicks may be bald or have natal down. The natal down follicles are later used for contour feathers. The first basic plumage is quickly acquired after the juvenal, often before moulting the flight feathers. Adult plumage is longer and more tightly textured than the juvenal, is brighter and more complexly patterned than the first basic, and arrives in time for breeding season.

Order of Feather Regeneration

With few exceptions (e.g. penguins) feathers are never lost randomly or all at once. Feather regeneration is ordered and consistent among species - first the primaries, then the secondaries, body contours and rectrices.

¹⁸Keratin - a special type of protein that is resistant to bacterial decay even once the feather has finished growing.

¹⁹can take up to 4.5yrs as with the Bald Eagle or Herring Gull (see Figure 1.2)

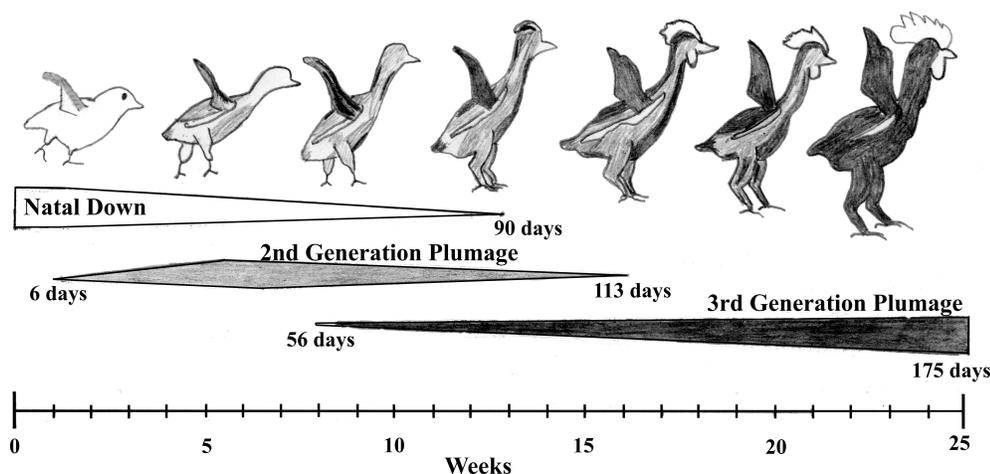


Figure 2.12: Successive plumages in a Single Comb White Leghorn Chicken. Approximated from data from Lucas and Stettenheim [LS72a, p. 207–210].

Within each group is a defined sequence. The primaries (on both wings) moult distally in sequence with growth of one starting just before the loss of the next²⁰. Once the moult of the primaries is half complete, the outermost secondaries (on both wings) moult proximally in sequence. Just before the secondaries finish moulting the body contours start moulting at the head and proceed posteriorly. Within each tract the feathers tend to moult from the centre outward or proceed in waves posteriorly or anteriorly. Finally, both halves of the rectrices moult either proximally or distally depending on the species. Some birds have minor exceptions to these orderings.

Different tracts moult at various frequencies, but a moult is complete only once all the bird's feathers have been replaced at least once. Thus the body feathers may moult three or four times (e.g. ptarmigans) before the completion of the first basic plumage. A single region's moult takes approximately two months.

2.7 Summary

This chapter outlined feather and feather coat specific information that is necessary for simulating feather coat morphogenesis. First, the structures, shape, and curvature of various types of feathers were detailed. Next the distribution of follicles and division of the feather coat into tracts was given. A brief mention of feather and feather coat patterns was outlined including the use of pigment and structure for colour.

²⁰aquatic birds - most rails, many alcids, loons and the dipper have compressed moult leaving the bird flightless

2.7. SUMMARY

Finally the chapter concluded with a discussion of feather growth and feather coat morphogenesis including outlining the various moulting stages. The methodology of feather coat morphogenesis presented in this thesis will use much of this information with particular emphasis on contour feather structure, arrangement (follicle distribution and orientation) and moulting.



Computer Graphics and Feathers

Modelling of natural phenomena is often challenging due to the complexity of the simulated phenomena. The desire to attain realism not only involves modelling many intricate and interdependent structures, but also simulating changes in structure, form and illumination as seen in the simulation of a variety of phenomena including waves [FR86], clouds [Nis98], smoke [SF95] water [Ree83] and plants [PL90].

Modelling feather coats is only a very recent topic of interest in computer graphics and modelling of feather coat morphogenesis is unexplored to date. The goal of modelling feather coat morphogenesis is to achieve realistic changes in feather coat appearance. To achieve this goal, two related modelling problems need to be addressed: the feather coat structure as well as the changes in the structure, while not restricting the use of a variety of methods for constructing the pattern, colouration or illumination of the coat. As well, these realistic appearances must be achieved in a manner that facilitates animation of the follicles or skin. This chapter will first overview general challenges in modelling natural phenomena. Next, an overview of common level of detail approaches in this context will be given. Then some methods used to model similar phenomena, in particular plants, hair and fur is reviewed. Finally, the current research on modelling of feather coats will be presented.

3.1 The Natural Phenomena Challenge

The most challenging aspect to modelling any type of natural phenomena is handling the sheer complexity of the objects and scenes to be represented. Not only do these scenes contain many objects with many

substructures and surface details, but often it is desirable to model the interaction between these objects, either to simulate the passing of time or the dynamic properties of the objects themselves. However, even with the recent advances in CPU and GPU power, it is still not optimal to model and render all the detail in many of these types of scenes. As well, some of the detail is only discernible from certain viewing distances, and thus does not need to be rendered at further viewing distances. There are exceptions to this rule, especially where the subsurface properties affect the overall illumination of the object's surface, which is still noticeable even at viewing distances where the subsurface structures are not. The common approach taken to this complexity management problem is to employ the use of multiple levels of detail (LOD).

By modelling and rendering the phenomena at various LODs, only the necessary amount of detail needs to be revealed, therefore optimizing computational efficiency. However, while reducing computation is desirable, revealing the various modelling methods to the viewer/user is not. Thus, not only must different modelling methods be employed at various viewing distances, but a method of smoothly integrating between the methods is necessary to prevent distinction of the models, especially when continuously decreasing the viewing distance. Finally, while a LOD approach overcomes some of the issues with these highly complex scenes, it does not handle the interaction and dynamics between objects. As mentioned in Chapter 1, if object interaction, dynamics and animation are required, it is best to include this as part of the model. The next section will overview common LOD methods used in modelling natural phenomena. Then an outline will be provided of phenomena similar to feather coats, namely plants, hair, fur and grass.

3.2 Level of Detail Modelling

At the highest level of detail every structure can be geometrically modelled. However, as the viewing distance increases, and structures occupy very few pixels, the need to model individual geometry is reduced. Thus methods for reducing the complexity of the model at these higher levels of detail have been used.

A very common method to reduce model complexity is to simplify the geometric model, reducing the polygon count. However, this is only adequate until changes in form and structure are noticed. A very common example of this is for a bumpy surface to be simplified to a smooth one. *Texture mapping* [BN76] was one of first methods for generating detail on a geometrically simple surface and can be used when an object is far enough away to simplify the computation and give the illusion of surface detail. While texture mapping can be used to simulate the visual effects of the bumps (macro or micro structure), the lack

of structure is noticeable at closer viewing distances first at the silhouette edges and then on the surface. Thus, techniques have evolved to blend between textures and geometry including mapping normals, or displacements to the surface as well as using reflection models (BRDFs) based on subsurface geometry to convey the proper illumination without modelling the meso¹ geometry.

The issue of blending between models of geometric representation to achieve multiple levels of detail has long been an open problem in computer graphics. As mentioned previously basic texture mapping works for surfaces of objects at a distance, but close inspection of the surface requires modelling the fine structure. Poulin and Fournier [PF90] addresses the problem as hierarchies of geometry with the model geometry at the highest level, followed by bump maps and displacement maps, then a texture that is based on the micro structure of the surface and finally the BRDF (*Bidirectional reflection distribution function*). Becker and Max [BM93] help to smooth the transition by redistributing the bump maps to have the same underlying energy distribution as the BRDF.

3.2.1 Texture Maps

Textures are evaluated at a particular point using local information and are mapped to geometry to add the illusion of detail to a geometrically simple surface. Textures are typically two dimensional and can be procedural [Per85, Lew89] and evaluated on demand by a program or stored in a “look-up table”. The textures are then projected in some manner onto the surface in a step called texture mapping [Hec86]. Originally only colour intensity was mapped, but mapping of the normals (*bump mapping*) to add the illusion of small height variation [Bli78] and mapping of surface offsets (*displacement mapping*) [Coo84] to alter the height at particular points is also now commonly used. Displacement mapping is slightly more complex than bump mapping, but yields better results, exhibiting self-hiding, shadowing and changing silhouette edges.

Where surface textures are not adequate, *volumetric textures* [Per85, Pea85] have been used. Volumetric textures map any point in space, rather than just the surface. Sometimes textures throughout objects are more intuitive (e.g. carvings, fur [KK89], textiles [GRS96]) and thus 3D textures or solid textures are easier to use. More recently these techniques have been extended for multiple scales [Ney95b, Ney96] and animation [Ney95a, Ney98]. The *Hypertexture* technique [PH89] is a volumetric modelling technique that implies surfaces where explicit surfaces have not been created, thus actually changing the surface of objects.

¹In this thesis macro is 1 cm. or larger, micro is 1–100 μm . and meso is 0.1–10 mm.

The rendering of these objects is done using volumetric methods. *Texel mapping* [KK89], maps an entire surface description (i.e., shading model) to the surface of the object in a bundle called a *texel*.

3.2.2 Texture Mapping Problems

Two main areas of research with regard to texture maps are the synthesis of the texture and the mapping of the texture. Textures can be digitally acquired (scanned), painted by an artist or procedurally generated. Scanned textures often lack quality and painted textures are very time consuming to construct, often require a great deal of skill, and lack photo-realism. Work has been done to generate arbitrarily large textures from small samples using perceptual rules to match features [HB95], probability to match frequencies [Bon97, PL98, EL99], or random noise to grow the sample [WL00], to ensure they are perceptually identical. Textures have also been synthesized procedurally from a set of initial parameters [WK91, Wor96].

Secondly, the texture must be mapped to the surface without distortion due to perspective, stretching, or aliasing. Traditionally this has been done using well-parameterized surfaces, solid textures [Pea85], two-part texturing [BKRS86] or manually placing the textures [LM94] on the object. Some methods approximate the surface with an easily parameterized object such as Walter and Fournier's use of cylinders to simultaneously grow the model and produce accurate texture [WF97]. Other methods warp the texture [Ara97] or warp and/or split the surface and "unravel" it [PB00] for correct texture placement. Still other methods attempt to generate a parameterization that minimizes distortion by optimally "flattening" user defined patches [BVIG91], minimizing energy [MYV93] or optimizing the homogeneous spacing of isoparametric curves [LM98]. Others have warped, rotated or scaled the pasted texture patches to fit most appropriately on curved, polygonal or subdivision surfaces [Ped96, PFH00, PB00]. Once an appropriate mapping is found the texture must then be filtered or re-sampled [Wil83, Cro84, Sch91] without causing distortion or aliasing.

Recent work to solve these fundamental texture mapping problems has pointed to using integrated methods to generate the texture on the surface while accounting for geometric distortions [DG94] or to generate the texture from geometry to simulate special surface properties or imperfections [WNH97]. Volumetric textures can be used when more than a surface texture is desired. Besides using 3D textures [Per85], textures can be painted directly on the 3D surface [HH90] or generated on the surface [Tur91, Wor96]. These techniques have been used to simulate weathering and mineral phenomena [DEL⁺99], but have been used more prevalently in conjunction with biological modelling [Tur91, FMP92, FLCB95, WFR98].

3.2.3 Reflection Models

In computer graphics literature lighting models are used to determine the colour of a surface and usually consist conceptually of two components: illumination and reflectance. Illumination defines how the light travels, and reflectance what happens when the light interacts with the surface. Illumination models are divided into global and local, where global illumination describes how the entire environment contributes to the illumination at a single point, and local illumination computes the illumination at each point using information local to the point (the surface and light attributes). In this thesis, the focus is on feathers and how their structures and pigments affect their colour, thus the focus is on local reflectance models.

Reflectance models can be categorized into models that simulate the interaction process of light and surface either physically or empirically (i.e. analytical models) and models that utilize actual data (e.g. measured or simulated) of how light interacts with a particular surface. Most of the previous work has been on the construction of analytical models. In either case a notion of the ratio between the incoming and outgoing light at a single point is described and is termed the BRDF (*Bidirectional Reflectance Distribution Function*).

Analytical Models

Analytical models range from those that are entirely empirically based simplified models for generating a “visual impression” of light transport to those that closely attempt to approximate the physical surface and light transport properties. Schlick [Sch94] and Glassner [Gla95] provide comprehensive surveys of reflectance models up to 1995. The three most simplified models include the Lambert model for diffuse surfaces (constant BRDF over all directions and polarizations), the Phong model to incorporate specular highlights and the addition of an ambient constant and the Phong-Blinn model that simplifies the computations of the Phong model. Other empirical models have been created for efficiency [Lar92], controllability [Str90] and to account anisotropic reflections.

More complex models attempt to more closely simulate physical optics, however, often many approximations are made, including using a small sample of wavelengths for lighting calculations. One of the earlier models accounted for variations in reflectivity by considering the surface as a collection of randomly oriented micro-facets. Then the reflectivity was defined by the probability distribution of the micro-facet orientations, the Fresnel term on individual micro-facets and shadowing and masking. The first model of this type was proposed by Cook and Torrance [CT82]. From this have evolved other models to account for

anisotropic reflections such as Poulin and Fournier's model [PF90] and Ward's model [Lar92]. The most complete model based on wave optics was proposed by He et al. [HTSG91]. Their model has been extended by Stam and Sun et al. [Sta99, SFDC00] to create iridescence effects similar to those found on CD-ROMS.

Non-analytical models

The non-analytical reflectance models include both measured and simulated BRDF models. BRDFs can be directly measured using *gonioreflectometers* to collect large numbers of point samples comprising the BRDF. BRDF's have also been measured less accurately using simpler CCD devices (digital cameras). Simulated BRDF models also generate samples of BRDF's, but use a *virtual gonioreflectometer* to generate the data. Knowledge about the micro-structure of the surface and an algorithmic process is used to compute values of the BRDF for samples of incoming and outgoing directions. Cabral et al.'s [CMS87] suggests a method to compute the horizon of a small number of directions at each point in a height field in order to improve bump mapping. Westin et al. [WAT92] extend this method using a Monte-Carlo ray-tracing technique to simulate almost any kind of microscopic geometrical structure. Becker and Max [BM93] redistribute the bumps maps in order to match the energy of the BRDF, helping to smooth the transition between bump maps and BRDFs. Once these values are either computed or measured they can either be stored in sampled hemispheres or spherical harmonics [Gla95, pp.753-756]. The idea is that the incoming light from each sample direction is used to compute the outgoing light for a hemisphere of directions. Then to apply the model the hemisphere with the closest incoming light direction is chosen and the hemisphere is oriented around the surface normal and grain, a set of table look-ups reveals the reflected light. Kajiya [Kaj85] was one of the first to use this type of model to create anisotropic effects.

Other methods have been proposed based on computing simulated BRDF data for how light scatters inside an object. Most of these methods are based on the idea of suspension of particles and often include the surface reflectance models for boundary cases. Models have been created to simulate Raleigh scattering where the particles are smaller than the wavelength of light and have been used for atmospheric modelling [Nis98] and to simulate Mie scattering where the particles are comparable to the wavelength of light and have been used for cigar smoke, dust and hair/fur [KK89]. Haase and Meyer [HM92] and Callet [Cal96] propose models and data based on the Kubelka-Munk model [Gla95, p. 770] for simulating light transmission through materials with selective reflection and selective absorption (i.e. pigments and dyes). They use their model for simulating the effects of paint on a surface. Hanrahan and Krueger [HK93]

present a reflection model for surfaces composed of a series of layers. By evaluating the transmission of light between boundaries and the volume scattering of light within the layers, physically based models for biological materials such as skin and leaves are created. More recently, a reflectance and transmission model specific to plants was proposed by Baranoski and Rokne [BR97] and a general complex subsurface scattering model for very thin surfaces such as hummingbird feathers was proposed by Schramm et al. [SGM97]. As well a method for modelling the subsurface scattering and internal reflection of human hair has been proposed [MJC03].

Current research on reflection models attempts to make analytical models more accurate and efficient by simplifying the computation [PH00]. Other researchers are working on generating simulated BRDF data more efficiently using graphics hardware [HDKS00]. Still others are working on generating methods to simulate a wide variety of micro-geometries with less information about micro-geometry itself [APS00]. Most recently, a model for skin colouration has been proposed using an inverse lighting technique and concentrations of melanin and hemoglobin in the skin [TOS⁺03] to create skin textures. As well, a model for subsurface scattering and internal reflectance to simulate the reflectance and colouration of dielectric hair fibres [MJC03] has been recently introduced. Finally, with the lowering costs of CCD's the use of actual measured BRDF data for various materials is also becoming more common [MPBM03].

Feathers are more than a pattern on a surface, they are known biologically to be cutaneous appendages that have movement independent from the surface to which they are attached. Thus, general texture mapping techniques are insufficient for modelling the complete feather coat. Of the analytical reflection models mentioned, the last two models for pigments and multi-layer surfaces seem most applicable to the feather colour problem. Of course both simulated and measured BRDF methods could be utilized based on the micro-geometry of the feathers. These characteristics lead to the conclusion that biologically based techniques which work directly with the object model and a physically based reflection model are most applicable and hence are further detailed in the next section. Other models for reflectance and rendering functions will be outlined in Sections 3.4.1 and 3.4.2.

3.3 Mathematical-Biological Models

The models most closely associated with feather coat models are biological models. One of the first researchers (Smith [Smi84]) in this area proposed creating complex images/models from a small number of

parameters - a principle called *database amplification*. This same process in biology has more recently been termed *emergence*. More specifically Taylor [Tay92] defined emergence to mean that the entire system gains complexity through interaction of simple components both in space and time and studying emergence of form and structure is known as *morphogenesis*.

In the analysis of biological morphogenesis two perspectives have been formulated. The first was identified by d'Arcy Thompson [Tho61] recognizing the study of growth rate as a precursor to the study of form, stating that form is determined by the rate of growth in various directions. The second was identified by Turing [Tur52] stating that pattern is not formed by growth, but rather by substances reacting and diffusing through the mass of tissue that form the biological entity. Below is a discussion of both perspectives. Other methods of classification were proposed by Prusinkiewicz [Pru93] including the embedding of the biological model in a discrete or continuous space (*exogenous* part of model), a constant or dynamic space (expansion and contraction), a static or dynamic relation between components (*endogenous* part of the model), parent/child (*lineage*) or lateral communication between components and communication through model or space.

Working with the spatial notion of growth and morphogenesis, Turing theorized that chemicals (morphogens) diffuse through the embryo causing different patterns to emerge in regions of different concentrations in a process called reaction-diffusion. Since then mathematical variants of this model have emerged [Mei82]. Meinhardt and Klinger [MK87a, MK87b] introduced an activator-substrate variant and notion of differentiation causing lateral and dichotomous branching. Murray [Mur81, Mur93] explored the reaction-diffusion process in relation to geometries and scale. Bard [Bar81] introduced cascading reaction-diffusion processes and region-based diffusion rates. Harrison [Har93] proposed a variety of specialized methods for generating a range of patterns. Other variants of reaction-diffusion include diffusion-limited accretive growth [Ede60, Mea86] and diffusion-limited aggregation [WS83]. Mechanochemical techniques proposed by Odell et al. [Ode81] use a similar diffusion mechanism, but instead of the substance diffusing on a purely chemical basis, the spatial arrangement is formed by mechanical forces acting on cells (possibly produced by chemical reactions) to produce many patterns especially structural. Many researchers have further developed these techniques to include the strength of forces between cells [Ost83, Ost83, WO90]. Most recently Savic [Sav95] used this type of model for simulating pattern formation in animal coats. Shaw and Murray [SM90] propose a combination of traditional reaction diffusion and mechanochemical models for formation and location of feather follicles on the common coot. This will be discussed further in

Chapter 4. Cellular automata systems [TM87] also formulate a notion of spatial arrangement and can be viewed as the discrete space and time counterpart of reaction-diffusion models. Young's model [You84] is similar to Turing's yet more local. Ulam [Ula62] proposed a model for branching structure and Cocho et al. [CPR87] presented a variation of Toffoli's technique [TM87] that assumed an initial small number of arranged "clonal" cells which act as "seeds" for elements of the final pattern and advance according to the automaton rules. Both reaction-diffusion and cellular automata methods only allow growth on the entity's boundary. Gottlieb [Got93] proposed a method of growing the entity internally by subdividing the space upon each division of cells.

Structural models, unlike spatial models, describe how components of the model develop. L-systems describe the development of linear branching structures from discrete components [Lin68] using formal language theory. The development can be controlled by lineage (context-free or OL-systems) or by endogenous interaction (context-sensitive or IL-systems). L-systems can also model changes in shape throughout development. The biological entities formed using L-systems can grow at the extremities (sub-apical growth) as well as in the internal parts (elongation of internodes). However, changes in relative position of the components make it difficult to have exogenous control. Map L-systems [LR79] use maps to enable the description of structures with cycles. A group of cells represent a map, and there are map rewriting rules to act on the cells to control their properties. Lindenmayer originally used this method for modelling small biological entities such as algae.

As Prusinkiewicz notes [Pru93], spatial models are excellent at capturing information flow through the medium, where structural methods do not encapsulate this information flow. However, spatial models are limited in capturing the spatial changes of the medium and the biological entity within it (growth is typically limited to the boundary), while structural models encapsulate the expansion of the entire structure. In either case, models have developed that account for interaction of the model with its environment. For instance, Ulam's cellular automata method tried to account for collisions between branches by not allowing growth to take place in an already occupied cell. Also, context sensitive L-systems, have often been used to adapt the development to the environment. Some results can often be generated by variants of both methods, sometimes seemingly opposite techniques. For example the Maltese cross pattern can be generated by both cellular automaton with collision checking and context-free L-systems.

3.4 Relevant Computer Graphics Models

Many of the biological models previously mentioned have been applied in the computer graphics domain, sometimes used as a tool for biological research. Prusinkiewicz [Pru93] gives a good overview of various biological and mathematical theories of emergence and outlines work in biological development up to 1993. A briefly summary of this work and other work related to the problem of feather coat morphogenesis is presented below.

Some computer graphics models for natural phenomena are closely integrated with their biological model, such as Turk's model [Tur91] for texture generation for Turing original reaction diffusion model and Meinhardt and Klinger's [MK87a] variation, Fowler et al.'s [FMP92] use of Meinhardt and Klinger's model [MK87b] [Mei95] for seashell pigmentation and Walter et al. [WFR98] use of Choco's models for creating a variety of large cat (e.g. cheetahs, leopards) and giraffe coat patterns. Others investigate the complications of extending the model in a 3D domain [Gre91, Kaa92, CR95] and interacting with the 3D environment, such as climbing plants [AK88, Gre89] and the development of the path of root growth [Gre91]. Earlier work on structural models developed models based on growth rules [Smi84], geometrical information and features [AK84], physical or biological rules [dREF⁺88] and features [Hol94] mainly for modelling plants. More recently most of the work in the structural model domain has been concentrated on the use of Lindenmayer's *L-systems* for simulating branching structures. Prusinkiewicz and Lindenmayer [PL90, PLH88] added geometry to Lindenmayer's original models and shifted some aspects from the discrete to continuous domain creating parametric *L-systems* and differential *L-systems* which were used for simulating plant growth over time [HP96, HRD95] and computer animations [PHM93]. *Context sensitive L-systems* were used by Prusinkiewicz et al. [PJM94] to alter the external shape of plants by allowing a mechanism for pruning, for combining exogenous and endogenous control. and incorporation of tropisms. Prusinkiewicz claimed [Pru93] that *L-systems* can create plant images similar to Aono and Kunii and Reffye et al. [AK84, dREF⁺88] and that stochastic *L-systems* can create the models for grass similar to those created by Reeves and Blau [RB85].

Noting seemingly orthogonal characteristics of spatial and structural methods, there has been some work done on combining these methods particularly for use in simulation or animation. The presence of cycles in map *L-systems* (*OL-systems* [PL90, p. 146]) presents problems for creating geometric models because of the inability to assign metric properties locally. Fracchia et al. [FPdB90b, FPdB90a] solved this problem by proposing a physically-based solution using physical cell properties such as osmotic pressure

and wall tension to iteratively stabilize the mechanical system. They also extended this idea [Fra95] to use context sensitive L-systems. Fowler et al. [FPB92] describes a method for arranging primordia on a surface (of plants) based on both developmental rules of structure (path of primordia growth) and physical properties such as collisions. Fleischer et al. [FLCB95] proposes a method for generating structures based on a spatial distribution of “cells”. The simulation distributes spherical cells based on user defined cell states, extra-cellular environment and interactions and cell programs while constrained to the surface. The final distribution is then passed through a particle-to-geometry converter, to construct the shape and appearance of the cells. While they produce organic-looking textures on model surfaces, their assumption of spherical cells for the simulation and use of a differential equations as an interface can be problematic in attaining a desired result.

Feather coat morphogenesis involves the development and management of an approach with a potentially large number of fine geometric primitives and some interactive modelling. The coupling of development and complexity is similar to many plant systems and the coupling of movement of the elements and complexity is similar to modelling of hair and fur. Structurally, feathers are similar to the branching structures of plants and biologically, feathers are similar to hair, fur, and scales - all being cutaneous appendages. While the structure of hair and fur differ from feathers, distribution and rendering are similar. And while feather structure is similar to plant structure, the level of intuitive design and need for styling is different. Rendering techniques used for some hair models and plant models may also be applicable to feathers.

3.4.1 Plant Models

Many of the plant models have been mentioned above. Two others are worth mentioning here a rule-based model [LD96] for interactively modelling and animating branching botanical structures and modelling of a plant ecosystem [DHL⁺98]. Lintermann and Deussen [LD96] use a principal set of components and rules, that the user combines and modifies graphically. They base their model on Weber and Penn’s [WP95] and can simulate tropisms, integrate free forms deformations (in order to allow the user to apply overall changes), animate any parameter of the model, iterate objects algorithmically, and incorporate key-framing which they claim cannot be done with conventional string rewriting systems. The second plant modelling work involves modelling of an entire ecosystem of plants [DHL⁺98]. The complexity of the resulting system, the specification of the initial parameters, and the simulation of development closely parallels the feather morphogenesis problem. The user interactively or procedurally defines the initial plant distribution and the

parameters of the various species of plants. The simulation then runs, determining from the parameters which plants die, grow, are propagated and by how much. While plant models provide a starting point for modelling complex, evolving models, even if they were adapted to model feathers, they would lack many of the features needed for modelling feather coats and feather coat morphogenesis as well as the intuitiveness that a model specific to feathers coat morphogenesis would provide.

3.4.2 Hair and Fur Models

Much recent work in computer graphics has been done on the modelling of hair and fur. The previous literature on hair and fur will be discussed here in the context of distribution, structure, interaction, dynamics, and rendering.

Early methods of generating hair or fur did not suggest geometry, but rather tried to use reflectance maps or texture mapping to convey the impression of combed hair or fur textures. Reeves and Blau's [RB85] created the impression of 3D lines (grass) using motion blur and particle systems, Miller [Mil88] created the impression of aligned hair (anisotropic surface) using evenly spaced cylinders to create a pseudo-reflectance map and Perlin and Hoffert [PH89] used volumetric textures to create furry tribbles. Kajiya and Kay [KK89] have produced one of the best models for fur using texels spread over the surface and a volumetric reflectance function to create the illusion of fur with self shadowing of the strands. Other more recent work in designing hair and fur textures has used exponential functions to control the shape, size, perimeter, and density of the haired region [Bid95] as well as utilized LIC - Line Integral Convolution [MKKI00].

Later methods of modelling hair attempt to explicitly model the hairs or some subset of them. Miller [Mil88] was one of the first to suggest modelling a set of cylinders. Rosenblum et al. [RCT91] use cylindrical segments with connecting hinges and mass/springs for dynamics. Watanabe and Suenaga [WS92] model wisps of hair as a connected set of trigonal prisms for styling purposes. Each wisp has a density representing a group of hairs whose dynamics are taken from the wisp trajectory and are rendered as trigonal prisms with no mention of collision detection. Anjyo et al. [iAUK92] represents hair as a set of polylines and simulates dynamics and curvature using the cantilever approach, a set of forces and parameters for stiffness. Hair to head collisions are detected using an ellipsoid to approximate the head. Daldegan and Thalamann [DTKT93] model hair by interactively placing characteristic hairs in varying densities, orientations and lengths. They animate hair by wisps of polylines with mass/spring weightings and hair to body collisions are detected using a cylindrical approximation for the body. The hair is rendered using ray-tracing.

Gelder and Wilhelms [GW97] model fur as an under-coat and over-coat and with a variety of methods from straight line fur, NURB-curves and polyline segments. They introduce a method of distance based culling to reduce the number of hairs per triangle to maintain the fur look at further distances. More recently, Iones et al. [IKVZ00] provide a method for modelling and rendering fur and hair using control hairs (a set of polylines) whose interpolated dynamics are used for the rest of the hairs. A volumetric reflectance model similar to [LTT91] is used for rendering. Leblanc et al.'s rendering model sorts hairs by distance and then uses pixel blending to approximate density of the volume. They also use a shadow buffer for each light source to produce shadows and self-shadows. Koh and Huang [KH00] animate strips of human hair in real time. Their animation and collision model is adopted from [DTKT93]. They use alpha blending along the strip to give the illusion of hair. Hadap and Thalmann [HMT00] interactively style hair by placing a flow field around the polygonal model of the head and using positioned sources of flow to prevent body-hair collisions and positioned sources, sinks and vortices for styling. They also allow clumping to improve realism and use LeBlanc et al.'s volume rendering method. Finally, Bruderlin [Bru98] introduces a method of interactively specifying clumps of fur following "hair tracts" to simulate wet or clumped animal fur.

The distribution of hair or fur follicles is known biologically to be much the same (although slightly less organized) as feathers. However, a uniform random distribution is often sufficient, due to the symmetric and simplified structure of hair and fur. Kajiya and Kay [KK89] state that fur is typically distributed similar to a Poisson disk distribution² which was also used by Goldman [Gol97] to reduce hair overlap probability of sufficiently short hairs. Rosenblum et al. [RCT91] interactively define a follicular map and uses a spherical projection to find the relative location on the head. The follicle map can be mirrored to reduce computation and the follicles "jittered" to remove unwanted patterns in placement. Watanabe et al. [WS92] interactively distributed wisps of hair over half of the individual triangles of the head model. Anjyo et al. [iAUK92] describes a model capable of styling hair by interactively defining the length of hair and a pore region on an ellipsoid representing the human head in order to prevent complex collision computations. Gelder and Wilhelms [GW97] randomly distribute the hair (both long and short hairs) over each area-weighted triangle of the polygonal model given user defined density and quantity.

Finally, a range of rendering methods have been proposed. Some methods based on volumetric textures were mentioned previously. Goldman introduced a probabilistic method for rendering fur attributes such as shine and soft illumination using an opacity function without actually modelling the fur.

²no two samples are closer than a fixed distance radius around the sample [Gla95].

These results are good for medium distance viewing. Much of the current work in hair rendering attempts to improve the reflectance model by incorporating efficient methods for doing good self shadowing of hair [LTT91, KN00, LV00]. Other methods achieve real-time rendering rates [Len00] by building up a concentric shell of volume textures, pre-filtering across level-of-detail, and procedurally introducing geometry (similar to the way Kowalski et al. [KMN⁺99] introduce geometry). And most recently, a model for subsurface scattering and internal reflectance to simulate the reflectance and colouration of dielectric hair fibres [MJC03] has been developed.

3.5 Feathers

Dai et al. [DSC95] was one of the first to attempt modelling individual feathers. Very recently, some further research on modelling and rendering of feathers has been published. Previous to this most computer generated feathers and feather coats were constructed manually by an artist typically in a commercial environment. Now, there are models for both photorealistic and non-photorealistic feather coats.

3.5.1 Commercial Productions

Many of the computer generated birds seen in commercial productions until recently used very simplistic methods for feathers and such were viewed from far away or with a lot of motion blur. Some of these include the bomber WWII pigeons in the Nissan commercial (1998 by ILM) and the penguins of the Canada Dry commercial (1999 by MarteStudio) or the Pelicans in TriStar Pictures's *Jumanji*. In Pixar's *A Bug's Life* the bird was meant to give an impression of realism with a cartoonish overtone. Some may speculate that the feathers on these characters were tedious, labor-intensive, hand-painted textures from artists. In all of the above mentioned productions the focus was almost always on something other than the birds.

3.5.2 Dai et al.

Dai et al.'s paper [DSC95] was one of the first techniques that attempted to model feathers. They model the initial structure of the feather and provided a method for constructing textures with feather-like pattern elements such as eye-spots and river-stripes. They use an iterative approach using complex maps to both create the feather texture and the feather structure.

The texture synthesis focused on simulating two particular characteristics of feather patterns common to Galliformes: eye spots (found on the tail feathers of peacocks) and river stripe. They use a complex polynomial map and Newton's method as an iterator to converge to the roots in a maximum number of steps or within a threshold. The convergence is colour mapped using distance with the rational roots forming the eye spots centres and the complex roots forming the river stripes. The resulting pattern can be tailored by initially defining the pattern window and complex roots within it.

The feather structure is formed by interactively specifying parameters such as barb length f_{bl} , barb angle f_{ba} , angle variation during barb propagation f_{bpa} , length variation between two barbs f_{rl} and number of barbules f_{bn} . The feather shape is driven by the next propagating orientation defined as $\theta_{po} = f_{bpa} \times f_{ba}$ where $f_{bpa} = au^2 + bu + 1$, where $0 \leq u \leq \pi$. They then interpolate the values of a and b linearly with time between two user defined bounds a_0 to a_1 and b_0 to b_1 to produce a set functions which define the smoothly altering angles in barb propagation. A fixed f_{ba} means that the angle of the barb with respect to the rachis is constant for all barbs. They used a generalized cylinder whose polygonal tessellation was dependent on view distance for the rachis, line segments for the barbs and triangles for the barbules. They adjust the polygonal resolution of the rachis according to camera distance. They then texture map the 2D planar feather using their unlimited texture resolution previously produced. While Dai et al.'s model provides a particular starting point for feather modelling it does not address any of the issues of feather coat morphogenesis or handle collisions or skin movement.

3.5.3 Non-photorealistic Feathers

In the animated short *For the birds* from Pixar, the focus was the birds. The birds were portrayed with realistic behaviours and motion including individual feather movements, but as very cartoony characters. The feather coat was composed of individually modelled feathers, not a texture-mapped surface and there was a great deal of animation of the feathers [Rob01]. Robertson [Rob01] states that "the biggest technical challenge for the animators...were in making the contacts between the birds work and in animating the feathers".

The producers recognized that the feather realism "just added that element of birdiness" [Rob01] and hence was worth the extra computation. The feathers were very short, planar, texture-mapped surfaces. In order to increase the element of *birdiness*, geometry was added to soften the shingle-like feathers to make certain areas appear more downy and fuzzy and to create hairy edges and silhouettes. To create this

look they added little spheres mapped with hair and transparency textures in choice locations. In addition to making the feathers look soft, the producer also requested that they have shine since the scene took place on a hot summer day. Robertson's article does not disclose any more details about this topic.

Another technical challenge that presented itself with this work was the ability to fix bald spots and to handle contacts between birds. When birds' faces were animated, the surface of the bird often stretches causing bald areas in the feather coat. Pixar fixed this by simply creating copies of local feathers and randomly distributing them within the bald spot. Since much of the short relies on the birds looking like they are "squished together" interpenetration of feathers and/or bodies was not an option. Pixar handled this issue by creating disc shaped collision detection widgets that were placed between the birds that were ultimately shaped and animated by the animators. As far as is known most of the animation of the feathers and birds was done manually by the animators. The animators initially worked with guide feathers that were geometrically simpler and smaller than the final feathers. As Robertson states each bird had 2873 feathers.

3.5.4 Concurrent Photorealistic Feathers

In parallel to this thesis work, there has also been work on creating photo-realistic feathers [FW01] and feather coats [EAB⁺02] [CXGS02]. Since modelling realistic feather coat morphogenesis is the goal of this thesis, this previous work is most applicable. Two of the main contributors in this area are *Sony Pictures Imageworks* and *Microsoft Research Asia*.

Sony Pictures Imageworks

Sony has been working on feathers over the past few years for their 2002 release of *Stuart Little 2* which starred a mouse, Stuart and two birds: an American Goldfinch as the heroine (Margalo) and a Peregrine Falcon as the villain. The movie included close-up shots of animated birds and moving feathers. From previous movies they had developed an extensive hair modelling and rendering pipeline and parts of this pipeline were used for their feathers.

The individual feathers were modelled using NURB patches. They have a few hundred parameters to control the structure and rendering properties of each feather [EAB⁺02]. This included things like applied texture maps, length and curve of the feather, breakup or splitting and transparency. They distributed the feathers over the surface of the model using the repulsive force mechanism of Witkin and Heckbert [WH94] with an alteration of being able to vary the density of the follicles over different regions of the surface.

The feather direction, orientation, twist and grooming were all controlled using hundreds of key curves. Interpolation between these curves generates the rest of the feather coat (approximately 3000 feathers). By manually editing these curves, the feather coat is able to be groomed. Feather-to-feather and feather-to-body collisions are resolved manually using these key curves in a static model position.

One of the biggest hurdles tackled by the Imageworks team was to prevent interpenetration of the feathers during animation. It was necessary to have feathers slide over one another without piercing each other or ending up in the wrong order. To handle this they developed a one and two dimensional volume deformation technique. Since the key curves were animated with the model, there were possibilities for incorrect feather directions or feather interpenetrations with each other or the model. The volume deformation information stored a height displacement for each control vertex from the surface and the closest surface location, so that as the model is animated the feathers stayed aligned with the surface of the skin. While this solved interpenetration with the skin it did not solve interpenetration with neighbouring feathers, so a second set of volume deformation information stored how close adjacent control vertices were to each control vertex of each feather.

They then enhanced the realism and “cuteness” factor of their results by adding hairs in-between feathers where required and making manual adjustments (including plucking feathers, manually combing, hand animation and blending static effects). The feathers were shaded using the RenderMan®³ shading language employing an anisotropic lighting model and some extra back-lighting effects. The feather textures were generated both procedurally and hand painted. Each bird required about 200 key curves. Margalo was complete with 3,082 feathers and approximately 70,000 hairs and the Falcon had 3,241 feathers and around 7,000 hairs.

While the results of their technique are good, a lot of manual intervention was required as well as the addition of hair. Using NURB patches to model their feathers restricts the type of feather they can model (e.g. they cannot create down which perhaps explains the use of hair) and their model does not allow and was not intended for, modelling of feather coat morphogenesis.

Microsoft Research Asia

Chen et al. [CXGS02] recently published a method for modeling and rendering of realistic feathers. Their method was based largely on the fact that the feather is a natural branching structure and hence used a

³RenderMan is a registered trademark of Pixar

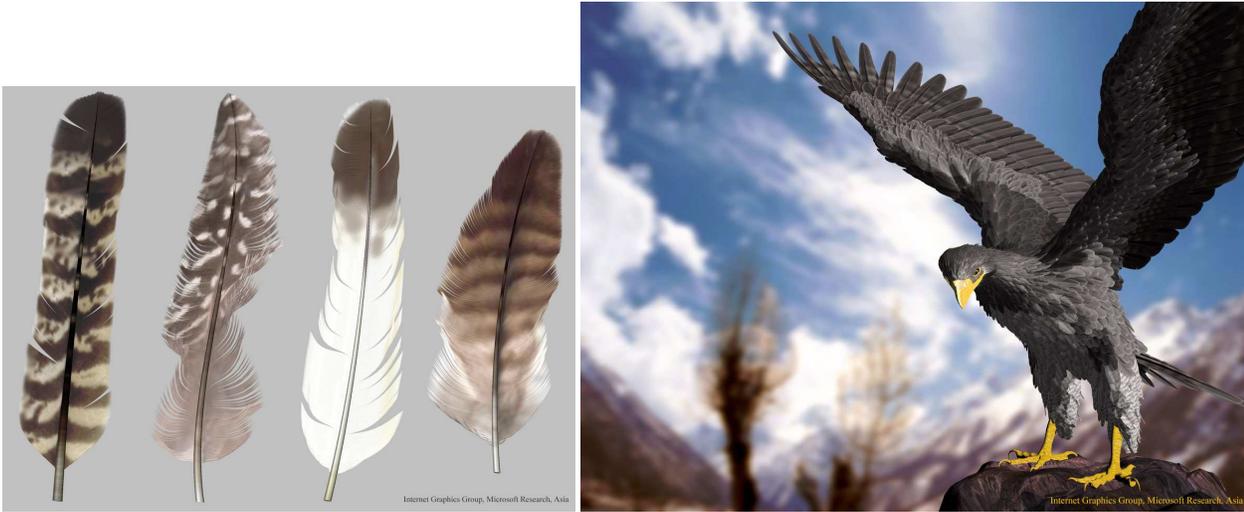


Figure 3.1: Two images from Chen et al.'s [CXGS02] work on modelling and rendering feathers. **Left:** Some example feathers. **Right:** A feathered Eagle. These images were obtained from the authors and used with their permission.

method common to computer graphics for modelling plants, L-systems. A feather was modelled using a set of five curves to define the shape of the feather: rachis curve, left and right barb curves and left and right outline curves. They then assume that the shapes of all barb curves on the same side of the rachis are identical except for their lengths. The structure of their feather is described by a set of context-free L-system productions of the form:

$$id : pred : cond \rightarrow succ$$

They use a parametric L-system as follows:

$$\begin{aligned}
 w &: R(0) \\
 p_1 &: R(i) : i < N \rightarrow [B_L(i, 0)][B_R(i, 0)]R(i + 1) \\
 p_2 &: B_L(i, j) : j < M_L \rightarrow B_L(i, j + 1) \\
 p_3 &: B_R(i, j) : j < M_R \rightarrow B_R(i, j + 1)
 \end{aligned} \tag{3.1}$$

where N defines the length of the feather as well as the density of barbs on each side of the rachis, while M_L and M_R define the lengths of the left and right barbs respectively. This means production p_1 grows a small bit of the rachis and then productions p_2 and p_3 grow the left and right barbs respectively, with production $R(0)$ constructing the entire feather. They then introduce forces into this system to simulate splits in the feather vane. The splits are placed in locations where the accumulation of a random set of forces exceeds some threshold and then the barb is rotated by a random angle in the direction of the resulting force.

Their feathers are rendered using a precomputed bidirectional texture function (BTF) of sample geometry of the barb/barbule structures. They ray-trace this geometry offline and use the resulting BTF as a texture to alpha blend with the pattern texture of the feather.

The feather coat is generated by first re-tiling the polygonal model using a variation of Turk's algorithm [Tur92]. Again this is similar to the method used by SONY Pictures Imageworks in that they use repulsive forces on the surface to determine the follicle distribution. They alter Turk's algorithm slightly to vary the density based on feather size rather than surface curvature. They use a Gaussian radial basis function to determine the area of influence and use Dijkstra's shortest path algorithm to compute distances on the surface. The user is able to control the spatial extent and the weight of each basis function.

Once the follicle locations are determined, the user places *key feathers* and defines a growing direction at each of these locations. From the key feathers and their growing directions, interpolation computes feathers and growing directions at all intermediate locations. However, these are only the initial growing directions. The final growing directions are determined using a recursive collision detection algorithm. The collision detection is computed using a simplified version of the feather geometry. A Voronoi region is determined around each feather and only the feather at the vertices of each Voronoi region are considered candidates for collision detection. To further simplify the computation, only those feathers lying below this feather are used to compute collisions. When a collision is detected the feather is rotated slightly toward the surface normal. This process is repeated recursively until no collisions are detected.

Again, the results they obtain are quite good as shown in Figure 3.1. However, due to their chosen approach in modelling individual feathers, the types of feathers that can be modelled are limited (i.e. down cannot be modelled) and the internal shape of the feather is limited because only one curve defines the shape for all barbs on one side of the feather. The use of a recursive collision detection algorithm is fairly time consuming (30 minutes off-line for a bird with 3500 feathers) and because it is based on simplified geometry it is not exact. Finally, they too do not provide any mechanism for simulating feather coat morphogenesis.

3.6 Summary

This chapter summarized previous work with aspects similar to modelling feather coat morphogenesis. The interaction placement and simulation of a large complex systems such as Deussen et al.'s ecosystem [DHL⁺98] parallels the complexities of the feather coat problem. Also the interactive modelling of

3.6. SUMMARY

plant systems using rules and icons [LD96] and interactive modelling of hair styles using fluid flow streamlines [HMT00], characteristic hairs [IKVZ00], wisps or clumps [WS92, Bru98] and parameters [DTKT93] is comparable to the control and interaction required in feather coat modelling. As well, the reflection models of Hanrahan and Krueger [HK93], Schramm et al. [SGM97] and Haase and Meyer [HM92] simulate many of the optical effects found in feathers. However, none of these models were designed for feathers and using them to model feathers would be at minimum awkward and difficult. Also, there is no provisions for handling the complexities of feather arrangements and morphogenesis.



Problem Scope, Goals and Approach

The modelling of feather coat morphogenesis is a multi-faceted problem as shown in Chapter 2 and is a relatively unexplored area of computer graphics as shown in Chapter 3. In order to solve a problem with these attributes the scope of the problem must be well identified. First, a set of model expectations need to be made. This will identify what the final model should be capable of and what aspects of the model are most important. Second, a set of assumptions will be made in order to reduce the scope of the problem to a manageable size. Finally, this problem will be subdivided into addressable, possibly non-distinct elements. Each element of the problem will also have a defined scope. In the next three chapters the approach used to solve the main elements of this problem will be discussed.

4.1 Model Expectations and Goals

As already outlined in Chapter 1 the goal of this work is *to design a model of feather coat morphogenesis for the purposes of animating the follicle surface or skin*. Thus change in the feather coat must be simulated, including movement of the feathers, as well as loss, re-growth and consequential change in form and pattern. Animation of the feather follicles or feathers is necessary for creating the appearance of change in overall shape or form. Thus, the goal is to have the feathers respond appropriately given movements of a skinned surface, but not to determine the skinned surface based on animation of the skeleton or appendages of the bird, which is a separate area of research.

To ensure unconstrained creational possibilities, the process of modelling feather coat morphogene-

sis must work for a reasonable range of topological models without specialized specification of the model. Thus, not only would it be possible to place feathers on models of birds and other natural objects, but possibly on any closed three-dimensional object. Also in order to ensure that this approach can be used easily with commonly available packages the model is assumed to be a closed set of polygons with normals and no special properties. Any other unknown attributes of the model will be obtained as parameters or inferred from the model itself. This allows for a wide variety of packages to be used to create the initial input model to the methodology.

In computer graphics it is important to minimize the amount of time needed to create an object or character, whether computation time or user time. Lessening creation time increases the possibility of modelling and/or animating in real-time, allows for more objects or characters to be included in a scene or more complicated objects or characters. Time or computation that is only needed once in the initial stages of designing the object or character is normally nominal compared to the time needed to create each frame of an animation or simulation. However, the goal of this thesis is not static once designed. In fact the feather coat is composed of many individual segments that can either move or change appearance. Thus, it is plausible that the model changes continuously (possibly frame to frame as in simulations), hence time needed to make changes to the model can largely impact the resulting rate of creation. Thus, the model must enable efficient change and movement of the feather coat.

Finally, this efficient change and movement must also be controllable. As mentioned in Chapter 1 there is a trade-off between automating a potentially time consuming process and allowing for flexibility of the final result. Since the number of structures (feathers) that will typically be used is on the order of hundreds or thousands, individual control and positioning of the feathers is unrealistic. Thus, the process must involve some semi-automatic means of controlling the feathers in a fashion that is not cumbersome or time consuming.

Previous and concurrent work on feathers provides some initial results, however lacks automation, flexibility and a means to model feather coat morphogenesis. While Sony Pictures Imageworks [EAB⁺02] uses curves and surface patches for their feather models and styling, their models were very parameter (manually) intensive (i.e. on the order of a couple hundred parameters) and many of the arrangement and orientation issues needed to be resolved manually. However, their results are stunning. They did not attempt to address feather coat morphogenesis. Work done recently by Microsoft Research Asia [CXGS02] provided a slightly more automatic method for both modelling and rendering their feathers, however lacked the

flexibility to model a wide variety of feathers types as well as lacked a means to model feathers grown as a result of feather coat morphogenesis.

4.2 Assumptions and Limitations

Optimization of processes and models often result in compromises. For instance, incorporating fine detail commonly requires more computation thus trading-off amount of detail and process efficiency. Thus, in designing the model certain assumptions must be made in order to allow for optimizations in other areas. These assumptions and limitations will be specified with respect to the feather coat morphogenesis problem in the context of the principles of natural phenomena models as outlined in Chapter 1.

4.2.1 Level of Detail

The level of detail desired can affect the entire design of the model. For instance if only a very specific limited level of detail is needed then many optimizations can be made to take advantage of this fact, and thus increase performance. However, if the level of detail desired is variable, often not only are these assumptions impossible but much care must be taken to ensure the preservation of multiple levels of detail. A naive solution to modelling feathers would be to model the geometry of every structure right down to the substructure in barbs and barbules to ensure realistic results. For a few feathers this may seem like a reasonable approach, but for an entire feather coat or possibly multiple feather coats, this much geometry would require an unreasonable amount of resources (e.g. computation time, memory, and processor ability) making this approach seem less than adequate.

The goal of this work is to simulate the appearance of feather coat morphogenesis. The level of detail chosen for this work can be thought of as the feather. At a scale where an individual feather is not discernible from the rest of the feather coat some modified form of a texture mapping approach would probably be a good trade-off between the quality of results and computational efficiency. Thus the concentration is on meso-scopic structures of the feather, where meso-structures have size 0.1–10 mm. While modelling microscopic substructure (i.e. less than 0.1 mm., barbules and barbicels) is necessary for creating the colouration, it is not always visible to the naked eye and is not required in the final model, but may be modelled initially and used in the rendering stages. So the targeted level of detail for this work will be for mid to close range (excluding microscopic) viewing of the feather coat for a small group of birds. In this range, texture map-

ping is not adequate and feather geometry is necessary because individual feather movements and structures are noticeable, but methods that derive texture from geometry [DG94] may be used at the boundaries of this viewing range.

4.2.2 Object-Specific Models

As mentioned in Chapter 1, using a generic technique for modelling the characteristics of a surface is sometimes not adequate, but precise geometric modelling of the entire object causes a complexity explosion. Modelling of feather coats and more so, feather coat morphogenesis, is exactly one of those problems. Thus, methods that take advantage of the specific attributes of the structure, such as regularity and knowledge of the how the structure develops can be used to construct an object-specific modelling technique. This approach will be used in order to optimize the results and the efficiency of the technique.

One downside of creating an object-specific model is that it is often limiting in its applications. The modelling of feather coat morphogenesis has two main components: modelling of feathers and modelling of the feather coat. While these two components are not distinct, they can be designed separately to allow flexibility in the methodology. Modelling of the feathers involves developing a parameterization for the feathers that allows interpolation between types and ages of feathers. Modelling of the feather coat involves modelling the placement, orientation and arrangement of the structures in order to form the feather coat. Although, the feather coat modelling handles the distribution and movement of the feathers, the feather structure could theoretically be substituted for any object with similar form and movement or motion like blades of grass, leaves or any cutaneous appendages (e.g. hair, fur, scales). Thus, while the model will be feather specific, incorporation of other substructures is plausible.

4.2.3 Emergence, Control and Efficiency

As already mentioned, an emergent system is one where complex structure results from a relatively small number of parameters. While it is desirable to minimize manual modelling and maximize efficiency and simplicity of the model, it is also necessary to maintain flexibility, control and predictability of the results. Control does not mean that the user specifies a large set of parameters but rather that a user is able to create what they desire.

In order to facilitate user control and efficiency as well as emergence, a minimal set of parameters is specified at the beginning of the simulation. During the simulation an automatic process runs allowing

the user to stop and restart the simulation at any point. The development of the final structure is automatic based on the model at the particular time with the particular set of user specified parameters.

4.2.4 Input and Output of the System

As previously mentioned the input is assumed to be a set of polygons (possibly with normal information) and a set of user specified parameters to tailor the model. One assumption made about the model is that the surface area is proportional to the number of follicles or size of follicle region. This means that if polygons grow (shrink) in size due to movement of the model, this corresponds to the skin stretching (relaxing) and the follicles will simply be spread (compacted) apart (together). The addition of polygons would correspond to the addition of a patch of follicle-less skin which is unrealistic and therefore will not be considered in this model. The second assumption is that the model's surface has some maximum curvature with respect to the polygon resolution. Since sharp edges do not exist on the surface of animals the skin model is assumed to have bounded curvature. Other than this restriction, the model must be robust enough to handle models of any topology. The only other surface abnormality to handle, is skin creases or folds. These will be handled as they are in the real world; if the skin has a cutaneous appendage attached to it and the skin folds, the appendage neither disappears or shrinks, but simply stays attached at the same location on the skin.

The feather arrangement algorithm is automatic. The feathers are assumed to always lie naturally given the body axis. They will adjust their direction and orientation to accommodate movement and situations such as creases and change in curvature. Not only do different tracts have various distributions (see Chapter 2), but different distributions may be desired by the modeller to create various effects. However, the follicle location is fixed to some location on the skin, which may or may not move in relation to the skeleton, but this is outside the scope of this work.

Finally, the output of the system is meant to be used with any commercial rendering package. As a post processing stage to the model, certain physical surface characteristics may be converted to typical rendering parameters to be used with any "off-the-shelf" renderer.

4.2.5 Other Necessary Model Attributes

In addition to level of detail, object specificity, control v.s. efficiency and input/output characteristics of the approach there are two other attributes that must be considered in designing the model: inclusion of time and use of stochastic elements. Since the goal is to model a changing result the inclusion of time in the

methodology is imperative. Many of the parameters will vary continuously, requiring a mechanism of both tracking and storing the variation as well as interpolating between variations.

4.3 Problem Segmentation

The goal of this research is to simulate the changes in appearance of the feather coat. The components of feather coat morphogenesis that most contribute to changes in appearance have been identified as: feather structure, feather arrangement and feather alterations (see Chapter 2). The feather structure identifies the type of feathers present and contributes largely to the overall appearance of the coat. The feather arrangement defines both the shape (form) and pattern and so also contributes largely to the appearance. Finally, feather alterations define both when and to what extent the feather structure (evolution from pin feather to final state) and arrangement change over time due to growth and movement of the skin and thus, is also necessary in capturing changes in appearance. This problem can be thought of as being composed of many sub-problems (i.e. elements) belonging to these three main components. Each element can be thought of as either static or changing, in the sense that the static elements do not change once formed (e.g. a feather's structure or colour) and the others are continuously evolving (e.g. feather growth). A rough categorization of elements is shown in Figure 4.1.

These three main components are non-distinct in the context of the feather coat morphogenesis problem and must be considered in this context. The feather structure (and colour) coupled with feather growth forms individual feather patterns. This coupled with the feather arrangement constructs the feather coat pattern. And finally, the feather coat pattern coupled with the feather alterations component forms the morphogenesis of the feather coat. The three main components and their scopes are discussed below.

4.3.1 Feather Structure

The first component, feather structure, involves modelling the substructure and its variations of the feather. Since these variations define different feather types (see Chapter 2), this component involves modelling of different types of feathers, which requires creating a parameterization that defines the physical structure, shape, size and curvature of each feather type. The parameterization must allow for interpolation between the feather types and feather ages by interpolating elements of the parameterization. Colour of the feather must also be modelled, but development of individual feather patterns is a topic of research in developmental

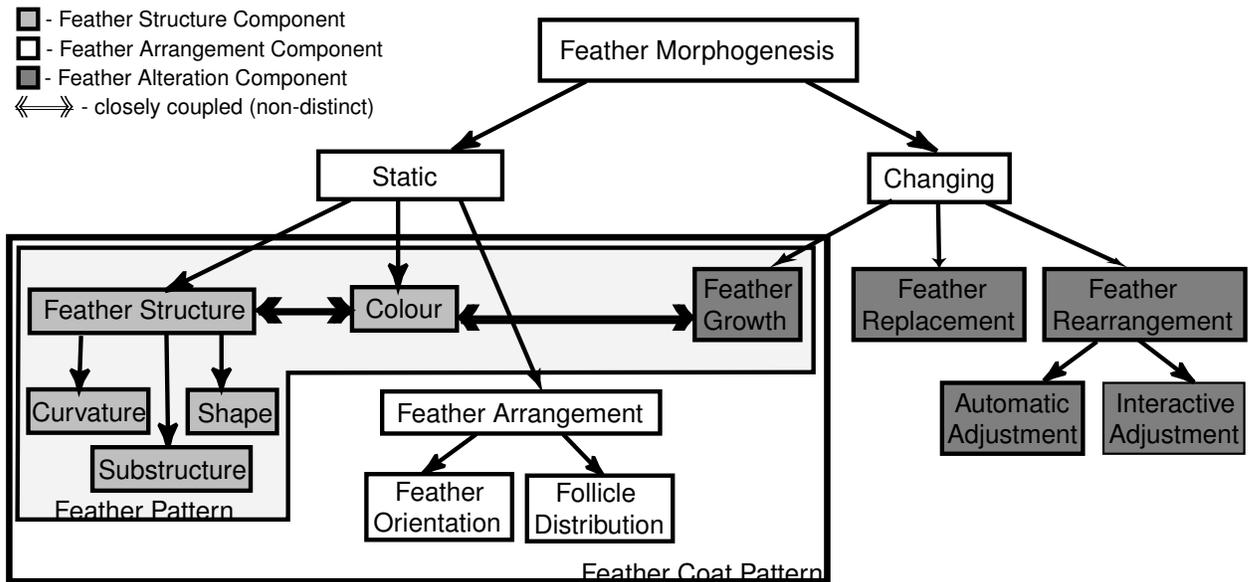


Figure 4.1: Breakdown of Feather Coat Morphogenesis Problem.

biology and is outside the scope of this thesis. Thus, the feather structure component (see Figure 4.1) includes the modelling and *parameterization of structure* including substructure, curvature and shape and the colour. In addition, as will be outlined in Section 4.3.3, the feather structure parameterization must be extendable to parameterizing the growth of the feather. Thus, this alteration aspect must be built into the structural parameterization.

4.3.2 Feather Arrangement

Modelling the feather arrangement is the second component. This includes the *follicle distribution* which is fixed once it is established, as well as the *feather orientation*. The follicle distribution must be accurately and consistently identified on the surface of the model. A small set of parameters must exist to control various types of follicle distributions for the various regions, types and ages of the bird. A method of both defining the pterylae regions and the distributions within them is needed. Once this distribution is defined it is fixed in relation to surface location. The method of mapping the distribution to the surface must be scalable. The feather orientation must define both a direction of feather axis and the rotation of the feather toward the surface. The orientation must allow for the coverage of apteria regions by feathers of the neighbouring pterylae in a reasonable way. This part of the problem will involve devising an algorithm for automatic

arrangement or lay of the feathers (orientation and direction) given the set of follicles and the set of feathers on the surface. The feather arrangement algorithm must account for feather-to-feather and feather-to-surface collisions and interpenetrations for a static models.

4.3.3 Feather Alterations

Finally, the third component, feather alterations, incorporates both of the above components in a changing environment. This component involves specifying a methodology for how the feather coat evolves over time both due to growth and movement. It has three elements: *feather growth*, *feather replacement* and *feather rearrangement*. The feather growth element would identify how a feather changes over time. This element uses the feather structure, parameterization, and colouration and interpolates the parameters over time. The growth method used must be general enough to simulate the growth of any feather, with any structure, or for that matter, any cutaneous structure. The second element to this component is feather replacement. It is independent from structure and arrangement in that it identifies timing, frequency and order of feather loss/regrowth from various regions on the surface. It must however, trigger the feather growth process when needed. Finally, the third element, feather rearrangement, defines how the feather coat responds to changes such as feather growth, replacement, alterations of the surface or skin. The feather direction and orientation must change given changes in feather type and size as well as changes to surface curvature and topology, but the follicle distribution must stay fixed. Also, there must be a mechanism for altering the lay of the feathers according to user input. Birds often ruffle or lift their feathers to preen, clean or to exhibit various behaviours (e.g. a frightened Peacock). In order to create changes in form and appearance of the feather coat the ability to alter the lift of the feather is necessary. This element will use the feather arrangement algorithm of component two, along with additional information to both *automatically and interactively adjust* the feather arrangement.

4.4 Main Contributions

The main contribution of this work to the area of computer graphics is a methodology for defining and managing an intricate, geometrically complex surface, that is both diverse and constantly changing, in the context of feather modelling. The solution is the proposed flexible, versatile and efficient methodology for modelling feathers, feather coats and feather coat morphogenesis. While the proposed approach is not

biologically exact, it does provide a biologically based initial model for assisting in visualization of feather structure, growth and feather coat morphogenesis.

4.5 Summary

This chapter details the scope and goals of the feather coat morphogenesis problem. The scope and goals were outlined first in the context of natural phenomena models outlining the limitations, the desired level of detail, the degree of object specificness, balance of emergence, control and efficiency as well as the input and output to the problem. The overview of the approach was outlined by dividing the problem into three non-distinct components: feather structure, feather arrangement and feather alterations. The next four chapters will outline the integrated solutions to these three components and compare the details of the approach to previous work. Chapter 9 will show results of this feather coat morphogenesis methodology as well as a comparison with current research results and real feathers.



Modelling Feather Structure

The feather structure component as outlined in the Chapter 4 encompasses modelling the *physical properties* of the feather namely all structures and substructures as well as colour and growth of the feather. Modelling of the physical substructures of the feather is necessary given our¹ goal to simulate changes in form and our chosen level of detail - the feather. Two aspects of physical form that must be addressed are the *parameterization* of structure in order to create varying feather structures and *surface generation* given the structural description. Also, in order to construct the feather coat a representation for the wide spectrum of feather types and sizes is required, as outlined in Chapter 2. Since these variations in substructure and consequentially feather type contribute largely to the appearance of the feather coat they must necessarily be part of our model. Obviously, manual modelling of the entire feather coat would be too tedious. However, if a parameterization of the feather properties (e.g. substructure, curvature, size and shape) is constructed, then a variety of feathers could be produced by altering a small number of parameters.

5.1 Basis for Feather parameterization

As outlined in Chapter 3, utilizing a small set of parameters to produce complex structures is often used in modelling of natural phenomena. Since our goal is to simulate the structure and development of the feathers, biological structural models were used as a basis.

L-systems [PL90], as previously mentioned, have been used for modelling both the structure and

¹The rest of the thesis is written in the active voice. Our is used to mean the author and the reader in order to include the reader in the discussion of the development of the methodology.

5.1. BASIS FOR FEATHER PARAMETERIZATION

development of many different types of branching structures including plants and trees, and most recently, feathers. L-systems are designed to create complex structures with variable branching mechanisms. Rules are used to specify the branching type, the number of branches, the location and frequency of the branches and angle of the branches. L-systems have also been used to correlate final shape with growth rates [PL90, p. 123-126]. Feathers, as noted, are composed of multiple levels of branching barbs and barbules and hence, the use of L-systems seems natural for this problem.

In the past year, recent work has been done to model feather structure using L-systems. Chen et al. [CXGS02] from Microsoft Research Asia proposed a production based system as outlined in Chapter 3. However, feathers are very regular monopodial [PL90, p. 70] branching structures, where the angle, length and number of the branches are essentially uniform over the entire feather and thus the overhead for specifying variation is not needed. Also, as stated in Chapter 2, since feather growth occurs proximally at the base rather than throughout the structure or distally at the apical ends, the functionality of branches elongating to simulate plant growth is not needed. Finally, if shape or form is defined indirectly by a set of productions or rules, then it is not clear how smooth interpolation between these forms can be obtained as is required with feather types. Chen et al. [CXGS02] overcome this problem by defining a bounding curve for the silhouette edge of the feather. As recognized by Chen et al. [CXGS02] L-systems are a valid option for modelling feathers and feather structure. Chen et al. [CXGS02] did not attempt to model growth or feather coat morphogenesis. Therefore using L-systems for their models was appropriate. However, our initially-proposed goal of specifying a set of simple parameters for each feather type and interpolating for various types and ages of feathers seems more applicable to the regularity of the feather structure as well as growth and morphogenesis and thus is the approach targeted in this work.

Dai et al. [DSC95] introduce a technique to model feathers of a specific family of birds with a functional set of parameters. They use a set of parameters for rachis length, barb length, barb angle, and a function for curvature of the barbs. However, the curvature of the barb is only two-dimensional, there is no way of specifying rachis curvature, and no provision is made to simulate asymmetry. While their technique provides good initial results, because of the way they iteratively grow their barbs, there is little control over the final shape or the feather limiting their results to only a few feather shapes and limited feather types. Since the characteristics separating the various types of feathers (as outlined in Chapter 2) are the length of rachis, barbs and barbules and the interlocking nature of the barbules, a parameterization that includes these characteristics is necessary for creating a model for a wide range of feather types. Our approach to

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	v_{ba}^l, v_{ta}^l
	right vane angle	v_{ba}^r, v_{ta}^r
	spacing	sp_b, sp_t
	# key barbs	m
	set of key barbs	$B^j, \{\forall j 0..m\}$
	\forall Key barb	# control vertices
	control vertices	$cv_i, \{\forall i 0..n\}$
	position on rachis	rp

Table 5.1: List of parameters in our model

modelling feather structure utilizes a set of parameters vaguely similar to Gelder and Wilhelms's [GW97] fur modelling and Dai et al.'s [DSC95] feather modelling, extending Dai et al.'s work to allow for a more general model of both feather type and development.

Our feather model emerges from the need to have an intuitive set of biologically-based parameters with both the flexibility to create a wide range of feather types, and the simplicity to generate thousands of uniquely structured feathers. The main aspect is based on 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. Further construction of the feather coat is discussed in the next chapter.

5.2 Our Parameterization of Individual Feathers

Our parameterization is based on the biological structure of the feather as outlined in Chapter 2. The two main structures modelled are the shaft and the barbs. The afterfeather is modelled as a secondary feather

5.2. OUR PARAMETERIZATION OF INDIVIDUAL FEATHERS

attached to the main feather and has the same parameter set as the main feather. The set of parameters used are shown in Figure 5.1 and are found in Table 5.1. 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 or lateral r_{lc} , and front-to-back or ventral-dorsal 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.

Figure 5.2 shows the implications of some of these parameters. All feathers in this image have constant barb length and density as well as, one key barb near the base and a default key barb at the tip. The two leftmost feathers show alteration in rachis length r_l and rachis width r_w . The two rightmost feathers show alterations in rachis curvature both lateral r_{lc} and ventral-dorsal r_{vdc} .

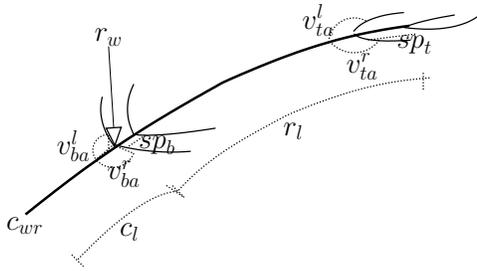


Figure 5.1: Illustration of the parameters of our model from Table 5.1.

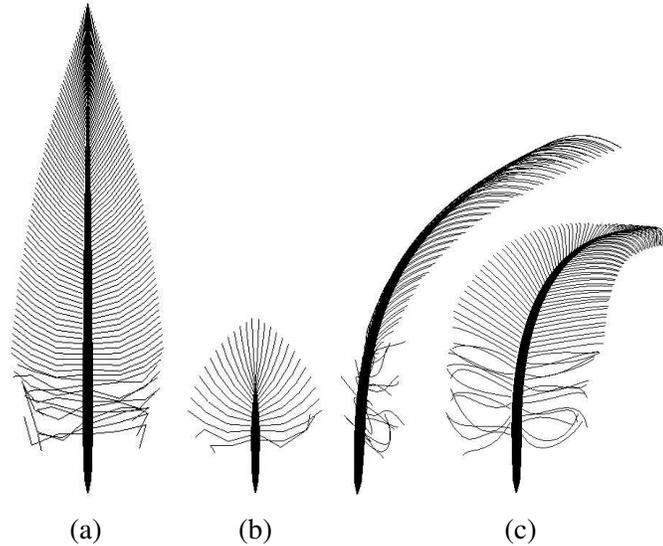


Figure 5.2: Feathers illustrating different parameters. All feathers have constant barb density and length, as well as one key barb near the base and the default key barb near the tip. (a) tall thin rachis r_l, r_w , (b) short fat rachis r_l, r_w , (c) side and front view of rachis with lateral and dorsal-ventral curvature.

The vane is modelled by a collective set of barbs. A select set of m key barbs is specified and from these the rest of the barbs are generated by interpolation. Each key barb $B^j, j = 0, 1, \dots, m$ has a set of n control points $cv_i, i = 0, 1, \dots, n$ 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 5.2.2. 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 (v_{ba}^l, v_{ta}^l) and right (v_{ba}^r, v_{ta}^r) vane as shown in Figure 5.1. 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 in-between. 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. The specification of the silhouette boundary points is similar to the approach taken by Chen et. al. where silhouette curves are defined. Since in our model, the barbs are specified as curves, a simple scaling of the parametric curve is used to both maintain the curvature and control the length of the barbs and hence shape of the feather.

Some feathers, such as primary flight feathers do not have symmetric left and right vanes. Thus, it is important to be able to create asymmetric vanes. This is done in our model by having parameter separation of the left and right vanes. By being able to specify separate key barbs for these vanes, as well as separate branching angles, dramatically asymmetric feathers can be created. An example varying the barb angle and key barb curvature to create a primary-like flight feather is shown in Figure 5.4. Note the decrease in angle between the rachis and barbs at the base of the right vane.

5.2.1 Interpolation

Interpolation is an important aspect of our feather and feather coat model. It is used 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. Details of creating a feather coat from key feathers are found in Chapter 6.

5.2.2 Interpolation Within a Feather - Barb Generation

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 Bézier curve defined by the set of n control vertices $b_{cv_i}^k$:

$$\forall i \in \{0, 1, \dots, n\} : b_{cv_i}^k = (1 - t)B_{cv_i}^a + tB_{cv_i}^b \quad (5.1)$$

where, $t = (b_{rp}^k - B_{rp}^b)/(B_{rp}^a - B_{rp}^b)$. 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. Since many feather barbs are simply usually quadratic and sometimes cubic, $n = 4$ is often sufficient.

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. The random barb is constructed by adding a small amount of randomness to the location of four evenly spaced control vertices along a straight line. Then random displacements [-0.5..0.5] are added to these control vertices to offset them in two dimensions from the line with a weighting factor to increase randomness toward the tip of the barb. This simulates the natural increase in displacement toward the tip of the barb. 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 key 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). The locations of these default barbs are always at the base and the tip of the feather. All intermediate barbs are generated as usual.

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 [RC96] that a Bézier curve of degree n with control vertices cv_0, cv_1, \dots, 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:

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

Once the barb control vertices are obtained, they are scaled by the length of the barb. With a Euclidean distance of one between cv_0 and cv_n the relative curvature of the vane is maintained through the scale, but when the distance is less than one the barbs scale non-linearly with respect to each other. This can be used to create discontinuities in the vane.

Figure 5.3 shows examples of alterations in barb spacing (sp_b, sp_t) and vane shape using key barbs. The leftmost feather (a) increases in barb density toward the tip. The feather to its right (b) shows how key barbs can alter the internal shape of the feather. The key barbs are shown in thick black on the right. Note the lack of symmetry in the key barb curves, location and number. Finally the feather in (c), shows how both

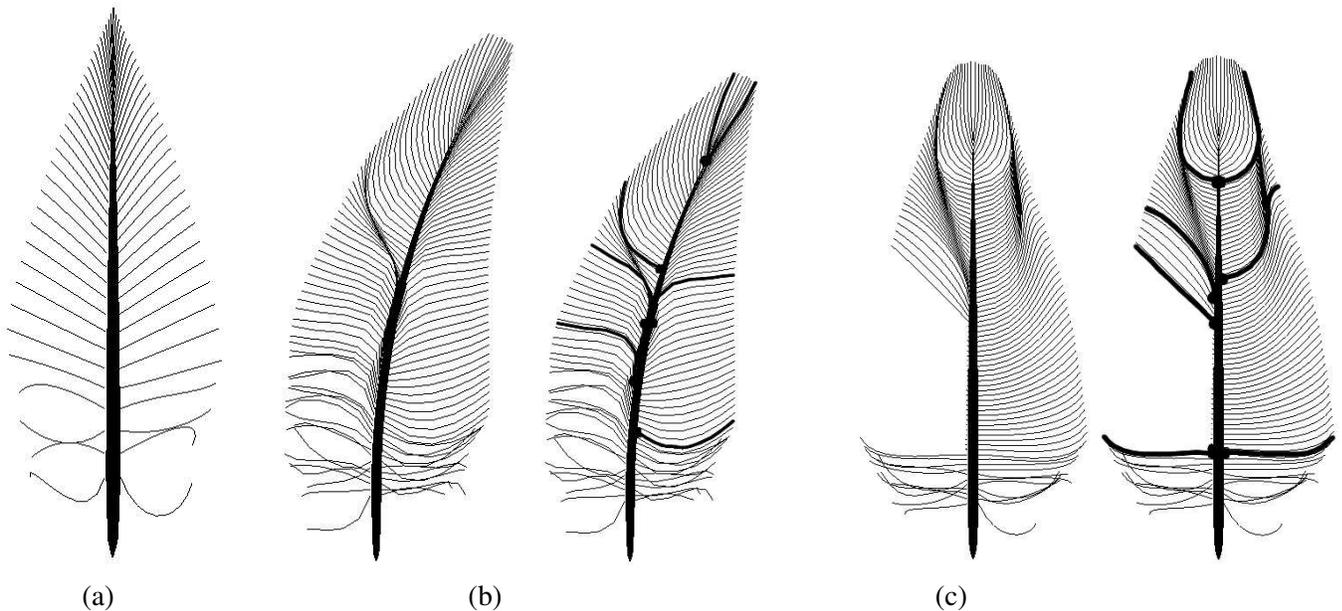


Figure 5.3: Shaping feathers with key barbs. **(a)** Feather with barb spacing variation (sp_b , sp_t). **(b)** Feather with variation in barb curvature and number of key barbs. Key barbs shown in black on right. **(c)** Feather with key barbs creating discontinuities. Key barbs shown in black on right.

the internal and silhouette shape can be altered to form discontinuities in the shape. This is done by placing two key barbs of varying parametric length or curvature immediately adjacent to one another.

Figure 5.5 shows some non-contour-like feathers constructed using our parameterization. The down feather is an example of a feather with no key barbs and a very short thin rachis and calamus. The filoplume and bristle are specialized feathers and were constructed using the key barbs on the right. Shape data was also used for the bristle. Compare these with Figure 2.4

5.2.3 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 segmentation algorithms, not only can the barb length be extracted from the scan, but also the placement and control points of key barbs, barb spacing and barb angle can be extracted. However, the ventral-dorsal curvature would still have to be added manually. An example of the scanning process is detailed in Chapter 9.

Figure 5.4 shows feathers using shape data. The shape data or contour is shown on the far left. A feather with no key barbs (except default key barb at tip) is to its right, followed by two feathers with a

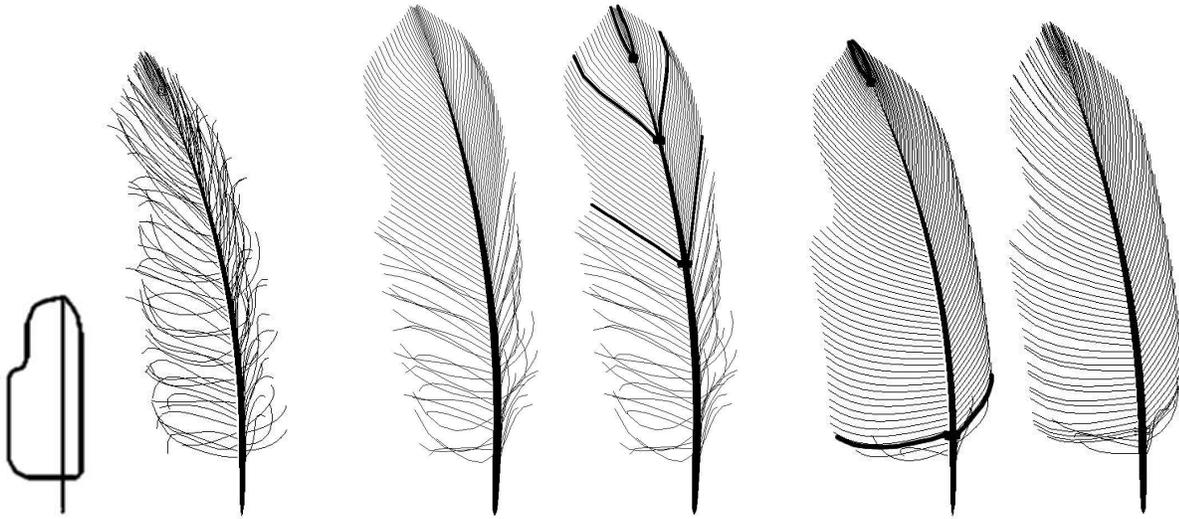


Figure 5.4: All these feathers are constructed using the silhouette shape data on the left. Note the variation in barb angle of the right and left vanes at the base ($v_{ba}^l, v_{ta}^l, v_{ba}^r, v_{ta}^r$). **Left:** Feather with no key barb, **Centre:** Feather with key barbs shown in black at centre right. **Right:** Feather (key barbs in black) with and without splits in the vane.

few key barbs distally and two feathers with a key barb at the base and tip. The feather second from right resembles a typical primary flight feather structure. Finally, the rightmost feather is similar to this primary-like flight feather, except splits in the vane have been added to simulate wear (see Section 5.2.5). All feathers are oriented with the vane parallel to the viewing plane. The vanes have been modified by varying the barb angle at the base of the right and left vanes ($v_{ba}^l, v_{ta}^l, v_{ba}^r, v_{ta}^r$).

5.2.4 Feather Types and Interpolations

As outlined in Chapter 2, 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 physical regions on a bird's body 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 5.2.2. 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 to generate a new key barb set then the remaining barbs are generated

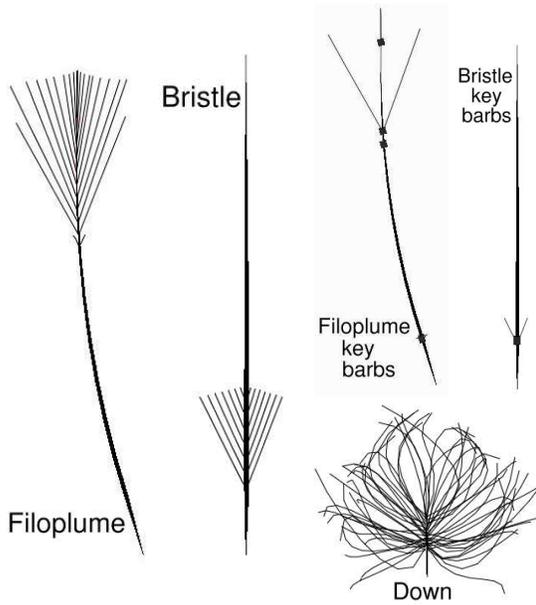


Figure 5.5: Bottom: Feather structure generated with our parameterization. **Top Right:** key barbs used for feather generation (none for down). Barb length was specified with a silhouette shape. Compare these with the feathers in Figure 2.4.

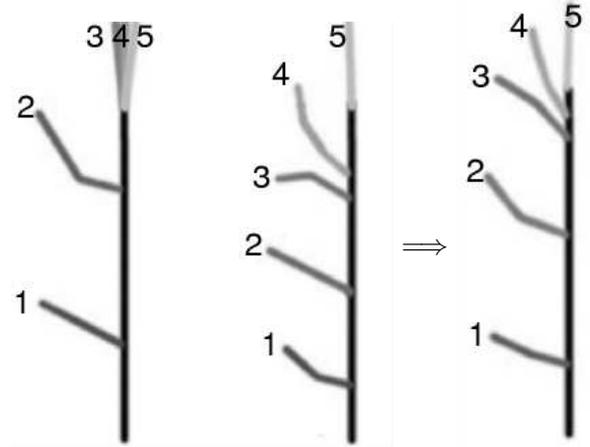


Figure 5.6: Left: Two feathers with left vane key barbs. **Right:** A feather evenly interpolated between the two on the left with new key barbs. Note the addition of extra key barbs at the tip to satisfy matching requirements.

from this new key barb set as outline in Section 5.2.2. When interpolating between two feathers with varying numbers of key barbs, the feather with fewer key barbs is artificially inflated until the number of key barbs are equal as shown in Figures 5.6 and 5.7. Figure 5.6 shows the left vane key barb interpolation between two feathers. The one feather originally had two key barbs labeled one and two. The other feather originally has five key barbs. Any extra key barbs (three) are added at the tip of the feather. The control vertices of the extra key barbs mimic the shape at this location (in this case the shape of the default barb). The key barbs are then matched from base to tip using the parametric rachis position (rp - attachment point of barb to rachis). This ensures an optimal shape match regardless of scale. A midpoint interpolation of these two sets of key barbs is shown on the right. Note how both the control vertices of the barbs and the point of attachment to the rachis are modified.

The control vertices of the new key barbs are generated using bilinear interpolation. To generate a new feather F' from feathers F^a and F^b , each new key barb is generated by first determining its parametric rachis position rp , and then computing the new control vertices. The rachis position is determined by simply interpolating the parametric rachis positions of feather F^a and F^b . The new key barbs are generated by constructing two temporary key barbs on each feather F^a and F^b . A temporary key barb Z_{rp}^a is constructed

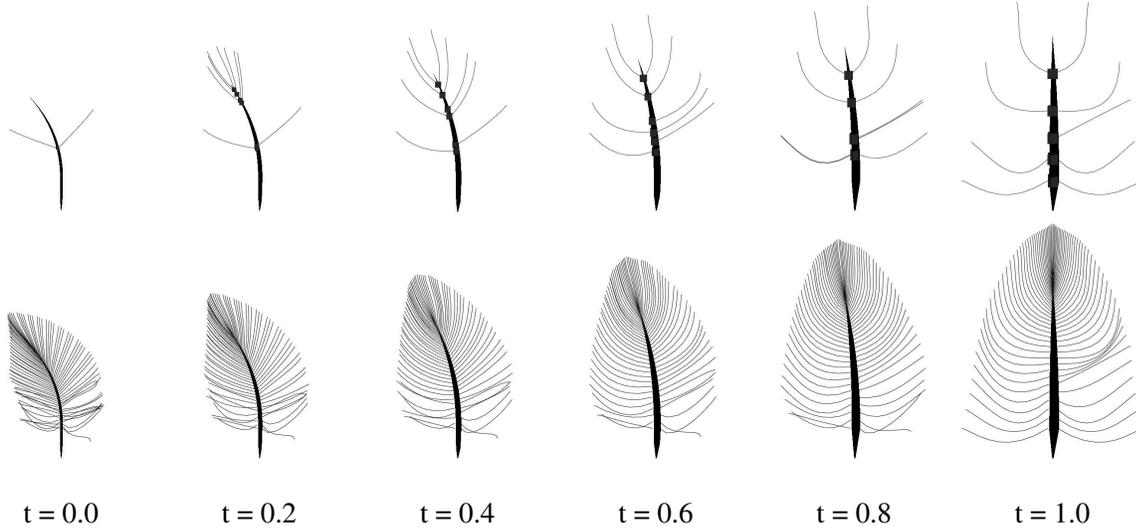


Figure 5.7: Interpolation between leftmost and rightmost feathers. **Top:** key barbs. **Bottom:** feather structure.

on feather F^a at location rp (from key barbs above and below) and a temporary key barb Y_{rp}^b is constructed on feather F^b at location rp (from key barbs above and below) as follows:

$$\forall i \in \{0, 1, \dots, n\} : Z_{cv_i}^a = (1 - r)A_{cv_i}^a + tB_{cv_i}^a$$

$$\forall i \in \{0, 1, \dots, n\} : Y_{cv_i}^b = (1 - s)C_{cv_i}^b + sD_{cv_i}^b,$$

where $A(C)$ and $B(D)$ are key barbs above and below rachis position rp on feathers $F^a(F^b)$ respectively and r and s are normalized interpolants from the rachis attachment points of A , B and rp . Linear interpolation of these two temporary key barbs (Z_{rp}^a, Y_{rp}^b) is then used to construct the final key barb X'_{rp} on feather F' as follows:

$$\forall i \in \{0, 1, \dots, n\} : X'_{cv_i} = (1 - t)Z_{cv_i}^a + rY_{cv_i}^b,$$

where $t \in [0..1]$ is the blend factor between the two feather types.

Figure 5.5 shows some non-contour-like feathers constructed using our parameterization. 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 5.5 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.

As mentioned it is desirable to have a continuum between feather types. Figure 5.7 shows a continuum constructed from the feather types at the far left and far right. The key barbs are shown on the top and the resulting feather structure on the bottom with the interpolant or blend factor.

5.2.5 Modelling Feather Growth and Change

Due to the sequential loss/regrowth of feathers and the cyclical nature of moulting, a feather coat very rarely consists of entirely fully developed feathers and in some moult stages only a few feathers are fully developed. Also, as fully developed feathers age they can wear altering their structure. 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. Then the sheath is shed from the tip to the base. As the sheath sheds, growth ceases in that region. 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. These extra key barbs control the the angle and curvature of the barb until the feather opens. This will be discussed more in Chapter 8.

The feather structure changes not only due to growth, but also due to wear. As shown in Chapter 2, as a feather ages, it is possible for the barbs to wear - effectively shortening their lengths. Also, due to preening, wind pressure and obstructions (i.e. foliage) the feather vane of the flight feathers may develop gaps or breaks and are thus not perfectly continuous. The interlocking barbs sometimes become separated (split), until the bird has a chance to repair them. Thus, in order to capture the realism of the changing structure, the split in the vane of the feathers must also be modelled. In our model splits are simulated by randomly generating split locations along the vane. Then the two barbs that fall on either side of the split location have their angles fixed to the angles of the barbs closest to them as shown in Figure 5.4 on the far right. The use of randomness for split locations is similar to Chen et al.'s [CXGS02] approach.

5.3 Surface Generation

Once the structural or mathematical model is constructed, a polygonal or mesh surface must be generated for rendering. Surface models for plant organs such as leaves and trees [PL90] use both bicubic patches and polygonal meshes. Plant organ growth has been simulated using sequential addition and scaling of polygons according to growth parameters. Hair and fur techniques using volumetric textures cannot be applied to feathers. With feathers there is a need for independently moving structures. Hair structure has typically been modelled using straight line elements [Mil88], a series of primitives to simulate a cantilever beam, such as polylines [iAUK92, SG94, RH96, GW97], cylinders [Mil88, Ski89, RCT91, DTKT93] and trigonal prisms (wisps) [WS92, AM95] as well as truncated cones [Gol97], particle systems [LTT91, Ree83] or NURB curves and surfaces [GW97].

Three previous works on modelling feathers used polygons or NURB patches. Dai et al. [DSC95] used polygonal meshes for the vane and generalized cylinders for the rachis. The vanes were texture mapped with a constructed feather texture and all barb information was discarded except what was needed for the final boundary of the feather. Chen et al. [CXGS02] used quadstrips to model each barb. They mentioned using level of detail algorithms for rendering, but not for geometric modelling. This was reflected in their one minute-per-frame rendering times. Engle [EAB⁺02] at Sony Pictures Imageworks used a set of NURB patches to construct a surface for their feathers. Each feather could be a single patch or many patches depending on the size, shape and splits in the vane. This approach integrated nicely with their NURB arrangement tools and controls. However, it did not allow for the creation of non-continuous vane feathers such as down or semi-plumes. This type of effect was approximated using hair strands.

Our technique constructs a surface for a feather directly using the substructures of the feather. From the barb curves a triangulation is constructed that reflects the structure, form, and detail of the feather. The triangulation is constructed in a two step process: first the barb curves are tessellated and second the triangulation is constructed given the characteristics and required level of detail. This level of detail approach will be discussed more in a rendering context in Chapter 7. Below we present the geometric details.

One last parameter, tessellation, is used to control the level-of-detail for rendering purposes. However, it must be noted, that the design of the model easily facilitates the ability to tailor a resulting generated surface. Each barb curve is approximated by a polyline set of line segments. The number of line segments used can be varied as shown in Figure 5.2 (a,b v.s. c). While the overall “look” of the feather does not change - silhouette shape, barb density and length are all preserved, the amount of geometry required is

greatly reduced due to the quantity of barbs. This tessellation parameter can easily be tied with the view distance to give view-dependent geometric representation of the feather surface.

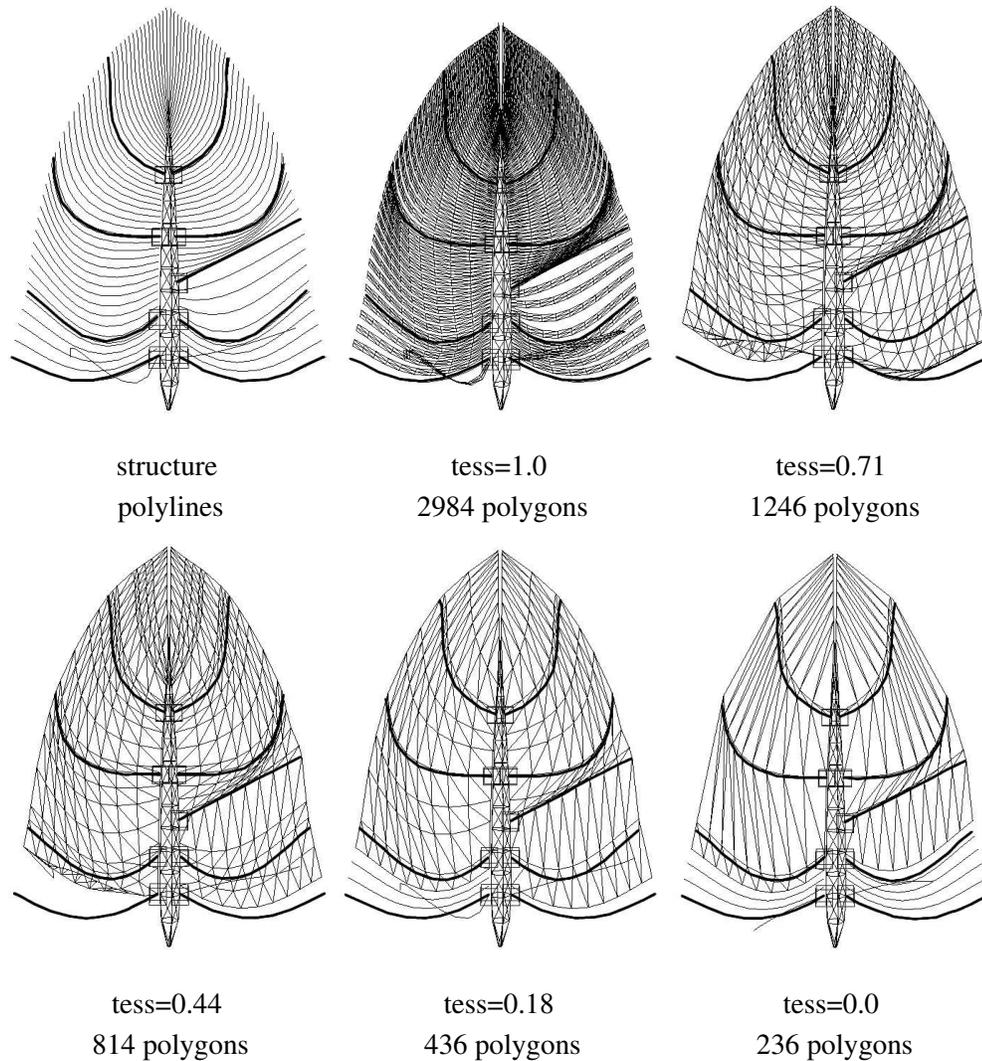


Figure 5.8: Key barbs are shown in thicker black lines. **Top Left:** Structure of the feather in polylines. **Rest:** decreasing LOD with tessellation parameters and number of polygons.

Once the barbs are approximated by a set of polylines, the vertices of these polylines are connected on each side of the vane to form a mesh. If splits are present in the vane there needs to be a boundary edge in the mesh along the barbs bordering the split. Since the triangulation is derived directly from the vertices of the barbs, as the barb curves increase in resolution, so to does the triangulation. At full resolution each barb is generated by two triangle strips (one on either side of the barb). At lower levels of detail in the closed vane portions (between key barbs and splits) the mesh is constructed from only a small portion of the

defined barbs. The algorithm used to choose the barbs to construct the triangulation attempts to minimize shape distortion by preserving the key barb representation as long as possible. This means that the barbs closest to the key barbs are chosen to be collapsed and merged last, and that the barbs between the key barbs are collapsed and merged in a dispersed order (divide and conquer), to avoid local distortions. Figure 5.8 shows the feather structure on the top left and five levels of detail with the triangulated surface mesh. More information on our level of detail approach and how it impacts feather rendering will be given in Chapter 7. Note how the key barbs are preserved through the lowest level of detail. Since the key barbs define the feather shape, by collapsing them last we ensure maximum preservation of feather shape even with extreme reduction in geometry. Using a fixed number of vertices per barb, the polygon count is reduced from 3,000 to approximately 200. This can further be reduced by reducing the number of segments approximating each barb.

5.4 Feather Colouration

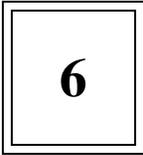
The second element to the feather structure component is the *feather colouration*. This includes not only colour pattern, but also reflectance properties. Past work has always used texture maps for colour pattern. The work of Chen et al. [CXGS02] incorporates transparency as an reflectance property. Since both of these attributes are not the focus of this work, only an approximation of these attributes is made.

Our approach uses the advances in image capture technology and recent trend of image-based rendering methods. Feather textures are acquired from real feathers. Chapter 7 further details our approximation to various illumination effects. Feather illumination is closely coupled with the meso-structure and colouration of the feather as specified in Chapter 2. Since local feather illumination is a complex physical property including multiple reflectance, transmittance, and sub-surface scattering it is an entire research area itself. Current research in hair rendering has just begun to address some of these properties [MJC03]. Perhaps as future work a similar model for feathers could be adapted or constructed for feathers.

Biological pattern synthesis has been an area of ornithological and biological research for decades and still only theories of feather pattern formation currently exist. While addressing this area is outside the scope of this thesis having believable colouration, patterning and illumination adds to the realism of the feather coat. Thus, we incorporate the pattern into our methodology by assuming that realistic patterns can be acquired or synthesized.

5.5 Summary

This chapter presented one component of our feather coat morphogenesis methodology. It focused on presenting the parameterization of the feather structure. It was shown that with this parameterization a variety of feather types could be constructed as well as providing a methodology for generating smooth continua of feather types which can be seamlessly interpolated between. A method for generating a surface representation from this parameterization was also presented. The characteristics of the parameterization allowed for generation of multiple surface representations that can easily be tied with viewing distance to create a level of detail approach which will be further outlined in Chapter 7.



Modelling Feather Arrangement

As specified in Chapter 4 the feather arrangement component is composed of two elements: *follicle distribution* and *feather orientation*. In Chapter 2 it was stated that the follicle distribution is relatively fixed over the lifetime of the bird. A few follicles may be added to the boundary of pterylae regions or the relative positioning may change slightly due to stretching of the skin during growth but only in very early stages of development. With the exception of these early developmental stages, the pattern remains roughly hexagonal and while the absolute density of follicles may change, the distribution is unaltered. The feather orientation is the second element of this component. Altering the orientation of the feathers provides a mechanism for changing both the shape and pattern of the final feather coat. However, some aspects of the feather orientation are a product of the physical attributes of the feather and follicle. As stated previously, in a natural state the feathers of pterylae regions have a specific ordering and direction and often cover adjacent apteria regions. The automatic computation of the natural state of the feathers is a necessary part of our feather coat model. Further control of alternate orientations will be explored in the feather rearrangement element of the feather alteration component in Chapter 8.

In order to generate realistic-looking collections of feathers many details must be incorporated. The feather coat is not only affected by the individual feathers, but also by the distribution, orientation and interaction of the feathers. Feathers as mentioned in Chapter 2, are distributed within feather tracts on the body of the bird and have an anisotropically stretched hexagonal distribution. Our approach uses this notion of a tract for both computing the direction and orientation and to allow for styling. Biologically accurate feather coats can be designed using accurate tract positions, but for computer graphics purposes liberties can

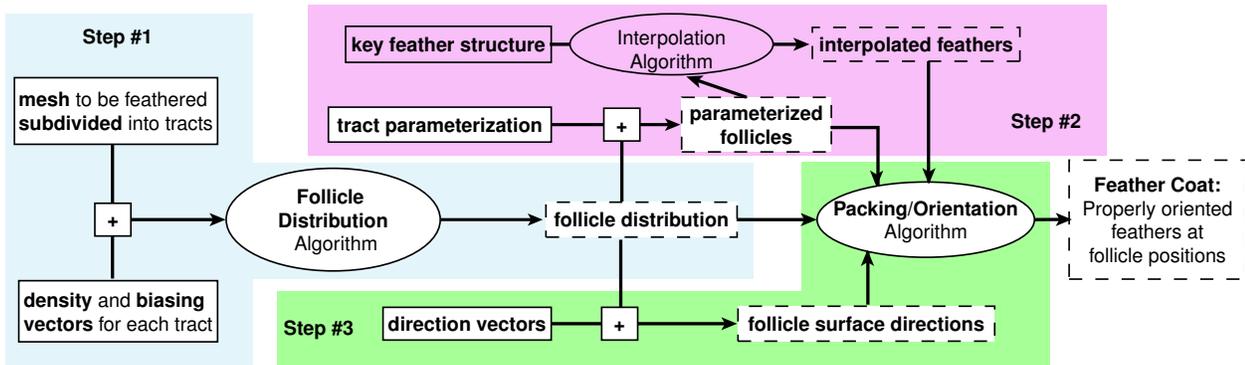


Figure 6.1: Modelling of the feather coat in three steps. Inputs are solid rectangles, output and data transfer in dashed rectangles. The white regions are pieces of our model that are explained in this chapter.

be taken with the tract layout. The orientation of the feathers is not a simple task since the feathers need to automatically compute a stable state as well as react to each other given changes in this state. They must not interpenetrate and yet they must be packed fairly tight. Given the enormous amount of geometry and the variability in feather shape, if normal collision detection schemes were used either simplified geometry or a lot of computation time would be required. In this chapter a method for determining the stretched semi-regular distribution of the follicles, as well as an efficient method for determining the orientation of the feathers is presented.

Our approach assumes that the model to be feathered is a polygonal mesh and that it is has been sufficiently subdivided so that feather tracts can be accurately formed from subgroups of polygons. This assumes a certain minimum resolution to the mesh, especially for forming accurate tract boundaries. The specific process we used for doing this will be discussed further in Section 6.1.

The process of modelling a feather coat is shown in Figure 6.1. There are three main steps: follicle distribution, feather generation and feather orientation. The second step involves interpolating the feather types which was largely explained in Chapter 5. Given a mesh with polygons grouped into feather tracts, the user specifies a *follicle density* and *bias vectors* for each of these tracts. With this information, the follicles are distributed geometrically, as described in Section 6.1 below. In addition to a location on the model, the follicles need an associated parametric location for interpolation and ordering. To avoid constraining the type of model to be feathered by using a global parameterization, an independent local parameterization is imposed on the individual tracts of the mesh by projecting a rectilinear grid from screen space onto the model as discussed in Section 6.2.1. Next the user associates key feathers with selected follicles positions. Using

the key feather structure, the imposed parameterization and interpolation method presented in Chapter 5, the feather structures at all follicle locations are generated. Finally, the resulting orientation of the feathers in each tract must be determined as discussed in Section 6.3.

6.1 Step One: Follicle Distribution

As mentioned in Chapter 2, our focus is on the modelling of modern day contour feathers. While some primitive birds and flightless birds have feather distributions similar to hair, most bird's contour feathers are restricted to pterylae regions. Thus feather follicles have a much more organized distribution than hair. Not only are they organized in pterylae regions, but they are organized in rows and form a rough regular hexagonal pattern, with some skewing due to axial growth (see Chapter 2).

Computer graphics hair modelling systems address hair distribution in a similar context. The areas where hair is present have been defined by interactive placement of follicular maps [RCT91, WS92] and ellipsoidal regions [iAUK92]. Creating a distribution of follicles within a follicular area or pterylae region is of less importance because hair is less organized (although also roughly hexagonal) and hairs are bilaterally symmetric, and thus a rough uniform approximation of the distribution is often sufficient. Some use manual placement of the hair follicles [WS92]. Kajiya and Kay [KK89] use a Poisson disk distribution as an close approximation to the actual distribution of fur. Goldman [Gol97] uses a Poisson distribution to ensure simplistic calculations of collisions between short hairs. Gelder [GW97] also approximated fur distribution with a Poisson distribution.

Generating the hexagonal pattern of follicles and cellular arrangements has been the work of many developmental biologists. In particular Shaw and Murray [SM90] create patterns similar to feather follicle distributions by using a superposition of two wave-like patterns. They use a reaction-diffusion mechanism to create a pre-pattern simulating the epidermal layer and a mechanochemical mechanism in simulating the dermal layer to generate a spatial distribution. They claim the pattern generated must be generated by two wave-like patterns with a significant wavelength difference, eliminating the possibility of a single process or a sequence of two or more processes as with Nagorcka et al. [NM85].

An approach similar to Rosenblum et al.'s [RCT91, WS92] hair follicle distribution can be used to interactively define the follicle regions or pterylae. Since our approach assumes we have minimal information about the model surface (i.e. a set of connected polygons), the user must first divide the model into

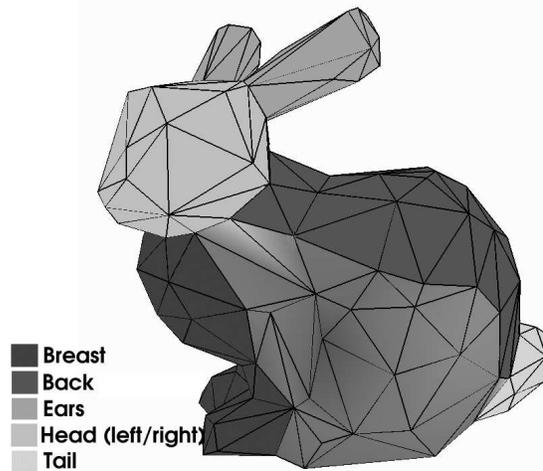


Figure 6.2: Bunny model divided into *tracts*. There are eight tracts: ears, head-left, head-right, breast, back, body-left, body-right and tail. Two are not labelled (body left and right) rest are coloured with gray tones. The ears will need to be divided in two for parameterization.

tracts. This is done by identifying groups of polygons that belong to each feather tract. A polygon can belong to more than one tract, and thus the tracts can overlap and intersect. In our models two sets of tracts were used: one for down and the other for contour feathers. The resolution of the model was assumed to be fine enough to accommodate tract definitions at the polygon level—otherwise, the model was subdivided. Our particular implementation allowed for the use of set operators on both individual and groups of polygons to create a selected set of polygons which defined an individual tract.

Once the tracts are defined, the follicles can then be distributed for each pterylae. The distribution within the pterylae regions can be done in two ways: explicitly or procedurally. The explicit specification of the distribution involves the user painting follicle regions on the surface or creating a texture map to be used as a density distribution. The procedural approach involves the user specifying a probability density distribution function or the parameters to a reaction diffusion process. With both the probability density distribution and the reaction-diffusion approach the distribution can take place directly on the surface of the model. While using a texture map for the distribution seems more intuitive and controllable, it either requires a proper parameterization of the model or introduces possible distortions to the distribution.

Within each pterylae the goal is to achieve an approximately regular distribution. Turk [Tur91, Tur92] has suggested a procedural force based method of achieving regular distribution over a polygonal surface by randomly distributing points, projecting them into the plane of the adjacent polygon, and using repulsive forces to move the points apart until stabilized. He also suggests a modification of this technique

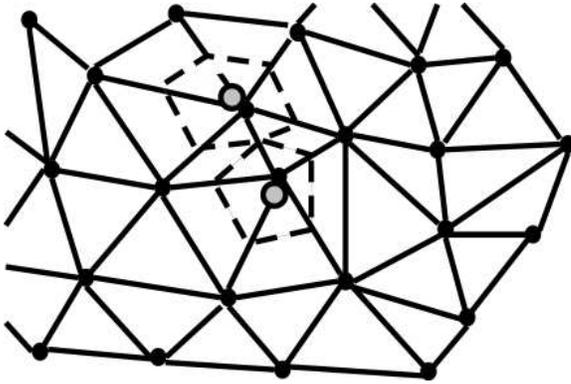


Figure 6.3: Lloyd's algorithm creates a uniform 2D distribution by moving the centre vertex of the Voronoi region (dashed region of Delaunay triangulation) to its centroid (gray circle).

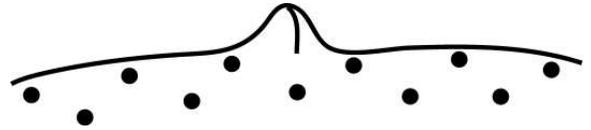


Figure 6.4: Uniform distribution may fail to be uniform with respect to geodesic distance if the follicle density is sparse in comparison to the feature size of the surface.

to adaptively increase the density of these points in regions of higher curvature. Chen et al. [CXGS02] use this method for distributing feather follicles, but instead of using curvature to adjust the density, they use the feather size. They assume coverage of skin is achieved by increasing the density of feathers where the feathers are smaller.

Our method is similar to that of Turk as well as Chen et al., but does not use forces and is purely geometric. It begins by generating random points (positions) within each triangle of the set of polygons defining a given tract. The local density of these points is based on an area weighting of the triangle and the tract, as well as a user-supplied density parameter. A large table of random barycentric coordinates are generated once initially to preserve consistency when adding follicles to the tract. These positions are then modified to form a more regular distribution. To this end, we employ a variant of Lloyd's algorithm [Llo82], in contrast to the use of repulsive forces in the work by Chen et al. [CXGS02]. Although no formal proof is available for dimensions larger than one, Lloyd's algorithm is generally believed to be free of local minima, while this is a known problem for repulsive force approaches.

6.1.1 Lloyd's Algorithm

Lloyd's algorithm achieves a 2D uniform distribution of the set of points which are the vertices of a mesh. The mesh must be Delaunay triangulated so that the dual (Voronoi diagram) can be used as a measure of point spacing. Each Voronoi region is computed and the centroid of the Voronoi region found. If the centre vertex of the Voronoi region does not lie at the centroid of this region, the centre vertex is moved to satisfy

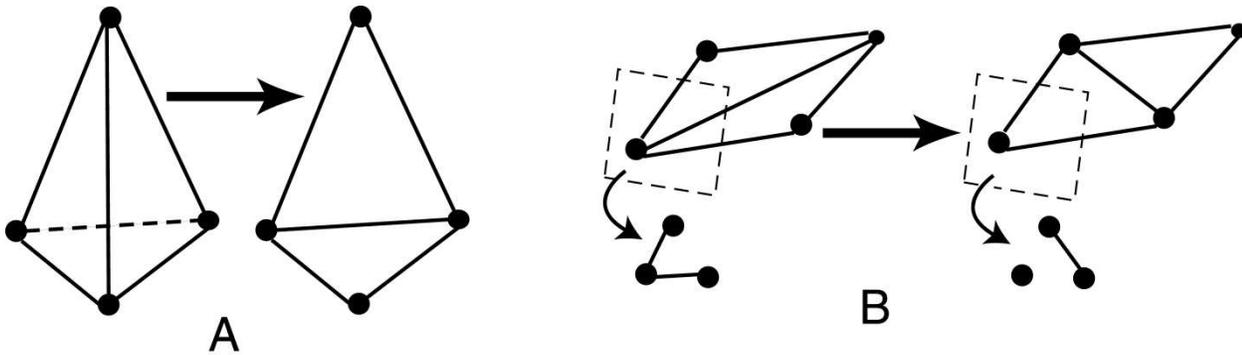


Figure 6.5: Two degenerative cases of using the 2D Edge-flip algorithm for creating a Delaunay triangulation on a 3D surface: **A)** Topologically degenerate **B)** surface smoothing or distortion.

this condition as shown in Figure 6.3. After a few iterations of this procedure over all Voronoi regions, a uniform spacing is achieved.

6.1.2 Delaunay Triangulation On a Surface

We use the original surface mesh and the randomly distributed points (positions) to generate a submesh whose vertices are the randomly distributed positions. Within each triangle of the original model the newly distributed points are triangulated using a plane sweep algorithm and then Delaunay triangulated using the Edge-flip algorithm [dBvKOS98]. Each triangle of the original model then contains a Delaunay triangulated submesh with a set of boundary vertices. Then for each edge E' of each triangle in the original model the boundary vertices of the submeshes are matched and connected using sorted minimum distances from E' . Once all of these boundary edges are connected, the submesh representing the randomly distributed vertices is formed. Up to this point the algorithms used and the triangulation constructed are confirmed stable, since these algorithms are proven to work in 2D and we have used them in the context of a triangle (plane).

Once again the mesh is Delaunay triangulated using the edge-flip algorithm to ensure proper triangulation along each E' edge. However, now the edge-flip algorithm is used in 3D instead of 2D which creates some restrictions and limitations. Instead of using a bounding circle to compute the edges to be flipped, a bounding sphere is used. While this is common practise with 3D Delaunay triangulations, we are still computing a 2D triangulation (on the surface), simply in 3D space and thus local planarity is assumed between adjacent triangles. The two main problems are creating a degenerate mesh, particularly in the case shown in Figure 6.5a, or smoothing sharp features, as shown in Figure 6.5b. For the degenerate mesh case, these edge-flips are simply not allowed and the smoothing case is irrelevant, since the mesh is only

needed to create the Voronoi regions for later relaxation of the points. Because, the points are constrained to move only on the surface of the original model mesh (as outlined below), these inaccuracies of the submesh are irrelevant.

Constructing surfaces or manifolds from sets of points is an ongoing research topic in many areas of research including 3D printing technology, mathematics, computational geometry and molecular biology [Ede98]. However, each of these techniques make certain assumptions or have certain limitations. Our problem requires a Delaunay triangulation of the surface, that does not alter the geometry or topology, however, is able to use information about the input surface topology. Many of these other techniques use 3D Delaunay triangulations or more complicated manifold learning processes or require the point set to form a closed surface, because no information is known about the surface topology. However, not only can we not provide a sample point set which forms a closed surface, but a 3D Delaunay triangulation does not guarantee a surface Delaunay triangulation. Leibon and Letscher [LL00] state that a surface Delaunay triangulation which accurately represents both topology and geometry of the manifold does not exist for an arbitrary set of points, but can be guaranteed to exist given certain density requirements of the points. This is the same requirement as our algorithm above and thus, a Delaunay triangulation representative of the topology and geometry can generally be achieved given this requirement. As well, a Delaunay triangulation of the point sets can always be obtained if topological or geometric distortion (possibly small) is permitted. Since the Delaunay triangulation is only used to relax the vertices and discarded after relaxation and the vertices are reprojected to the surface and thus guaranteed to remain on the surface, this topological distortion is irrelevant except for the impact on the resulting distribution as discussed next.

6.1.3 Applying Lloyd's Algorithm

Once each tract's follicles are Delaunay triangulated (or close to it), Lloyd's algorithm can be applied to generate a uniform distribution of the feather follicles. The algorithm proceeds on the 2D surface in 3D in the same way as the original 2D algorithm with one modification. The follicle density must be representative of the feature size of the object. The centroid of the Voronoi region is computed as before as a weighted sum of triangle areas using the centres of each triangle in the Voronoi region. Once the position of the centroid is computed it is projected back to the surface of the original model, to ensure that the follicle remains on the surface of the object. Thus, since there is no measure of geodesic distance, if the follicle density is sparse in

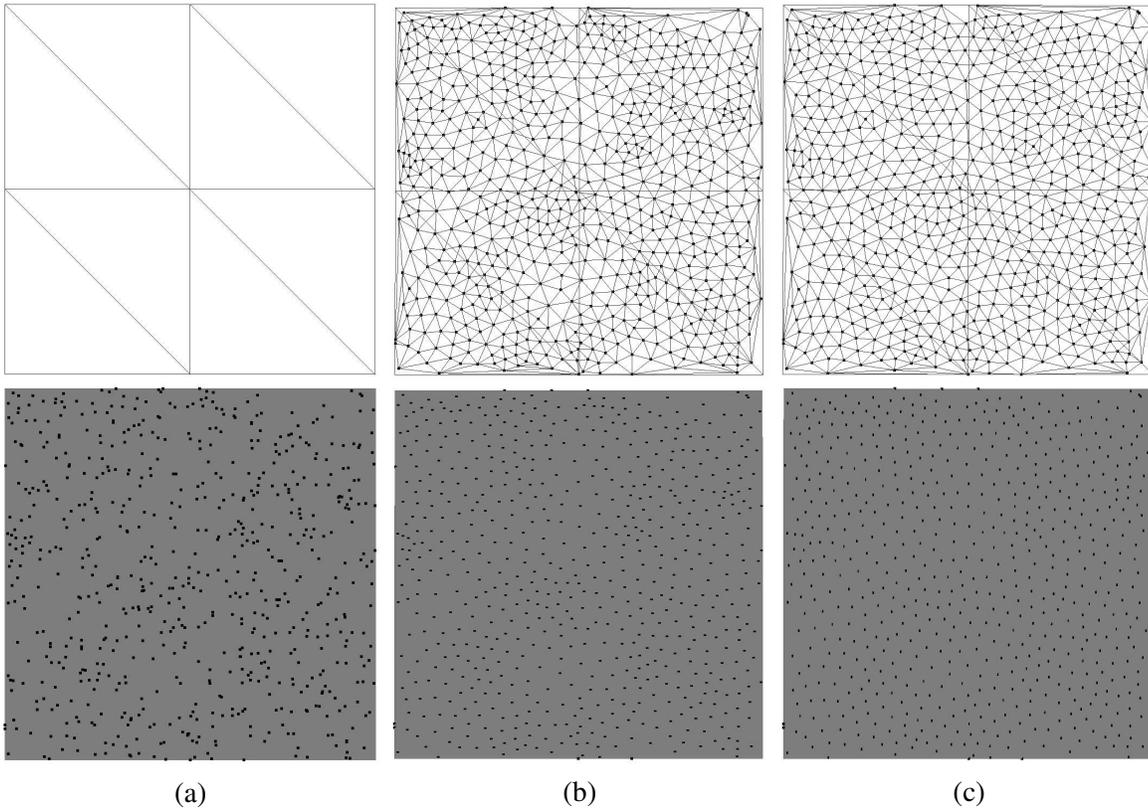


Figure 6.6: Various distributions. **(a)** the initial random distribution below, the original mesh above. **(b)** relaxed uniform distribution after three iterations of Lloyd’s algorithm submesh and original model mesh above. **(c)** uniform distribution after nine iterations of Lloyd’s algorithm - submesh and original model mesh shown above.

relation to feature size of the object, the distribution of follicles may appear to be non-uniform as shown in Figure 6.4.

Also the computed centroids may no longer belong to the same triangle as the centre vertex of the Voronoi region of the original model mesh. Thus the follicles or vertices of the submesh must be allowed to migrate between triangles within the feather tract of the model. Figure 6.6 shows examples of the initial random distribution and multiple iterations of Lloyd’s algorithm. Note that the follicles of the feather tract do not necessarily spread exactly to the edges of the polygons in which they are defined, only that the follicle density is somewhat preserved along these edges. This is due to the irregular triangulation along the edges and because the boundary follicles are not permitted to relax or the feather tract coverage could change significantly. Also we purposely do not permit follicles to lay on the edges or adjacent tracts may experience dramatically increased follicle density on their common edges. The complications of restriction will be further discussed in Section 9.2.1.

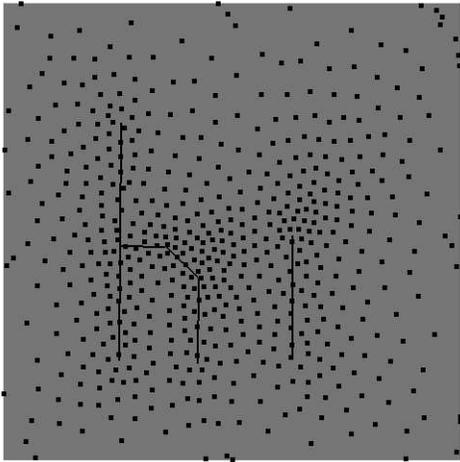


Figure 6.7: Bias vectors spelling “hi” and the resulting biased distribution after a few iterations.

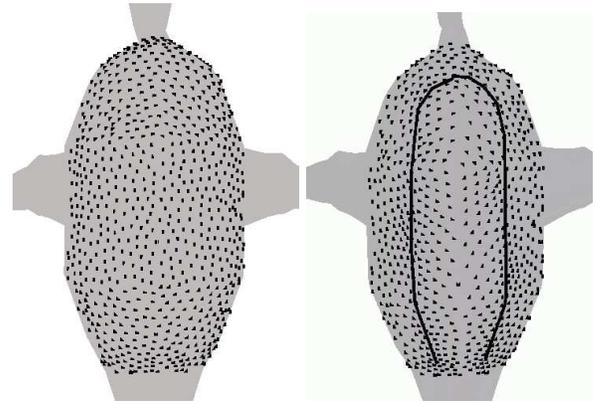


Figure 6.8: Follicles on the underside (belly) of a bird model **Left:** regular distribution from Lloyd’s algorithm. **Right:** “stretched distribution” from our adaptation to Lloyd’s algorithm clustered to the biasing vectors shown in black.

6.1.4 Density Biasing

Although many follicle distributions are uniform hexagonal distributions, some are stretched due to axial growth. In order to achieve the biasing in the follicle density the user can specify biasing vectors or paint density distributions on the surface. Since the density generally increases along the axis of growth, in some cases these vectors can also be used to generate the parameterization for each follicle, as outlined in Section 6.2.1. Then, instead of moving the point to the centre of the Voronoi region we alter Lloyd’s algorithm to move the point in a direction weighted by the user supplied biasing vectors and the centroid of the Voronoi region. The weight is based on distance to closest biasing vector. A Gaussian fall-off can be applied to prevent dramatic clustering. This adaptation, which is similar to the stippling method proposed by Secord [Sec02], creates a uniform distribution with varying density as shown in Figures 6.7 and 6.8. The latter reminiscent of the Ventral feather tract shown in Figure 2.7.

6.1.5 Manual Follicle Placement

In addition to automatically generating uniform distributions and biased distributions a specific placement of follicles may be desired. Interactive follicle placement may be desired because of irregular densities or in regions of large feathers with specific placement. In our model we use this approach for placing the large

flight feathers of the wings and tail. They are positioned manually since they are limited in number (typically less than 25), and are visually important. The rest of the feathers on the wings and tail are distributed using the adaptive Lloyd’s algorithm discussed above.

6.2 Step Two: Feather Generation

The second step of generating the feather coat is the addition of a feather to each follicle. Since it is far too tedious to place each individual feather, certain key feathers are placed. The same interpolation methods introduced in Chapter 5 are then used to construct all intermediate feathers from a small collection of these interactively placed key feathers.

6.2.1 Imposed Parameterization

Before interpolation can be used, a parameterization must exist to relate each follicle to its neighbouring follicles. As mentioned previously this is done by projecting a grid from screen space onto the surface. This assumes that the user orients the model such that when each tract is projected to screen space the projection is relatively undistorted (i.e. that is it fairly close to planar). While this is not an optimal assumption, work is currently being done on parameterizing arbitrary meshes [GG03]. If the surface is not close to planar or some non-linear interpolation of the feathers is desired, distortion of the grid before projection can be used.

Once the follicles are placed and have a parametric location, the user places previously designed *key-feathers* at locations that represent transitions in feather structure, direction, or texture. These key feathers are interpolated to create a continuously varying set of feathers as described by Streit and Heidrich [SH02] and in Chapter 5. Then, for each follicle the set of key feather interpolants and their weights must be determined. To this end a 2D Delaunay triangulation is created from the key feather follicles in the parameter space. Each follicle is then located in this triangulation using its parametric coordinates. The barycentric coordinates of the follicle in this triangulation are used as the weights for interpolating between the key feathers associated with the triangle’s vertices. Recall the barycentric coordinates (u, v, w) for a point p can be obtained from the vertices of the bounding triangle (p_1, p_2, p_3) by:

$$u = \frac{\text{area}(p, p_2, p_3)}{\text{area}(p_1, p_2, p_3)}, \quad v = \frac{\text{area}(p, p_3, p_1)}{\text{area}(p_1, p_2, p_3)}, \quad w = 1 - u - v$$

Figure 6.9 shows the key feather triangulation and the set of follicles. The new feather F at location

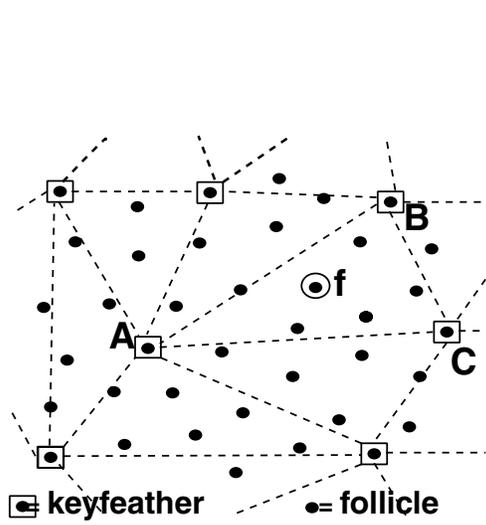


Figure 6.9: Follicles and key feather triangulation for computing interpolation weights.

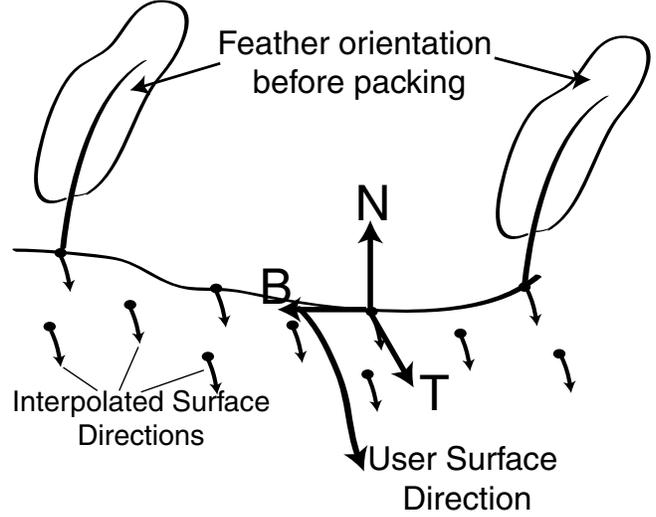


Figure 6.10: User defined direction vector, per follicle interpolated direction vectors and feather coordinate frame (Normal, Tangent, Bitangent) (N, T, B).

f , is determined from key feathers K_A, K_B, K_C at locations A, B, C respectively, as:

$$F = \frac{\text{area}(A, f, B)}{\text{area}(A, B, C)} * K_C + \frac{\text{area}(A, C, f)}{\text{area}(A, B, C)} * K_B + \frac{\text{area}(C, F, B)}{\text{area}(A, B, C)} * K_A$$

This is simply a barycentric weighting of the key feather parameter space.

6.3 Step Three: Feather Direction and Orientation

Feather direction and orientation is the second element to the feather arrangement component. The direction and orientation of the feather can greatly affect both the form and colouration of the resulting feather coat. The follicle parameterization and user-placed direction vectors are used to compute the surface direction at each follicle position. An efficient packing method, as outlined below, is then applied to determine the final feather orientation. The approach used decouples the mechanism used to compute versus control the final orientation of the feathers. These two mechanisms could possibly be the same as with some hair styling models [HMT00] where streamlines were computed from user-placed vortices, sources and sinks and an interpolation of the streamlines defined the hair direction. Our approach automatically computes the feathers' orientation given the feather structure and surface characteristics. Styling of the feather coat is

6.3. STEP THREE: FEATHER DIRECTION AND ORIENTATION

independent of the initial orientation or packing algorithm. The approach used to control the orientation or style the feathers will be discussed in Chapter 8.

Hair, fur, and plant models have also addressed orientation issues. Direction and orientation is generally unspecified in hair models due to the bilateral symmetry and pliability of hair. The symmetry implies that orientation is not discernible. The lack of stiffness means that the strands curve (dynamically) due to collision (with other hairs and the object), clumping, gravity, natural curliness etc. [DTKT93, WS92], which compensate for any discrepancy in the initial direction of the hair. Direction and orientation of hair has also been specified using weighted streamlines of flow fields [HMT00]. Gelder and Wilhelms [GW97] address short stiff fur direction by dividing the body of the animal into body elements and have the user specify body element vectors which when combined with the surface normal and a stiffness coefficient form a hair direction. Plants also have directional properties for their branching structures and organs. L-systems help to specify the orientation using a series of rotations for each branching element to determine the final orientation of the apical organ. The orientation is similar to hair in that the curvature along the branch or strand defines the final orientation. As the plant grows the orientation of the apical organs can change due to environmental conditions (tropisms) and growth factors.

However models for hair, plants and fur are not appropriate for feathers because their characteristics are quite different. While hair is pliable and changes curvature dynamically to alter the orientation, feathers have predefined curvature and are fairly stiff. While plants define orientation through growth, feather orientation is predefined. While fur and some hair orientation is not discernible, changes in direction and orientation of feather are very noticeable. Changes in feather orientation can occur by altering both the direction and orientation of the feather from the follicle. Like plants, alterations in orientation of branching structures can be used to alter the curvature of the feather, but does not help in defining the orientation of the entire feather with respect to the surface.

The direction and orientation of feathers can be specified in many ways. One method is similar to that of fur, utilizing the normal to the surface, the body element axis and possibly the curvature or shape of the feather to compute the direction. A second method may involve generating a flow field and using the streamlines for feather direction and orientation as was done with hair [HMT00]. If the flow field is constructed from the aerodynamics of a bird in flight it would parallel the ornithologists observation that feathers are oriented toward the posterior end of the bird in a normal flight position (see Figure 2.1). While this is an interesting approach, not all feathers are aligned with streamlines, especially with synthetic bird

characters and since the feathers in pterylae regions must cover portions or all of adjacent apteria region extra control may be needed and thus we will use the former method. A secondary process may also be used on an initial orientation to automate the process of covering adjacent apteria.

Once a desired feather direction is defined a method to compute the final orientation taking into account various factors, such as collisions and/or coverage of apteria areas is needed. The three factors that affect the resulting feather orientation the most can be identified as follicle angle, feather-to-feather collisions and feather-to-body collisions. Since the orientation of a single feather is not independent of the rest of the coat, a change in one feather's orientation can have a large affect on the neighbouring feathers. A solution for handling collisions must not only determine when a collision occurs, but also determine a proper collision response. Determining collisions and responses have been used for simulating the dynamics of many materials including cloth [HB00] and hair (characteristic hairs [DTKT93], hair wisps [WS92] or strips [KH00], volumetric probabilities [HMT00]). However, using the feather's structural information, an optimized technique to compute an initial feather orientation at interactive rates is created. While our method may not be as accurate as some collision detection schemes that use full original geometry, it is comparable with techniques that use simplified geometry [CXGS02] and is not only much faster than these methods, but uses the flexibility of the feathers to improve the realism of the results.

Our method for orienting the feathers is based on the assumption that the size of the feather is small relative to the feature size of the underlying surface. This means that the surface is smooth and the area covered by the feather is assumed to be locally close to planar. Both of these assumptions are reasonable since real birds tend to have smaller feathers in high curvature regions. The feathers are packed by traversing the feathers in layered order from bottom to top. The order is determined from the parameterization of the tract. Our method is based on the fact that the feathers are layered consistently from bottom to top, that the shaft is generally stiff and does not bend under the weight of the feathers and that the barbs are more flexible and may deform to generate a close packing. As the feathers are traversed in order, a subset of the already packed feathers are used to extract data points to pack each feather against. Each feather is then rotated maximally in a coordinate frame defined by the user assisted surface direction and the key barbs are adjusted as needed. Our method consists of three steps: determining the surface directions and a coordinate frame, computing a maximum shaft rotation and adjusting or elevating the barbs.

6.3.1 Surface Directions and Coordinate Frame

In order to determine an initial orientation of each feather compliant with what the designer of the key feather had intended, a coordinate frame must be constructed local to each follicle. Each follicle already has a individual normal vector. However, this is not enough to uniquely orient the feather. A surface direction vector is used for a second dimension.

The surface direction is computed either using the medial axis of the body segment, or a small set of user specified surface styling vectors. The medial axis of the body segment can be automatically computed, or as in our implementation roughly specified by the user. By projecting this axis to the surface at the follicle location it defines a surface direction vector. As already noted by Gelder and Wilhelms, fur tends to lie in the direction of medial axis for each body segment - this is also the case with feathers, except for the wing feathers. Similarly, by using averaged bias vectors to cluster the follicle distribution the user can specify a few direction vectors which are averaged to generate a surface direction for every follicle. Each surface direction is computed by projecting an interpolated object-space direction vector onto the surface at every follicle. These surface styling vectors can add more control to the desired feather direction.

As shown in Figure 6.10, a coordinate frame for each follicle is defined as the local surface normal N , a tangent vector (the interpolated direction vector on the surface) T and a bitangent B from the cross product of these. This coordinate frame is used to orient the feather. The feather is drawn with the vane in the NB -plane as shown in Figure 6.12. To compute a resting or packed position for the feather, it is rotated about the bitangent axis from the surface normal toward the tangent or surface direction, as explained next.

6.3.2 Shaft Rotation

Traversing the follicles in sorted order, each feather f' is packed using the control vertices of the key barbs and shafts from a subset of the surrounding feathers \mathcal{F} . This subset contains all the data points of those feathers whose follicles lie in the bounding volume of f' when oriented with the surface direction. As a bounding volume we simply use the projection of the 2D feather vane on the surface as illustrated in Figure 6.11. Once the set of data points \mathcal{F} is determined, they are projected to f' 's NT -plane (*shaft slicing plane*) as shown on the left in Figure 6.12. An orthogonal projection parallel to the NT plane is used. From the projected points a height field is computed from the distance between the projected points and the vector T . Using the length of the feather shaft and the height field, a maximum angle of rotation about the bitangent is computed for the shaft orientation. The angle is maximum such that the feather shaft (curvature

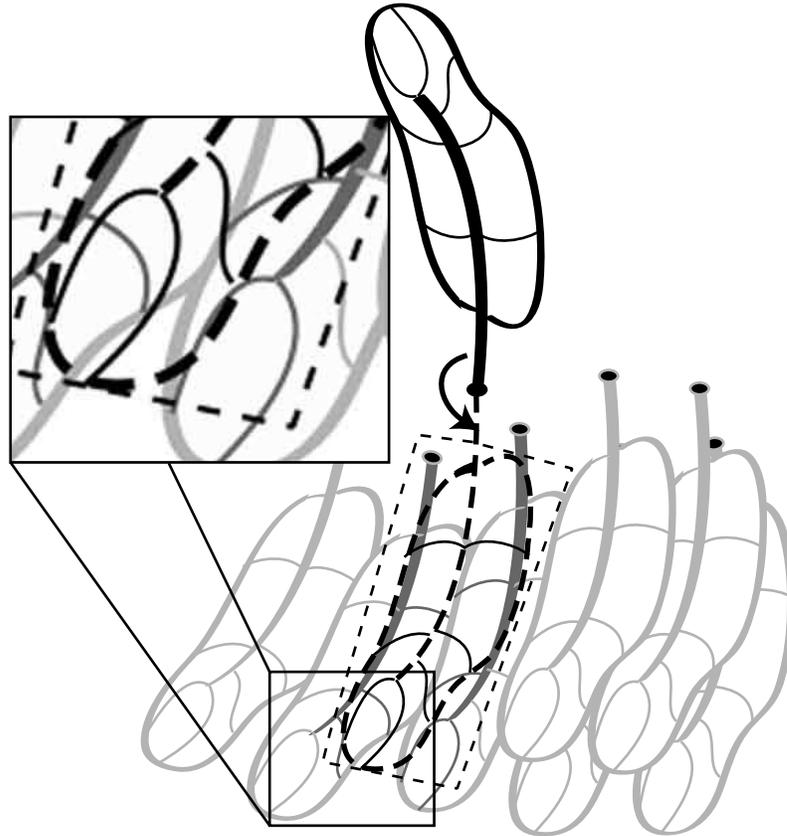


Figure 6.11: Data point determination for packing of feathers during shaft rotation stage. Bounding box is dashed black box. Feather to be packed is drawn in black. Previously packed feathers are in light gray. The candidate data points (curves) shown in dark gray.

and length) does not intersect the boundary of the height field. In cases where \mathcal{F} is empty, the feather is rotated 90 degrees less an allowance for the feather rachis curvature and surrounding polygon orientation, in order to prevent the feather from interpenetrating the model's surface (skin).

6.3.3 Barb Elevation

Once the shaft rotation is determined, the key barbs are checked to ensure given their curvature and length that they do not interpenetrate the surrounding feathers. To this end, we employ an algorithm similar to the one we used for orienting the shaft. For each key barb a subset of the points in \mathcal{F} is projected to a plane (*barb slicing plane*) defined by the surface normal and direction along the barb, T' as shown on the right in Figure 6.12. For key barbs whose curvature does not lie within this plane, a direction that bisects the

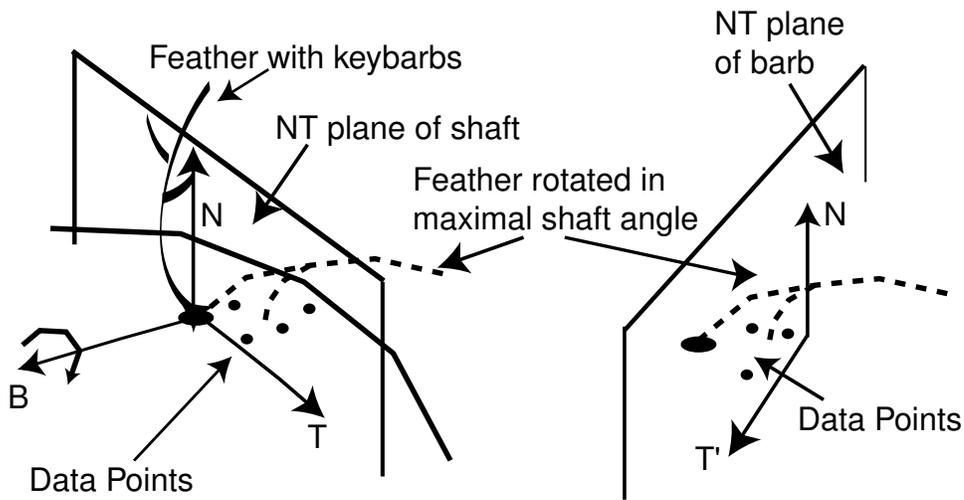


Figure 6.12: Slicing planes used for packing. Shaft slicing plane for rotation of the shaft shown on the left and barb slicing plane for elevation of the barb control vertices shown on the right.

key barb control vertices is used as an approximation to the barb direction. Again the candidate points are projected to the barb slicing plane using an orthogonal projection parallel to the barb slicing plane. The subset of points chosen for each of i key barbs k_i are those points of \mathcal{F} that lie on the same side of the shaft slicing plane as k_i and are between k_{i-1} and k_{i+1} barb slicing planes. This effectively creates a rectangular prism from which to select candidate points. The volume is defined by the shaft slicing plane, a slicing plane parallel to this at the feather's bounding box edge and two barb slicing planes, one before and one after the candidate barb to be adjusted. If this subset is empty the key barb is not adjusted. Otherwise, again a height field is computed using the projected points and the barb direction vector. However rather than computing a rotation angle the control vertices of each key barb k_i are fit to the height field. Again we do not consider any points in the height field which lie beyond the length of the key barb.

To fit the control vertices of barb k_i to the height field, the height field is divided into subsets according to the location of k_i 's control vertices. A combination of the global maximum in the height field and the local maximum of each segment is used to determine how much to elevate each of k_i 's control vertices. The global maximum is used as a minimum to prevent the barbs from bending in unnatural configurations. The barbs are allowed to flatten but never take on inverse curvatures. The control vertices are only adjusted in the barb slicing plane, and thus the lateral curvature of the barb is still generally preserved, in order to maintain the shape and overall character of the vane. To assist in the packing of the feathers, the default barb at the tip of the feather model is replaced by an actual key barb at the tip with the same curvature as the default.

6.3.4 Packing Summary and Result

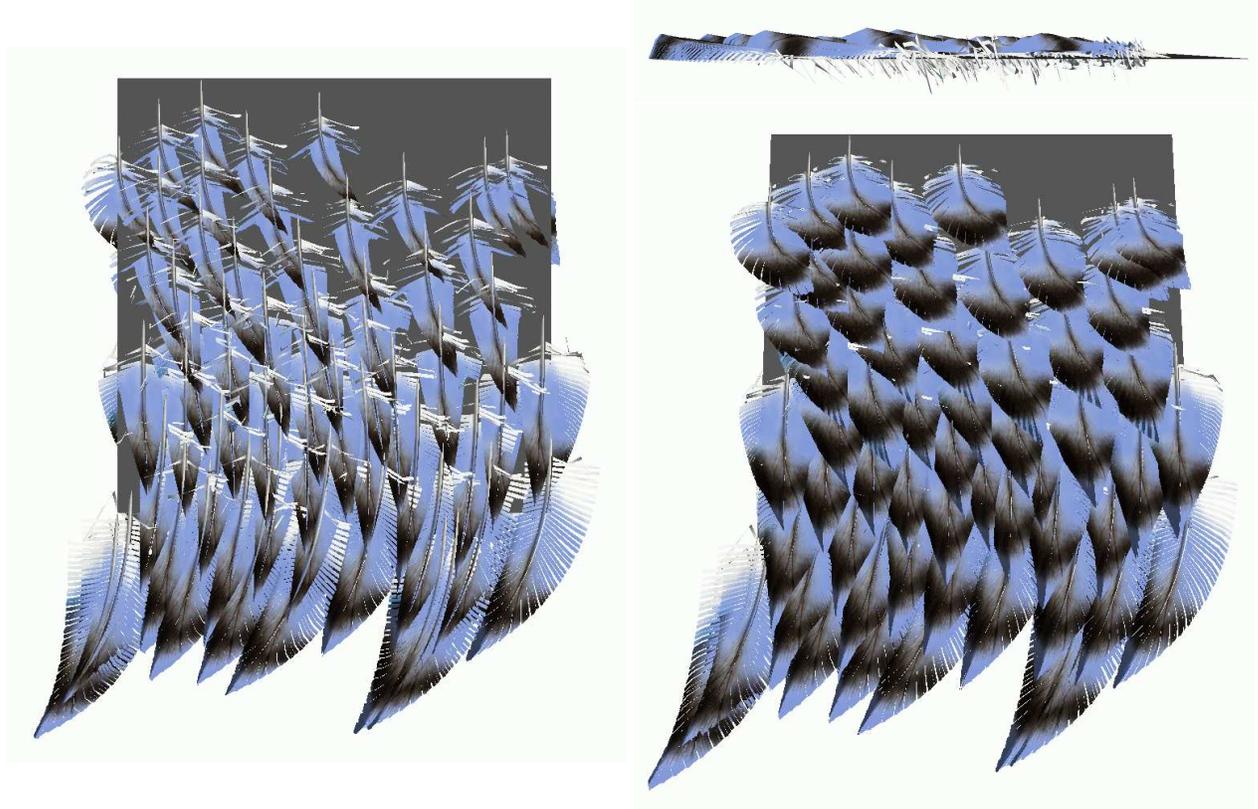


Figure 6.13: Results of the packing algorithm **Left:** The feathers position along the tangent vector, without packing. Notice the interpenetrations with both the surface of the model (skin) and other feathers. **Top Right:** side view and **Bottom Right:** front view of the same feather collection after packing. Notice the lack of feather-model and feather-feather interpenetrations.

The packing algorithm is summarized as follows. For each feather f' of the feather tract in sorted order:

- The candidate set of control vertices \mathcal{F} from already packed feathers are determined given the shape and size of the feather and the feather coordinate frame defined by the normal and surface direction.
- The maximum rotation angle about the bitangent is determined for the shaft using a projection of the candidate points \mathcal{F} into f' 's NT plane as a height field.
- For each key barb k_i of feather f' in rotated position:
 - A set of points $\mathcal{P} \subseteq \mathcal{F}$ are determined such that they lie on the same side of the shaft as k_i and are located between the NT' (barb slicing) planes of k_{i-1} and k_{i+1} .
 - The set of points \mathcal{P} are then projected to k_i 's NT' plane and a height field is determined.

- The key barb control vertices are fit to the height field minimizing inverse curvature.

Figure 6.13 shows an example of the results of our packing algorithm. On the left feathers are aligned with the surface direction. Notice how they interpenetrate both the model surface (i.e. skin) and adjacent feathers. On the right the same feather collection has been packed using our algorithm - on the top a side view and on the bottom a front view. Notice how there are no feather-model or feather-feather interpenetrations. The rendering of these feathers is texture mapped and is discussed further in Chapter 7.

6.4 Analysis of the Algorithm

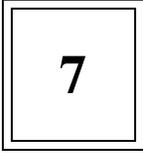
The use of a sample set of data points for packing is adequate since the shape of the feather is defined solely by the key barbs, which are defined by the control vertices of Bézier curves. All other barbs are interpolations of these control vertices. Hence, the key barbs form the bounding box of the feather, and due to the convex hull property of Bézier curves, these control vertices form a bounding volume of the feather. Therefore, only this small set of data points needs to be considered in the packing process. As well, since the control vertices of the key barbs are consistent across the geometric levels of detail as outlined in Chapter 5 and Chapter 7, the packing is independent of the level of detail.

Once all of the key barbs have been adjusted for the candidate feather f' , the next feather in layered order is packed in the same manner - first determining the shaft rotation and then elevating the barbs. Only one pass is done over each feather in the feather tract.

6.5 Summary

This chapter presented the details of how to form a collection of feathers from a subset of feather types. There are three steps to creating the coat: follicle distribution, feather generation and feather orientation. The two main elements explained in this chapter were the follicle distribution and the feather orientation. First, the follicle distribution involves appropriately segmenting the polygonal mesh into tracts and then specifying densities and possibly biasing vectors to create the final follicle distribution. Once the follicles are distributed, a parameterization is imposed, so that an ordering of the feathers can be defined. Key feathers are then added in selected locations and the interpolation methodology from Chapter 5 is used to create the intermediate feathers automatically. Secondly, an orientation for the feathers is devised. This is done using some user-specified surface styling vectors and the parameterization that defines the feather

layering. Finally, a packing algorithm is executed that uses a subset of data points in order to achieve a reasonably tight layering in an efficient manner. The performance results of this algorithm will be analyzed further in Chapter 9.



Level of Detail and Rendering

Once the structure and geometry of the feathers have been constructed, a method of rendering the feathers is required. While this is not the focus of this work, it is nearly impossible to decouple rendering from the modelling methodology, especially at certain levels of detail. Also, an adequate rendering methodology is required to properly evaluate the results. Until this point in this thesis, where possible, structural examples and illustrations have been used to help explain our feather coat model. This chapter will outline the rendering methodology and show examples of rendered feathers. Implementation details will be outlined where needed to help explain the performance results that will be presented in Chapter 9. This chapter will concentrate on the individual feather rendering model, as it was designed to be independent, so that minimal effort is required to render many feathers. A combination of texture mapping, alpha blending and an illumination model were incorporated into the rendering model.

7.1 Feather Rendering

As mentioned previously as part of the thesis statement, there is a need and a desire for a multi-resolution methodology. Chapter 5 mentions the ability to alter levels of detail by altering the amount of geometry. Our rendering model augments the spectrum of available resolutions.

The basic rendering model consists of texture mapping the feather geometry with an acquired or synthesized feather texture. Alpha blending is added to simulate the transparency of the feather. Finally, an illumination model based on the anisotropic model of Poulin and Fournier [PF90] was added to simulate

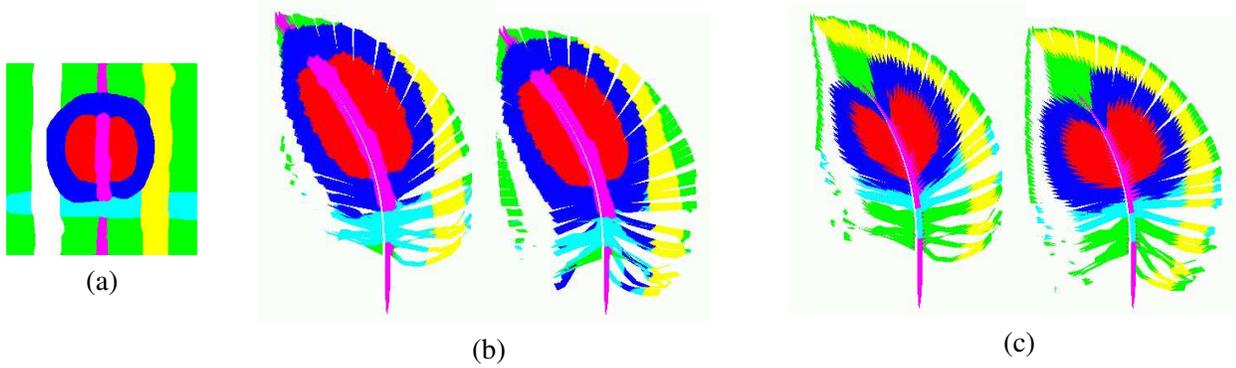


Figure 7.1: Different methods of mapping the texture to the geometry of the feather. (a) The texture used (b) texture mapped using feather parameterization left: original feather right: modified base barb angle (c) texture mapped using barb parameterization left: original feather right: modified base barb angle.

the “shimmering effect” observed due to the alignment of barbs and barbules. The illumination model was implemented as a series of textures in order to utilize graphics hardware to obtain interactive rendering rates.

7.1.1 Textures

Application of a texture to the feather geometry requires a parameterization (u,v) of the geometry. Since Bézier curves are used for modelling both the shaft and the barbs, a parameterization is inherent. The only decision is how to use the given parameterization. The texture can be mapped in three ways: globally to the feather, locally to each barb or relative to growth.

Using the local coordinate system of the feather with normalization produces a parameterization global to the feather. However, a few subtleties exist. Given a point p local to the feather structure, only two dimensions of the feather are used, p_y being proximally along the shaft, and p_x perpendicular to this in the plane of the vane. Then, given a point p :

$$v = v(p) = p_y / (ShaftLength + MaximumBarbLength)$$

$$u = u(p) = \begin{cases} 0.5 - p_x / (2 * MaximumBarbLength), & p_x / (2 * MaximumBarbLength) < 0.5 \\ 0.5 + p_x / (2 * MaximumBarbLength), & p_x / (2 * MaximumBarbLength) \geq 0.5 \end{cases}$$

where $MaximumBarbLength$ is the barb with the maximum length. In this parameterization the curvature of the feather shaft is ignored. This means that the texture is effectively glued to the feather as curvature is added or removed from the feather shaft (bending of the feather). However, this also means that as the key barbs are changed and the feather barbs are bent, the texture “floats” over these barbs. This may or may not

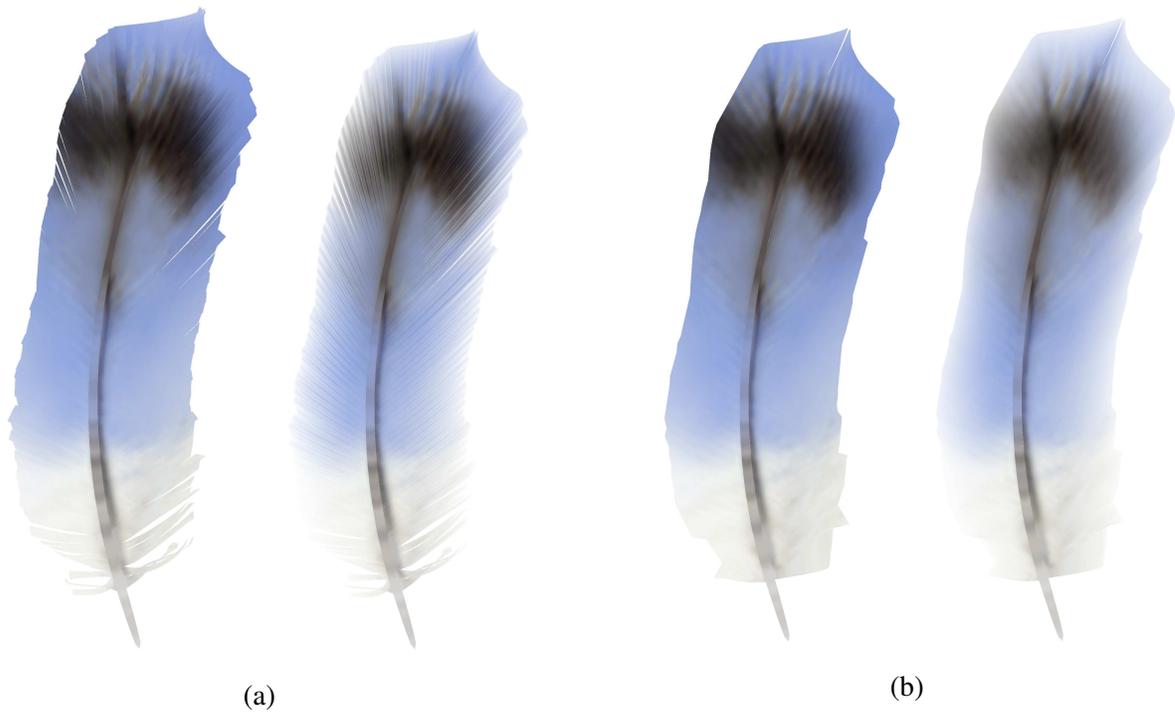


Figure 7.2: Feathers with and without alpha blending **(a)** full geometry *left*: without alpha blending *right*: with alpha blending **(b)** 60% of the geometry *left*: without alpha blending *right*: with alpha blending, but not using the alpha texture.

be desirable depending on whether the user is trying to maintain the scanned texture or the structure of the feather through modelling.

If a parameterization local to the barbs is used, this latter effect is overcome. In this scenario v is mapped the same as above. However, u is taken directly from the parametric value of the barb with an adjustment for the left and right vanes and shaft width. This causes the texture to be glued to the barbs rather than “floating” on them. Thus, as the angle of the barbs change, the texture moves with them. This is desirable when pigment or pattern locations are known relative to the barb and changes in pattern are being analyzed with alteration in barb angle.

The final scenario would be to map the texture to the parametric values of growth, where v is mapped along the shaft as the feather grows, and u is mapped around the cylinder at the base of the cone that forms the pin feather (refer to Figure 2.10). This approach is useful if the pigment concentrations at the follicle are known or being analyzed over the growth of the feather.

7.1.2 Alpha Blending

In order to add realism and to simulate the effect of transparency, alpha blending is added to the rendering model. At the highest level of geometric detail, each barb is constructed from two triangle strips. An alpha value is given to each vertex along the barb, with a quadratic decrease near the tip of the barb. The decrease in alpha value is started one segment earlier on the outside edges than the middle to simulate higher opacity of the barb versus the barbules and lower density of barbules near the tip of the barb. Since a single polygon represents multiple barbs, at lower levels of detail an alpha texture was constructed with these same characteristics, and simply wrapped multiple times, depending on how many barbs the polygon represents. Figure 7.2 shows a feather with the original texture, and alpha blending added at full geometry and 60% of the geometry. To further enhance the rendered detail a reflection model is used in addition to the alpha texture.

Using alpha blending also adds some subtleties to the drawing order of the feather. In order to blend correctly between barbs and between vanes, four decisions must be made: to draw the barbs in a top-down or bottom-up order, to draw the vanes in a left-right or right-left order, to draw each barb proximally (tip to shaft) or distally (shaft to tip) and to draw the afterfeather or feather first. These decisions are based on the viewing direction. For example, if the viewer is looking up the shaft of the feather from the front, closer to the right vane than the left, then the barbs would be drawn in top-down order and the left vane would be drawn first with the barbs in tip to shaft order followed by the right vane drawn in shaft to tip order. Whichever substructures of the feather are furthest from the viewer must be drawn first, followed by progressively closer feather substructures. Since the geometry of each feather is never stored and generated at run time, a polygon sorting method (z-buffer) would not work and thus, the geometry must be carefully generated in view dependent order at run-time. A hemisphere around the feather with a vector to the viewer can be used to determine the drawing order. Depending on which octant the viewing direction is in, the order differs. This ordering is consistent as long as the viewer is not too close to the feather (i.e. the entire feather is visible). Otherwise, if only a portion of the feather is visible, the order is ensured correct by using the local viewing direction.

7.1.3 Levels of Detail and Reflection Model

At levels of detail where the individual structure is not obvious, a method similar to that of Dischler and Ghazanfarpour [DG94], or Wong et al. [WNH97], where texture is simulated from geometry, is useful. Of

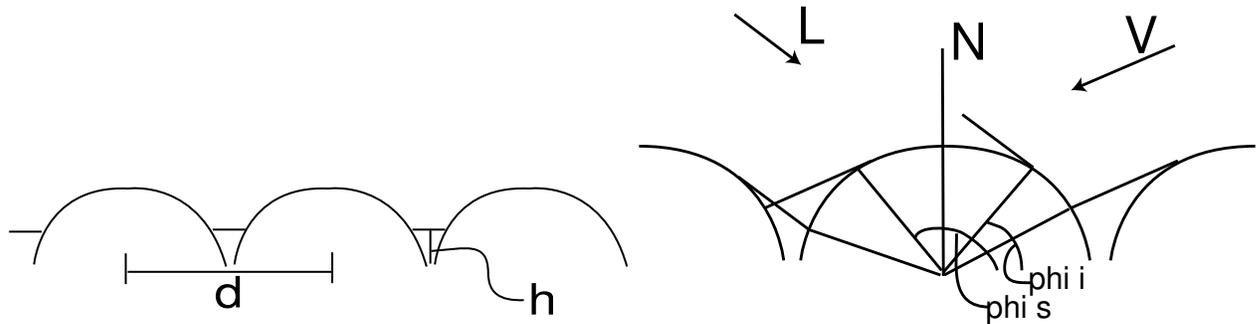


Figure 7.3: Right Parameters d and h of the anisotropic reflection model. Left: shading and viewing angles showing lit and visible portion of cylinder.

course, this still initially requires modelling the geometry, and then post processing to reduce the complexity for rendering. To achieve level of detail rendering we gradually replace the smaller substructures with a reflection model. This approach is reminiscent of the approach Meyer and Neyret [MN00] use for pine trees, in which geometry from higher levels of detail is used to generate a reflection model for lower levels.

The reflection model was chosen based on the shape of the feather substructures. Although both the barbs and barbules taper distally, they can be roughly approximated by cylindrical segments. Also, since pennaceous vanes (i.e. contour feather vanes, especially flight feathers) have aligned barbs, using a set of aligned cylindrical elements for an anisotropic reflection model is appropriate. This most closely parallels Poulin and Fournier's [PF90] anisotropic reflection model.

The Poulin and Fournier model approximate anisotropy with a large number of adjacent small cylinders. And thus, since the cylinders have a small radius in comparison to their length, the intensity reflected off each cylinder can be approximated by the reflection off a cross section of any one cylinder. They orient the cylinders using the coordinate frame N, T, B where N is the normal to the cylinder, T is the tangent along the length of the cylinder and B is the cross-product $T \times N$. So the cross-section of the cylinder whose reflectance is to be computed lies in the plane NB . This model works for lower levels of detail, because the cylinders are assumed to be small enough (or the viewer far enough) that individual cylinders, or in our case barbs are not discernible. Thus, the viewing and lighting directions can be assumed to be constant over each cylinder.

In their model they control the amount of anisotropy by the use of two parameters d and h as shown in Figure 7.3. The anisotropy is caused by the difference in the distribution of surface normals between the cross-section of the cylinders and along the length of the cylinders. The parameter d represents the distance

between two adjacent unit radius cylinders. Since the intersection point on the cylinder defines the normal, a value of zero results in simply using the surface normal and a value of two means the variation in normals is π . Any values greater than two result in a floor being formed between the two cylinders, which in our case contributes nothing to the reflectance model, because this would represent the space between the two barbs or barbules. The parameter $h \in [0..1)$ controls the height of the floor and thus the height of the cylinders. While this can be varied in our implementation, a value of zero was generally assumed. As Poulin and Fournier note, due to the parameters d and h the normals on the cylinder can only vary within a certain angle θ_M from the surface normal:

$$\theta_M = \min(\phi_d, \phi_h) \quad \text{where} \quad \phi_h = \cos^{-1}(h) \quad \phi_d = \begin{cases} \sin^{-1}(d/2) & \text{if } d < 2 \\ \frac{\pi}{2} & \text{otherwise.} \end{cases}$$

Depending on the orientation of the light vector L , some parts of the cylinder will be lit and others unlit. If L^p is the light vector projected into the NB plane of the cylinder then ϕ_{L^p} is the angle between the surface normal and L^p . The self-shadowing angle corresponds to the angle from the surface normal at which the cylinder blocks light from itself. The shadow angle is the angle from the surface normal at which the neighbouring cylinder blocks light. For any floor, some length of it is lit and the rest is in shadow. We refer the reader to the original paper for computation of the shadow and self-shadow angles as well as the lit and unlit floor lengths. The same logic can be used to obtain the self-hiding angle and hiding angle as well as the length of the hidden floor and the length of the visible floor. Where the self-hiding angle is the angle at which a cylinder hides itself from the viewer and the hiding angle is the angle at which a neighbouring cylinder hides the cylinder from the viewer. The portion of cylinder and floor contributing to the reflectance is the portion that is both lit and visible.

To compute the reflection, the lighting and viewing directions are projected into the NB -plane of the cylinder, which in our case represent the barbules. A diffuse and a specular term is computed from these projected lighting and viewing directions. If L^p is the projected light direction and V^p is the projected viewing direction in the coordinate frame N, B, T then both have components $L^p = (L_n^p, L_b^p, L_t^p)$ and $V^p = (V_n^p, V_b^p, V_t^p)$. Then as in many reflectance models we can compute the diffuse term as $N \cdot L$ as:

$$\int_{\phi_i}^{\phi_s} (N \cdot L) \cos(\phi_{V^p} - \phi) d\phi \quad (7.1)$$

where $\cos(\phi_{V^p} - \phi)$ is a correction factor due to the viewing position and ϕ_i to ϕ_s is the range of the cylinder that is both lit and not in shadow as shown in Figure 7.3. Then using $(\sin \phi, \cos \phi, 0)$ for N equation 7.1

equates to:

$$\frac{L_n^p}{2} \left(\begin{array}{c} \cos \phi_{VP} (\sin^2 \phi_s - \sin^2 \phi_i) + \\ \sin \phi_{VP} (\phi_s - \phi_i - \sin \phi_s \cos \phi_s + \sin \phi_i \cos \phi_i) \end{array} \right) +$$

$$\frac{L_b^p}{2} \left(\begin{array}{c} \sin \phi_{VP} (\sin^2 \phi_s - \sin^2 \phi_i) + \\ \cos \phi_{VP} (\phi_s - \phi_i - \sin \phi_s \cos \phi_s + \sin \phi_i \cos \phi_i) \end{array} \right)$$

which, due to the viewer correction factor, is normalized by:

$$\int_{\phi_i}^{\phi_s} \cos(\phi_{VP} - \phi) d\phi = \cos \phi_{VP} (\sin \phi_s - \sin \phi_i) + \sin \phi_{VP} (\cos \phi_s - \cos \phi_i)$$

The specular term was computed using an a Gaussian approximation to the falloff around the high-light with \mathcal{B} as the angle to half intensity:

$$\frac{2}{\mathcal{K}\sqrt{\pi}} \quad \text{where} \quad \mathcal{K} = \frac{\sqrt{\log 2}}{\mathcal{B}}$$

Once, the diffuse and specular reflectance is able to be calculated for the range of projected angles ϕ_{LP} and ϕ_{VP} for each cylinder the reflectance model is complete. In our implementation, we generated a 2D texture map which contains the BRDF values of this model for all possible combinations of angles between binormal B , and projected light direction L^p , $\phi_{LP} \in [0..\pi]$, as well as binormal B and projected viewing direction V^p , $\phi_{VP} \in [0..\pi]$. This approach is similar in spirit to the work by Heidrich and Seidel [HS99] as well as Kautz and Seidel [KS00] based in spirit from [SZH97]. For each vertex in the feather geometry the projected light L^p and viewing V^p vectors and their corresponding angles from the binormal B were calculated and used as the u and v parameters of the texture map.

Figure 7.4 shows two feathers with our method of multi-texturing using three textures: a colour pattern texture, an alpha texture, and a BRDF texture for the barbules. The feather on the top has full geometry and the one on the bottom has 60% geometry. Compare the alpha blending results with the feather in Figure 7.2(a). The dark regions on the vanes when compared with Figure 7.2(a) are a result of the reflectance model. In both Figures 7.4 and 7.2 the feathers have been alpha blended using white as the background colour. As shown in Figure 7.4 the light is directional.

The rendering of each feather was done using three textures with four texture look-ups and hardware multi-texturing. The first texture was the actual feather colour. The second was the alpha texture to be applied to the barbs and the third texture was the anisotropic reflection texture. Two texture lookups were used with the third texture, one for the barbules attached to the right side of the barb and the other for the

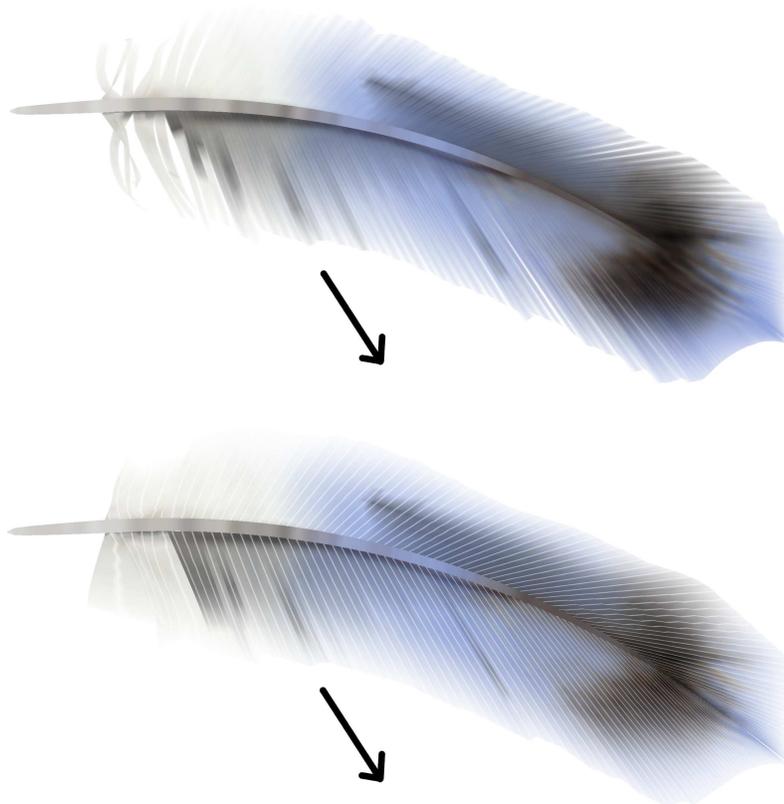


Figure 7.4: Feathers using our reflectance model with multi-texturing. **Top:** Full geometry **Bottom:** 60% of the geometry. Compare the barb alpha results with those of Figure 7.2(a). The darker portions on the barbs/vane of the feather are as a result of the Poulin/Fournier [PF90] reflection model.

barbules attached to the left side of the barb. Two texture look-ups were required since the NB -planes of the barbules are oriented differently and hence have different local projections of the light and viewing directions.

7.1.4 Other Levels of Detail

At closer levels of detail where the barbs are larger than polylines, geometry may be needed for the barbs, and a more sophisticated rendering model would be desirable. A rendering model that could account for pigment placement and internal structures to convey more of the illumination effects would be desirable at this scale. Current work at this level of detail is being done to render hair [MJC03]. This involves processes such as subsurface reflectance and scattering. While both of these are important in feathers, a complete rendering model for feathers is outside the scope of this thesis.

7.2 Rendering Feather Coats

Once, we have a rendering model for individual feathers, we can then apply it to render a feather coat. Due to the model design, not much alteration is needed for rendering feather collections. The only detail we must account for is the rendering order of the feathers, due to the alpha blending. However, because there is already an inherent ordering to the feathers from the layering arrangement, we simply render the feathers in the order they are arranged. Each feather is still drawn in view dependent order as defined in Section 7.1.2.

The other complication is order of rendering the model and feathers. If the model (skin) is rendered first, feathers that should be occluded by the skin may be alpha blended with the feather coat in the foreground. And, if the model is drawn last the feathers in the foreground will not be blended with the skin. To overcome this, a small z-buffer subtlety was used. The model is drawn first, writing to the depth buffer. While the feathers are drawn we allow the model to occlude the feathers by enabling the depth test, while disabling writing to the depth buffer so that the feathers are blended properly.

7.3 Alternative Rendering Styles

We also created an alternate rendering of feathers that illustrates their structure. A method using the location of the feather substructures was used to place scratches on a 2D canvas to simulate a scratchboard like medium. The technique involved altering the density and thickness of the scratch used to preserve relative tone in the resulting image. A result of this technique is shown in Figure 7.5. Since this technique is slightly tangential to the goal of this thesis the details have been omitted.

7.4 Summary

This chapter presented our methodology for rendering our structural model of feathers and feathers coats previously presented in Chapters 5 and 6. While rendering is not the main contribution of this thesis, a rendering model was nevertheless necessary for assessment of results. Since synthesizing feather patterns is an entire area of research on its own, our approach involved using acquired or synthesized feather patterns as textures applied to the feather geometry. An alpha blending step allowed for simulation of the presence of barbules and particular techniques were used to preserve this effect across multiple levels of geometric detail. To further enhance the rendered results, a reflection model was added to the geometry that not only

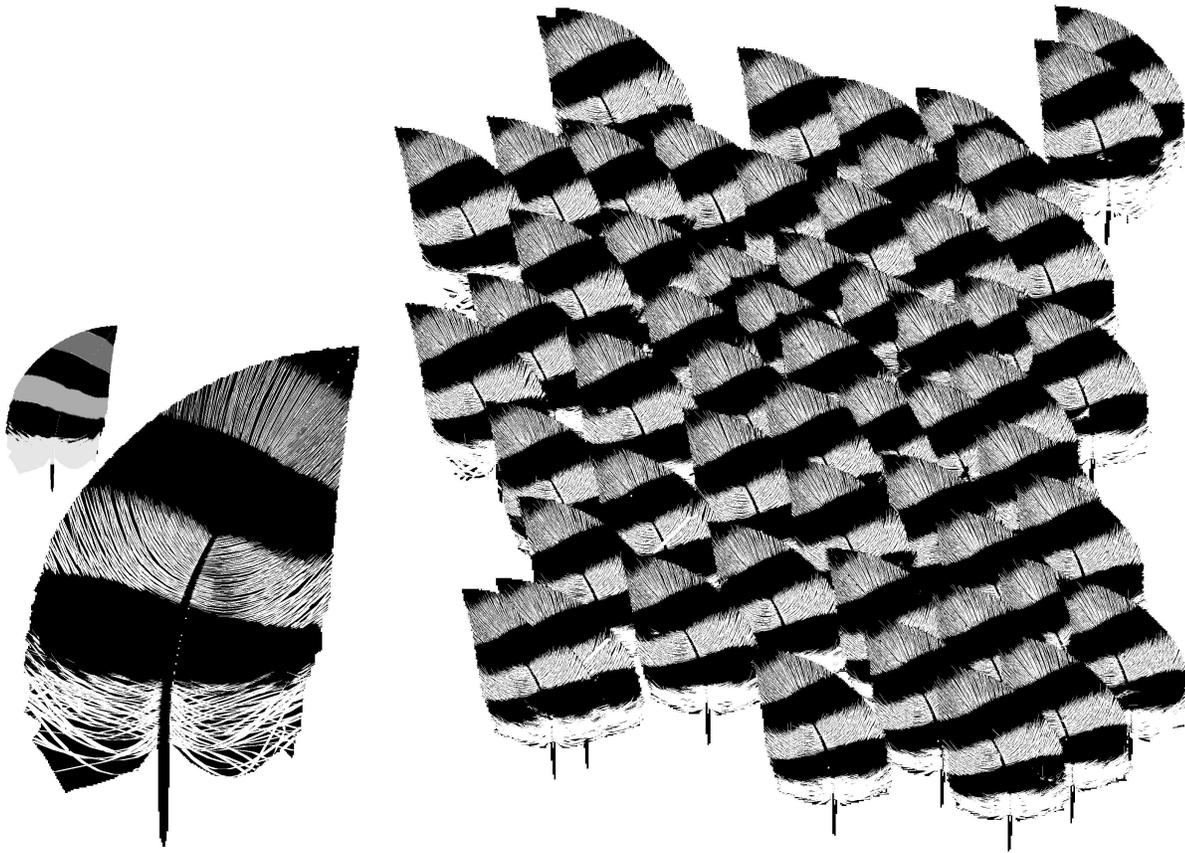
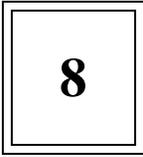


Figure 7.5: An alternative method for rendering the structural attributes of the feathers.

simulated the presence of barbs and barbules, but also mimicked the reflectance properties that a simplified version of these type of structures would possess. The simplified structure (cylinders) and the alignment between barbs and barbules were used to simplify the required rendering calculations.



Simulating Feather Alteration

The feather alteration component is composed of three elements *feather growth*, *feather replacement* and *feather rearrangement*. These three elements together define both how the feather coat alters its appearance (structure) over time and how it responds to movements of the skin and the underlying skeletal model. The development of the feather coat over time is the basis of feather coat morphogenesis. Ensuring the feather coat responds correctly to movements of the skin allows our feather coat methodology to be used on animated models. Since these alterations are integrated into the methodology this chapter does not add to the methodology, rather it provides as an initial validation of the robustness and objectives of the methodology.

8.1 Feather Growth

The first element of the feather alteration component is feather growth. Simulating the growth of the feather from the feather primordia, to pin feather, to final open feather (its death) and the eventual loss of the feather is strongly bound to the parameterization used to specify the general feather structure. Since the feather grows encompassed in a sheath, the canonical shape of the feather is that of a cone or tapered cylinder. As stated in Section 5.2.5 the feather growth is simulated by staggering the interpolation of the feather model parameters. The feather grows encased in a sheath until all substructures are nearly developed. The sheath is shed from the tip to the base. As the sheath sheds growth ceases in that region and the feather starts to open to its final form. From this information [LS72a, p.364-384] the interpolation schedule found in Figure 8.1 was constructed. The length and width of the shaft increases during the first 30% of the feather

8.1. FEATHER GROWTH

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. They help to maintain the canonical cone or tapered cylinder shape until the pin feather opens. 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. This key barb serves to interpolate between the pin feather shape and the final shape. 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. To summarize, the key parts of feather growth in order are:

- shaft lengthens and widens to its final length
- barbs lengthen while parallel to the shaft (they point toward the tip of the feather)
- barb spacing starts out with the final spacing as this reflects the attachment of the barbs to the rachis at a particular location
- the barb angle begins interpolation between 180° and the final angle
- Simultaneously in the last 20% of growth the actual key barbs start to be used for shape and the rachis obtains its curvature

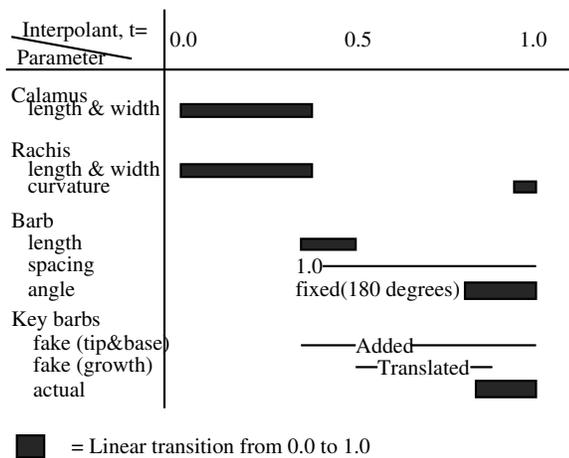


Figure 8.1: Sequencing of parameter interpolation for growth.

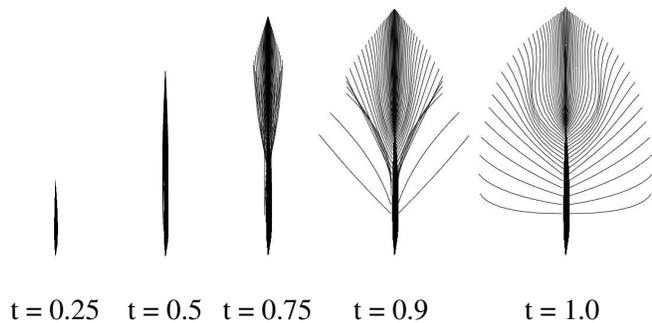


Figure 8.2: Simulated feather growth from interpolation sequence in Figure 8.1. Time steps at 25%, 50%, 75%, 90% and 100% of development.

This method of simulating growth generalizes the growth of any feather to first lengthen into a quill shaped formation of the required length and then as the straight, fake, 180° key barbs are migrated down the shaft of the feather, the feather gradually opens up from tip to base adding curvature to the rachis.

This means that packing non-opened feathers is approximately equivalent to packing only the shafts of the feathers. More will be discussed about packing of underdeveloped feathers in Section 8.3. An example of some of these feather growth stages are shown in Figure 8.2. Compare these results with Figures 2.10 and Figure 2.11.

The data used for growth was roughly approximated from Lucas and Stettenheim [LS72a, p.364-384]. The author is unaware of variations in growth rate between different feather types or the variation that may exist in the various parameters of feather growth, however, Lucas and Stettenheim state that at least in the Single Comb White Leghorn Chicken that the feathers moult at seemingly constant rates, however larger feathers take longer to regrow. Regardless, this information falls beyond the scope of this thesis. What is important is that our growth model is based on the substructures of the feather and by providing the right interpolation function and sequencing of these substructures, any rate of growth can be simulated. Computing the texture coordinates during growth is not an issue since, as with real feathers, the pigments are deposited in the barbs and shaft during growth. So, by the time the feather starts to open the pigment is already located in the barbs. Thus, we assign the texture coordinates in a final open position, such that the texture is attached to each barb and gets transformed with the barb transformations during the simulation of growth.

8.2 Feather Replacement

The second element to the feather alteration component is feather replacement. This element models the moulting process of feather replacement. For each feather tract the feather loss is specified as centre-outward or boundary-inward, except for the Alar tract which may be specified separately for the primaries and secondaries. With real feather coats the feathers tend to moult in *waves* over the coat, where a patch of feathers moult at once, thus they may not be perfectly sequential like the primary and secondary moult patterns. Given the feather loss ordering this element cooperates with the feather growth process to renew the feather coat.

Our method of feather replacement allows the user to specify a replacement sequence for each tract. Within each tract the feathers can be replaced centre-out, boundary-in, centre-out u, boundary-in u, centre-out v, boundary-in v, increasing or decreasing u, increasing or decreasing v, or random. Then according to the parametric location of each feather and the type of moult sequence chosen a distance d , from moult

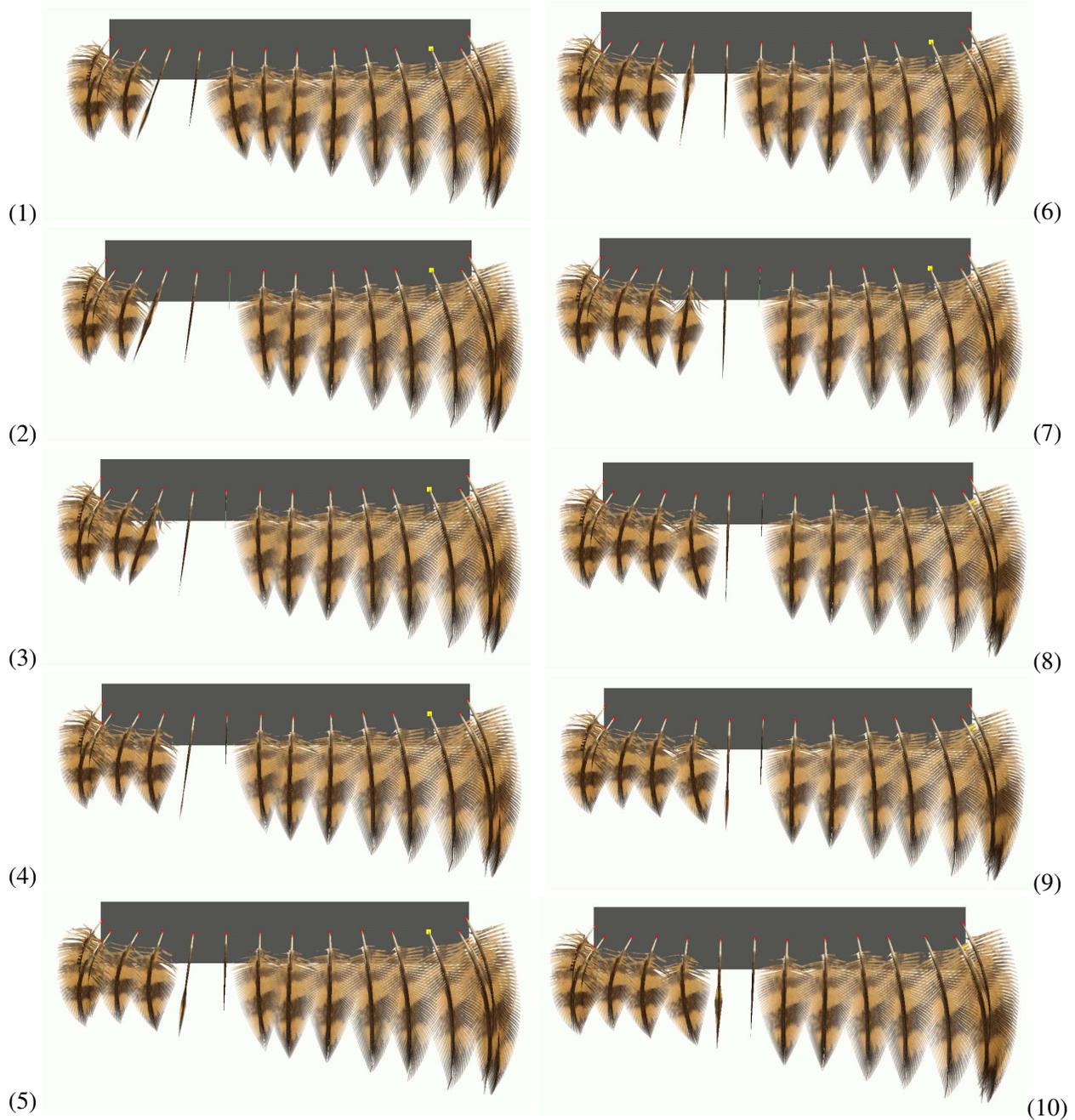


Figure 8.3: Moulting of feathers sequentially. Ten sequential, non-consecutive frames from a feather moulting animation. Note how feather growth and loss of adjacent feathers begins before each feather is completely open. This sequential moult is similar to the sequential moulting of the primary and secondary flight feathers.

starting position in parametric space is computed, where $d \in [0..1]$. This in turn provides an ordering for the moulting, which may or may not be perfectly sequential depending on the moulting pattern chosen. Then an overlapping window of width w is moved over the feather in moulting sequence simulating the wave of moult. The window width w determines how long the feather takes to grow. The number of feathers found within the window determines how many feathers develop simultaneously. The width of the window can be a function of parametric space (i.e. feathers take longer to grow near the boundaries of feather tracts) or relative size of the feather (i.e. larger feathers develop more slowly) or any other information.

Figure 8.3 shows ten frames of a moulting sequence. The parameterization is specified in increasing v , where v is the horizontal axis increasing to the right. In the first frame the feather third from the left is starting to open and the feather to the right of it is still developing its pin feather state. In frame two, both of these feathers develop further and the feather to the right of them has just been lost. In frame three the feather third from right is now more fully open, but the shaft remains straight. It is not until frame four that this feather obtains its final curvature. The two adjacent feathers to the right continue in a similar manner.

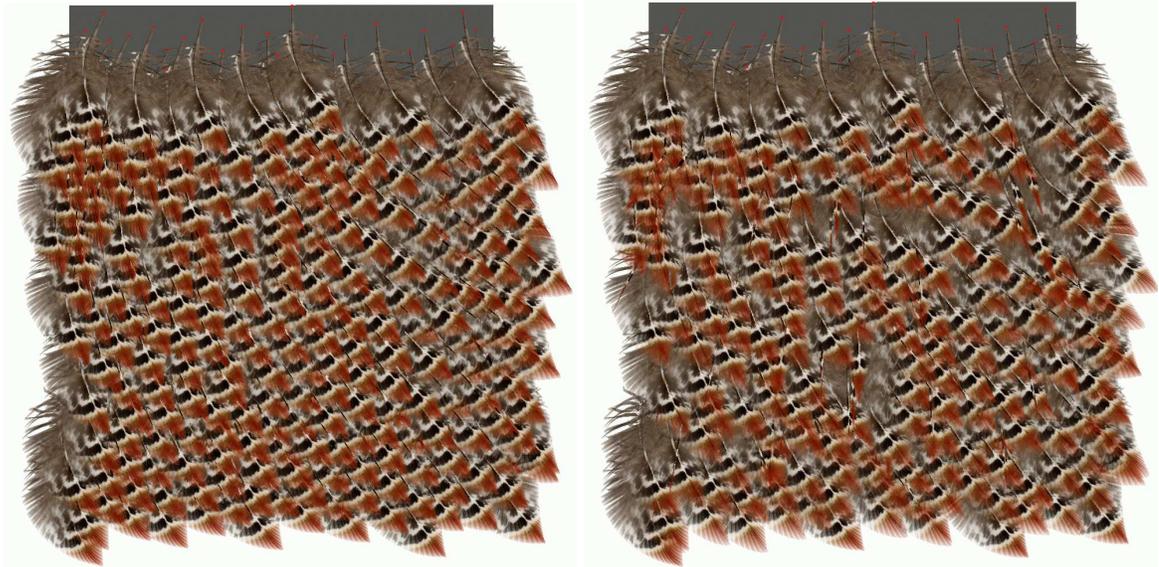


Figure 8.4: Moulting feather arrangements. **Left:** All feathers are mature. **Right:** Random feathers are at various stages of development. Note the difference in feather pattern.

Figure 8.4 shows a feather arrangement with and without random moulting. While this sequence is rare, it does occur and shows how the pattern and overall appearance of the feather coat can be altered due to moulting even with random moult. As shown on the right, there are “patches” where the underdeveloped feathers do not cover the feathers underneath as much as they would if they were fully developed as shown

on the left. Therefore, the pattern at the proximal end of the feather becomes more visible. In this example, the gray plumulaceous base of the feather is revealed causing patches of gray to be present among the red/black/white feather pattern, which are not visible in the fully developed feathers on the left.

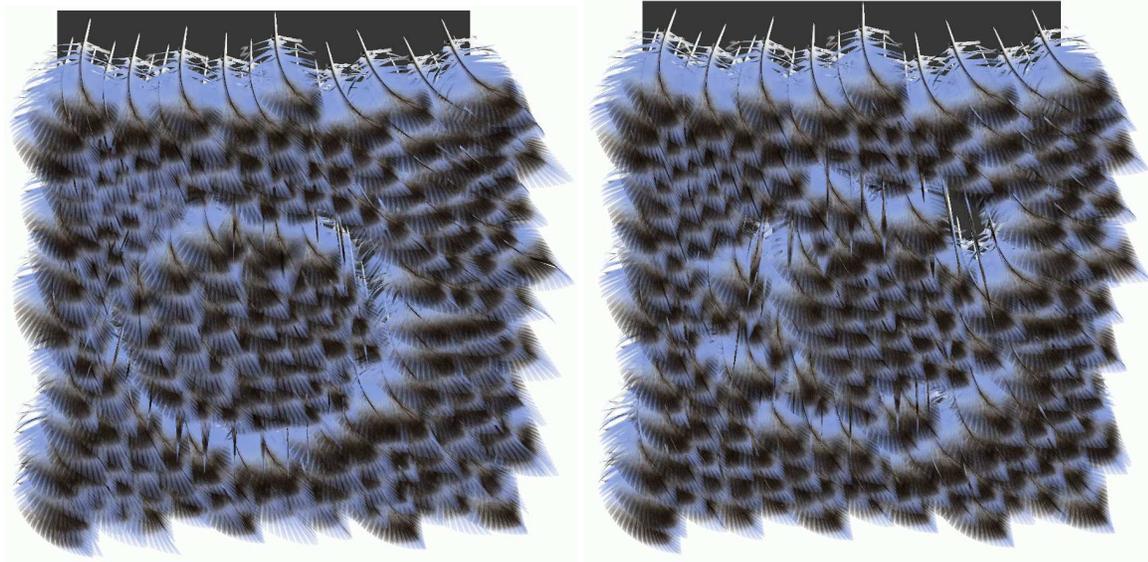


Figure 8.5: Two feather arrangements, both showing radial moulting sequence. **Left:** No randomness in feather chosen for moulting. **Right:** Randomness added to feather choice with a larger moult window size.

Figure 8.5 shows a snapshot of an example of a radial moulting sequence, where the wave of moult proceeds simultaneously in all directions. This is commonly found in the Capital and Femoral tracts. On the left there is no randomness added to the feather sequence. The feather chosen and its age are strictly a function of the distance from the follicle tract centre in the u,v parameter space of the feather interpolation, and thus, strictly sequential. On the right some randomness is added in the feather selected for development, however, development of each feather is consistent. This simulates the wave of moult without strict sequencing. Again note how in both cases, the moulting visibly alters the feather coat pattern.

To the authors knowledge, no similar previous computer graphics literature exists for comparison of this technique. Fur and hair regeneration have not been addressed in the computer graphics literature. There is some previous literature on the modelling and rendering of plant ecosystems [DHL⁺98], which addresses the simulation of ecosystems and the consequent death and regrowth of plants. They use the principle of self-thinning in combination with size and density for determining when plants die or where new plants to grow. For inter-specie ecosystems, rules determine which of two competing plants live based on a set of parameters identifying the species. Similarly, with feathers, rules determine loosely the sequence

of development. However, the location for development is fixed according to the follicle locations and the type of developing feather is also predetermined by a set of parameters.

8.3 Feather Rearrangement

The third and final element of the feather alteration component is the rearrangement of feathers. The feather rearrangement has two aspects: automatic and interactive control. The automatic aspect is responsible for adjusting the direction and orientation of the feathers during changes in the ages of feathers or the underlying skin and the interactive aspect is responsible for computing changes in feather direction and orientation due to external information, such as imposed feather angles or surface directions. Both of these aspects are handled quite seamlessly by our packing algorithm.

During feather coat morphogenesis feathers can be found in various stages of development. This means that a pin feather or partially opened feather has no curvature either into the surface or laterally. As a result, the feathers that are layered on top of this partially developed feather cannot pack any closer to the surface, than this feather in its straightened state. Hence, a feather coat with pin feathers may appear to be packed less tightly than a feather coat with fully developed feathers. If the feather is not fully developed it is assumed to be stiff enough that no barb elevation should occur, therefore only the straight shaft of the feather is considered and is packed in the same manner as detailed in Section 6.3.

During animation of the skeletal model and consequent skin deformations, the follicles alter their density and thus, the layering, direction and orientation of the associated feathers may also change. In this example the skin movement is generated using a very basic technique of vertex blending in an area of influence around the joints. Figure 8.6 shows four frames from an animation of a bending wing. In the top left (a) is the original wing pose. To the right (b) shows the wing slightly bent downward, and (c) and (d) show the wing bent upward at varying degrees. On the top is the skeletal movement with the resulting skin deformation. In the middle the follicles with their changes in density are shown and on the bottom is the resulting feather arrangement. The model was feathered using two tracts: one for coverts and the other for flight feathers, denoted as gray and white in the follicle diagram. At each frame the feathers are repacked. The consequences of this method are discussed in Section 9.1.3. Note how the changes in follicle density are reflected in the resulting feather arrangement and the feather orientation is maintained relative to the changes in bone direction.

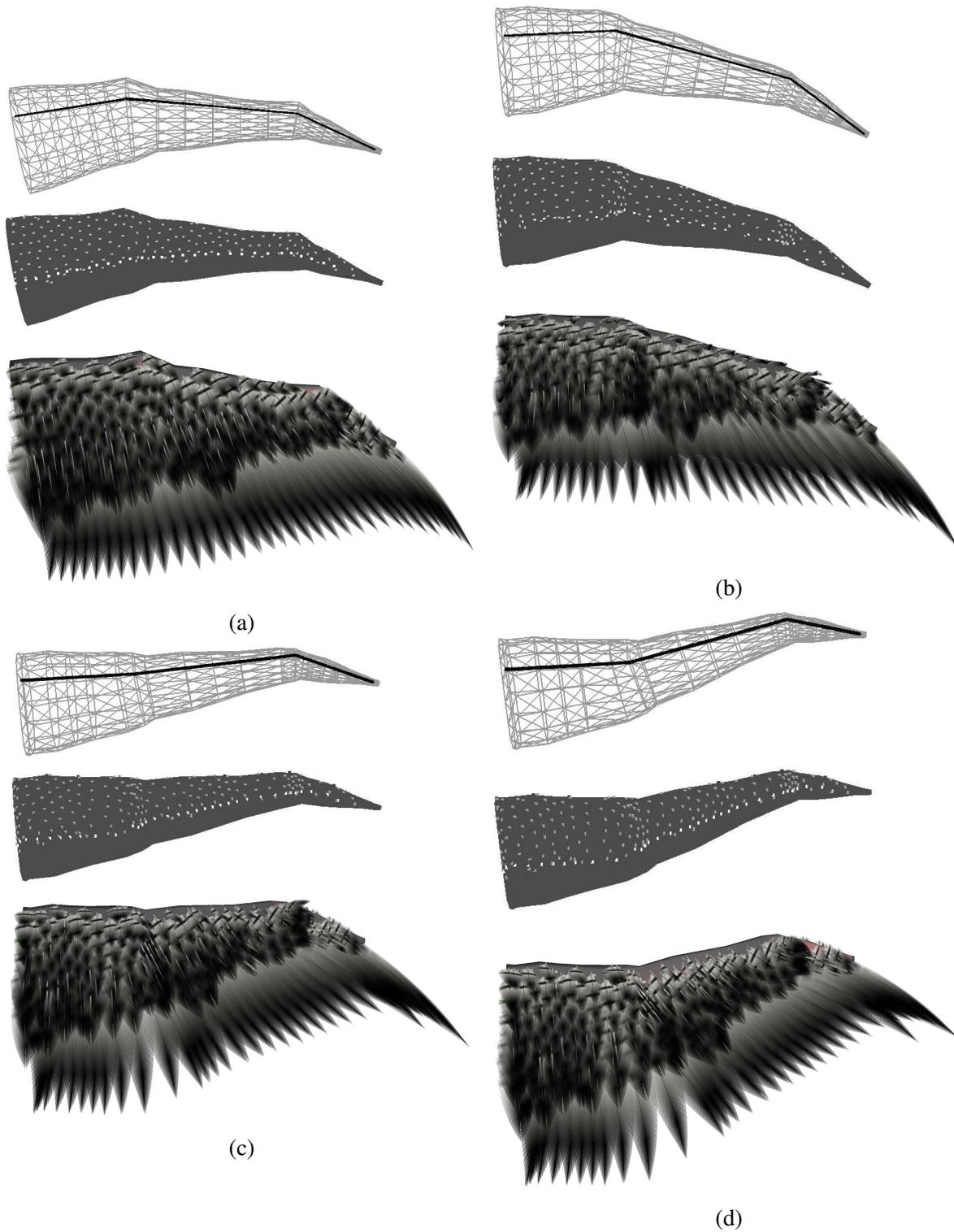


Figure 8.6: Bending of wing, movement of follicles and rearrangement of feathers. **(a)** Initial position. **(b)** Wing bent downward. **(c) & (d)** Wing bent upward various amounts. *Top:* Wireframe of model with skeleton (bones) in black. *Middle:* Follicles of two feather tracts: coverts in gray, flight feathers in white. *Bottom:* Feathered wing. Note how the density of the follicles has increased around the joints and how the feathers respond and are packed appropriately.

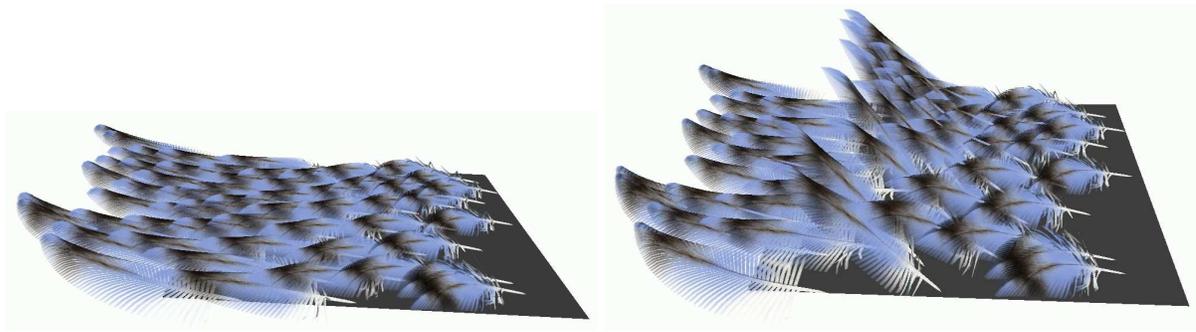


Figure 8.7: Using fixed feather angles to *style* the feather coat. **Left:** Original feather arrangement. **Right:** Feather arrangement with a row of constant fixed feather angles. Note how the other feathers have been packed appropriately.

Interactive control over the feathers is fairly limited since the feathers are fixed at one end in their follicle. However, a means of controlling or “styling” even the limited movement of the feathers can be useful in achieving specific feather arrangements or resulting appearances. In a similar context to hair styling, it is too cumbersome to style each and every feather. Common tools used in hair styling include parameter sets (i.e. clumps or strands) [RCT91, DTKT93], fluid flow streamlines field [HMT00] and characteristic hairs [WS92, Bru98, KH00, IKVZ00]. We have chosen to allow for interactive feather arrangement through two means: *fixed angles* and modification of surface direction vectors. The first interaction method allows for the definition of a *fixed angle* for a feather or set of feathers. In our implementation this is done by the user choosing feathers and defining a fixed angle, but a variety of tools which act as combs could also be used to lift the feathers to their desired fixed angle. The second interaction method allows the user to modify the surface direction vectors to obtain the final feather coverage arrangement. The surface direction vectors are modified directly on the surface of the object. In both of these cases, the feathers are then packed as before to form the resulting feather arrangement.

Figure 8.7 shows an example of how creating fixed feather angles can be used to create different styles. In this example the unstyled feather arrangement is found on the left and the styled arrangement on the right. In the feather coat on the right a straight tool was used to lift a row of feathers to a constant fixed angle. Then the other feathers were packed against these fixed-angle feathers. The interactive tools that have been developed are very limited. However, any type of interaction can be used to set feather angles and studying the usability of various tools is outside the scope of this thesis. Tools may include using any shape object as a attractor for lifting the feathers, including spheres, arcs, jagged combs etc. Regardless of the tool

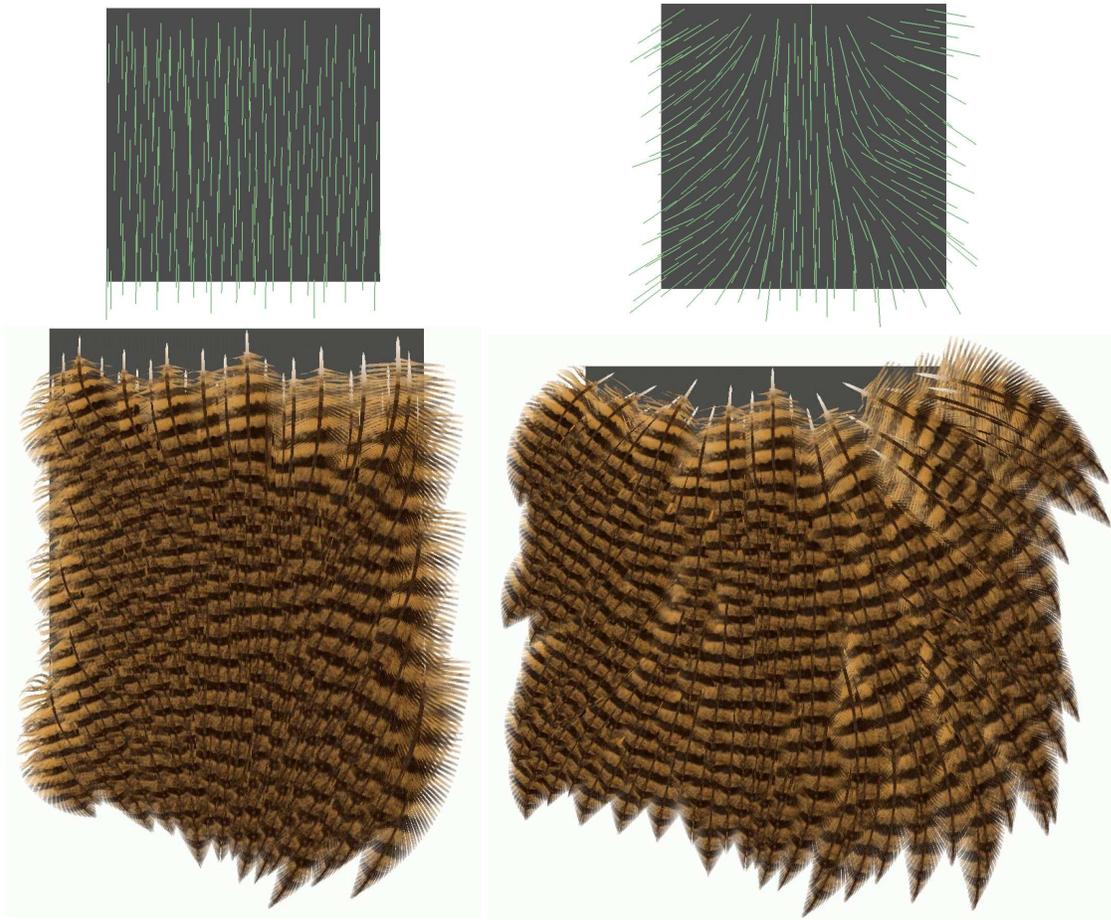


Figure 8.8: The use of styling vectors can be used to alter feather surface directions to cover adjacent apteria. **Left:** Original configuration. **Right:** After coat design, alteration of surface direction vectors to cover adjacent apteria.

used, or the set or amount of fixed-angle feathers specified, the rest of the feather coat will rearrange itself appropriately through packing.

Figure 8.8 shows how the alteration of the surface direction vectors can be used not only to alter the feather coat pattern, but also to change the feather arrangement to cover adjacent apteria. On the left the feather coat is first designed with only a single biasing vector making all surface direction vectors point straight downward. However, after designing the coat, there may be a need to cover regions adjacent to this tract (possibly apteria) and this can be done by modifying the surface direction vectors to the configuration shown on the right, without re-modelling the entire coat. If the change in direction vectors is extreme, an automatic repacking of the final configuration may be needed.

8.4 Summary

This chapter presented the alteration component of our feather coat morphogenesis model. This component contained three elements: *feather growth*, *feather replacement* and *feather rearrangement*. Together these elements define both how the feather coat responds to skin movements and how it changes over time - feather coat morphogenesis. Although this component and chapter do not add technically to the methodology, it exemplifies the methodologies robustness by showing that both the feather structure parameterization and feather packing algorithm can respond appropriately in changing environments.



Results and Analysis

This chapter discusses the contribution and relevance of our feather coat morphogenesis methodology described in Chapters 5 – 8 in the context of the thesis objective. This chapter begins with a further analysis of the results from each of the three components: *feather structure*, *feather arrangement* and *feather alteration* as well as discusses the capabilities and bounds of each component. Further analysis of the method, including details of integration is presented, concluding with the constraints and limitations of our method.

Recall that our thesis is that *modelling feather coat morphogenesis using a multi-resolution methodology that is feather parameter specific will best facilitate the creation of “realistic looking” feather coat for the purposes of animating the follicle surface or “skin”*. With this in mind it is important to show that our methodology firstly, is capable of modelling feathers and feather coats in a changing environment and secondly that the multi-resolution, parameter-specific approach we have chosen is best for this task. In the first part of this chapter, a discussion of the bounds and analysis of the results of each component in comparison with real feathers and other related work will be given. The second part of this chapter, discusses integration issues and compares the final results with both real feathers and other related work.

9.1 Components of the Model

Due to the nature of the work, it is difficult to analyze the final results. The product of the methodology is a realistic-looking model or image of a feather coat at a certain stage of development. Thus, an analysis of the results would include either image or model comparison with images or other models of real feathers.

Measures do exist for comparing synthesized and real or acquired images as well as synthesized and acquired textures. However, these measures are not adequate in the context of this model. A direct image comparison is not adequate, because the rendering method is not exact (i.e. does not take into account all lighting and optical effects), the individual feather patterns are not synthesized and no biologically exact method exists for creating these. As well, this work is focused on the structure of the feather, which can be obscured by lighting and textures in resulting images. Since there are no measures for determining the “likeness” of a feather coat, a visual comparison of the results must be sufficient. Though a visual comparison of synthesized feathers coats with real feather coats is necessary, it is not sufficient for similar reasons (lack of optical effects, and biologically accurate pattern generation). Thus, visual comparison of the individual component results as well as a specification of the component capabilities is made in this section. The two aspects of our methodology we focus on for assessment are:

- evaluating the capabilities, bounds, flexibility, performance and limitations of the methodology and comparing these to other related work
- comparison of the results both with real feathers and feather coats and with results of other models of feathers and feather coats

9.1.1 Feather Structure Component

As stated in Chapters 4 and 5, the parameterization for defining the feather structure must be capable of creating a wide variety of feather types and ages. To facilitate this, a biologically based parameterization was chosen (see Chapter 5), meaning that the parameters used in the model are based on the substructures of the biological description or taxonomy of the feathers as outlined in Chapter 2.

In Chapter 5 it was shown that a variety of feathers could be created and that the parameterization facilitated blending between different types. Figure 9.1 shows some examples of both contour and non-contour feathers as compared with flatbed scans of real feathers and shows feathers from the continuum from down to semiplume to plume (contour) commonly observed in nature. The top left (a & b) represent feathers from the continuum from down to semiplume with real feathers shown on the left and synthesized feathers shown on the right. Continuing from semiplume in (a) & (b) to plume in (e) & (f) with real feathers on the left (a,c,e) and synthesized ones on the right (b,d,f). The three key feathers used for this interpolation are the down (b top), semi-plume (b bottom) and plume (f bottom) feathers. The latter two key feathers have three sets of key barbs: one near the base, and two near the tip. The texture used for the feathers was

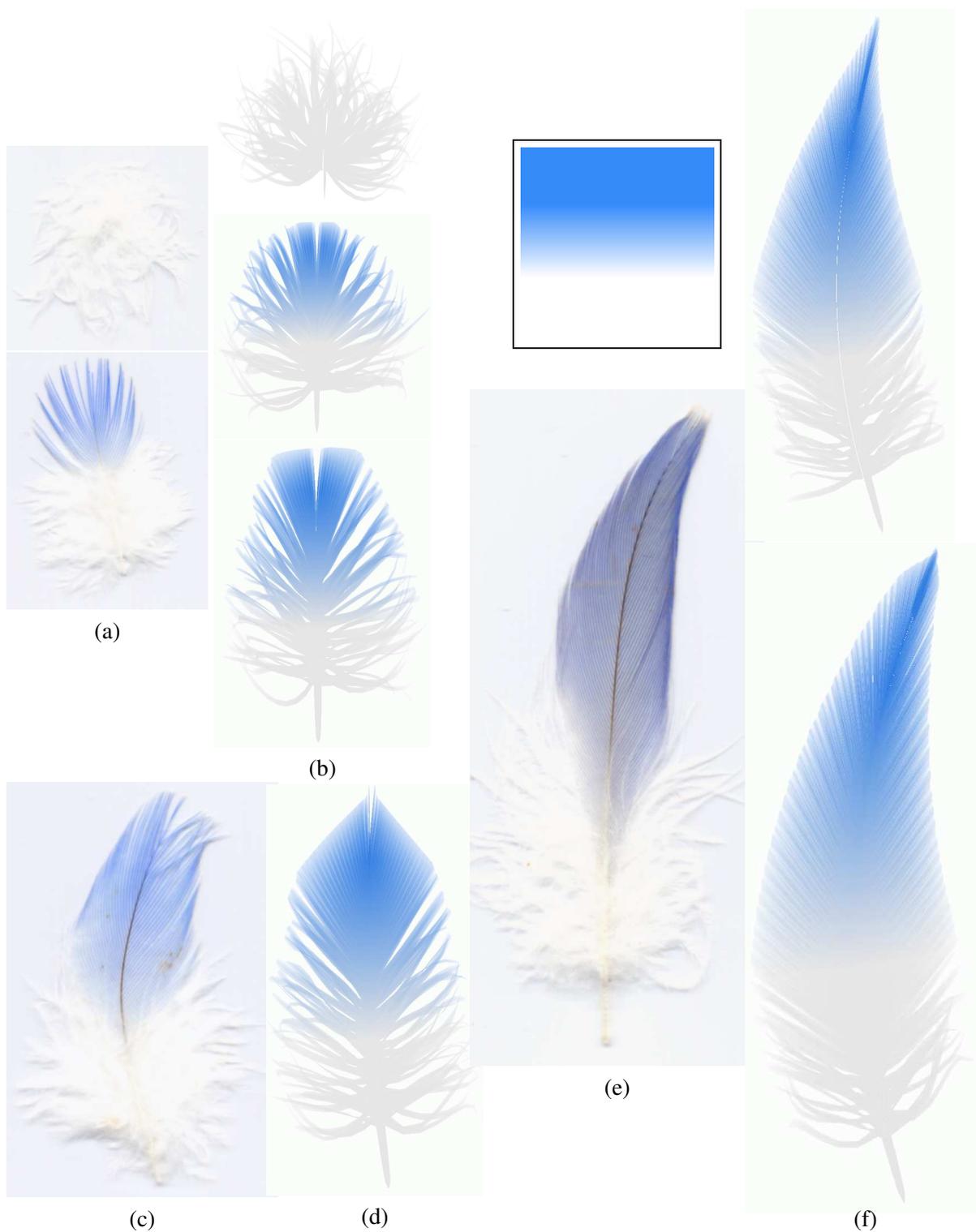


Figure 9.1: Down to Semiplume to Contour continuum. **(a) & (b)** Down to Semiplume continuum. **(a)** Real Budgie feathers. **(b)** Three synthesized feathers from key feathers on top and bottom. **(a)-(f)** Semiplume to Contour continuum. **(c) & (e)** Real budgie feathers. **(d) & (f)** Three synthesized feathers from key feathers (b) bottom and (f) bottom. The synthesized texture used is show (e) top.

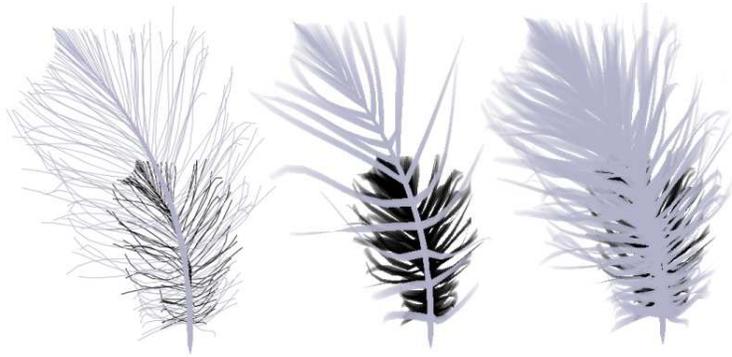


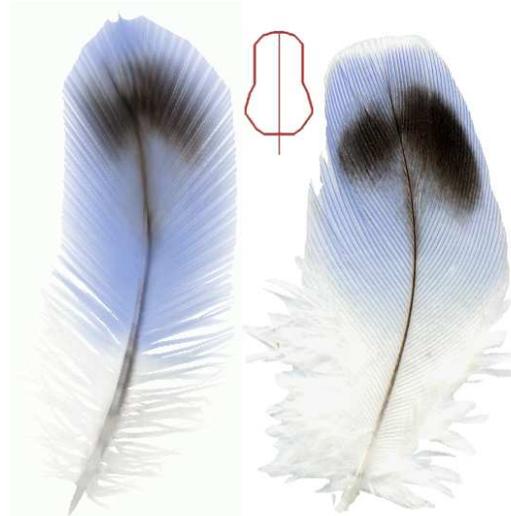
Figure 9.2: Semiplume (black) with afterfeather (blue/gray) **Left:** Poly-line version. **Centre and Right:** Triangle strips version. The centre feather has less barbs for illustration.



Figure 9.3: The synthesized budgie feather from Figure 9.1(b) bottom with an afterfeather added.



(a)



(b)

Figure 9.4: (a) Hawk feather *Left:* Synthesized. *Right:* Flatbed scan real feather. Synthesized feather uses barb length from the scan. (b) Budgie feather *Left:* Synthesized. *Right:* Flatbed scan of real feather. Synthesized feather uses barb length from shape data shown at centre. Both feather textures are acquired from the flatbed scans.

synthesized and is shown at the top of (e). Each feather can be easily made in under 60 seconds, once the texture is synthesized.

As mentioned in Section 5.2 the afterfeather is modelled as a secondary feather attached to the main feather with the same parameter set. Figure 9.2 shows a generated semiplume 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 afterfeather in black for illustration. Though the addition of the afterfeather is subtle, it does make a difference as shown in Figure 9.3. Figure 9.3 shows the synthesized Budgie feather from Figure 9.1(b) bottom with an afterfeather added. Compare this with the real feather in Figure 9.1(a) bottom that also has an afterfeather. All feathers in Figure 9.1(a,c,e) have afterfeathers of varying degree.

Figure 9.4 shows the comparison of our generated feathers with flatbed scans of the real feathers. In each pair the synthesized feather is shown on the left and the flatbed scan is shown on the right. The flatbed scan can not only be used for validations, but also as a source of data. The texture, the shaft curvature and silhouette or shape data can be extracted from the scan. In our implementation the scanned image is displayed for the user to interactively specify the location of the rachis. A basic image segmentation technique [GW92] was then used to segment the feather image from the background and compute both the shaft curvature and the silhouette or shape data which in turn specifies the barb length. The feather on the left uses barb length and rachis curvature from the scanned feather. The feather on the right uses barb length computed from the silhouette shown in the centre. Both feathers use textures from the scans. The only adjustment made to the texture was to the rachis placement and to re-sample the texture.

Figures 9.5 and 9.6 show more feathers. Figure 9.5 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*. A flatbed scan of an actual peacock feather is shown on the right. Figure 9.6 shows a synthesized pheasant feather on the right and a flatbed scan of a pheasant feather on the left. Both the shape data and the texture are taken from the scan. Only two sets of key barbs are specified here, one at the tip and one slightly below the tip to flatten the patterned pennaceous portion.

In addition to modelling many types of feathers it is also important to model different ages of feathers. Figure 9.7 shows the rendered growth sequence similar to Figure 8.2. Figure 9.8 shows another example of growth with the same growth data. Note how during the pin feather state the rachis stays straight due to it's stiffness and only once the feather is completely open, does it obtain its curvature.

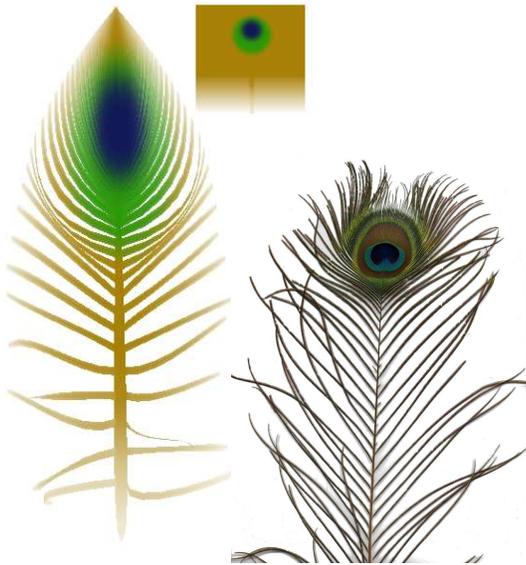


Figure 9.5: Peacock feathers. **Left:** Synthesized from texture on top **Right:** Flatbed scan of a real peacock feather.



Figure 9.6: Pheasant feathers. **Left:** Synthesized feather using scanned data. **Right:** Flatbed scan of actual pheasant feather.

Finally, as previously detailed in Chapters 5 and 7, our methodology involves the ability to create models at multiple resolutions. Figure 9.9 shows a feather at full geometry with and without splits in the vane on the left and then at half and a quarter the geometry to the right. The use of the alpha texture map and multi-texturing as outlined in Chapter 7 was used to prevent the loss of detail in lower resolution models. The bottom of Figure 9.9 shows a zoom of the feather vane at full (left), half (centre) and quarter (right) geometry.

Structural Component Analysis and Scope

As detailed above, the structural component is capable of creating a wide range of feather types and ages of feathers as well as enables the use of multiple resolutions for efficiency. There are however, bounds to the types of structures this model is capable of creating, mainly due to the use of linear interpolation.

Linear interpolation is the aspect of the methodology that automates generating the feather coat, reducing the amount of manual modelling and thus, is also one area that is less controllable. Linear interpolation is used in three areas of the structural parameterization: barb spacing sp_b, sp_t , barb length l and vane angle $v_{ba}^l, v_{ta}^l, v_{ba}^r, v_{ta}^r$. This means that non-linear interpolated results and discontinuities are more difficult

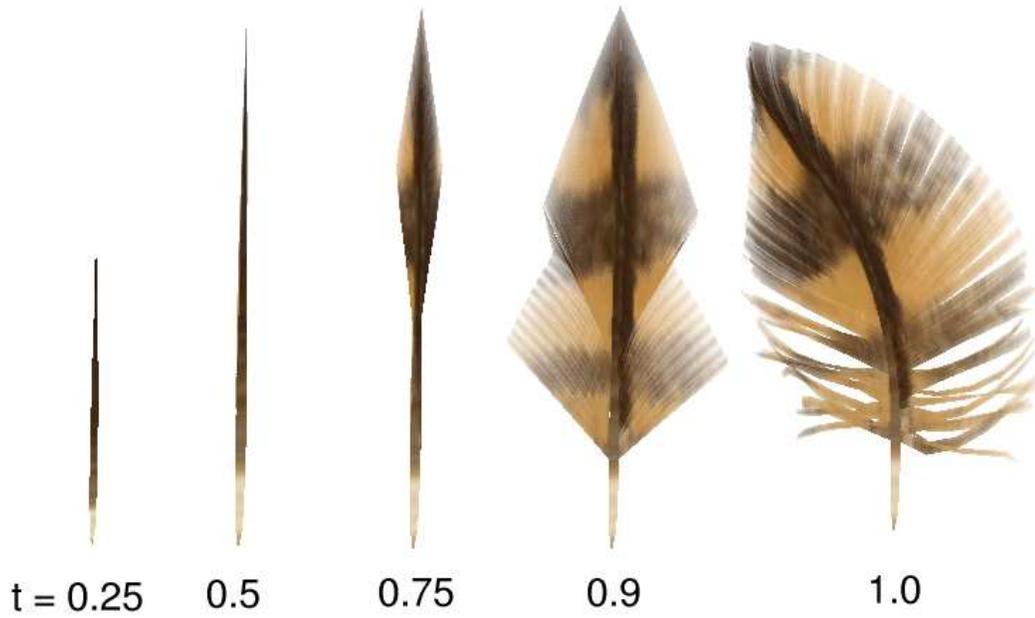


Figure 9.7: Growth sequence of feather similar to Figure 8.2.

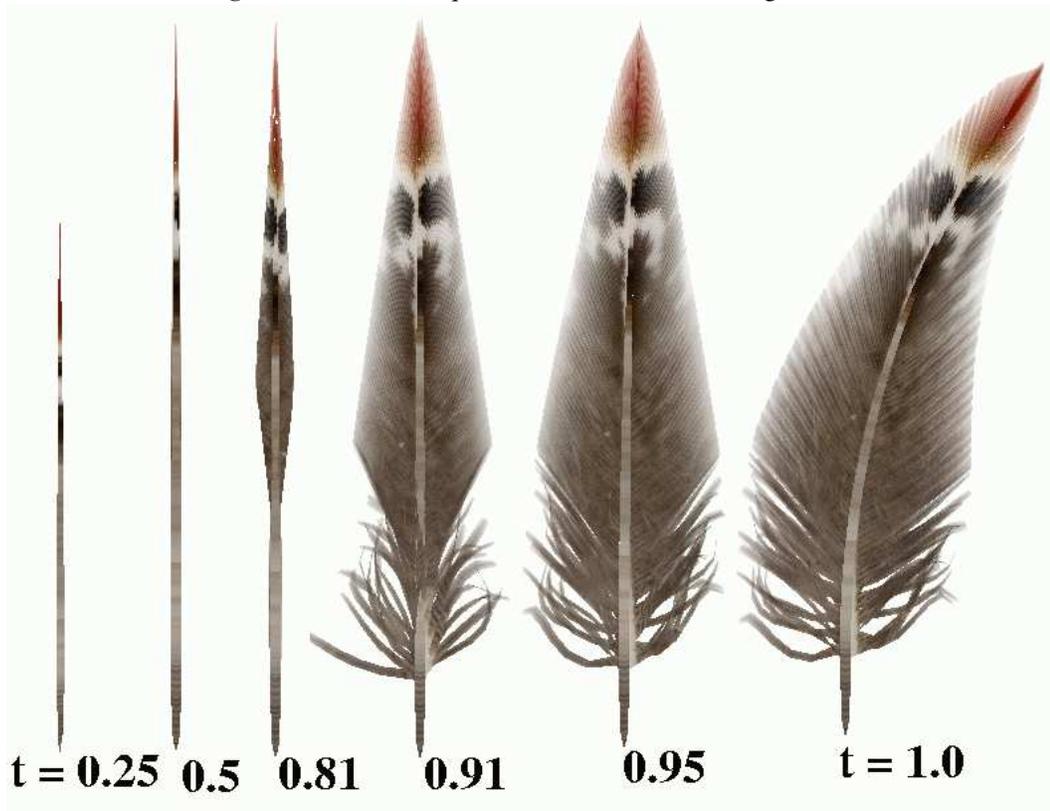


Figure 9.8: Growth sequence using same growth data as in Figure 9.7, but different feather model.

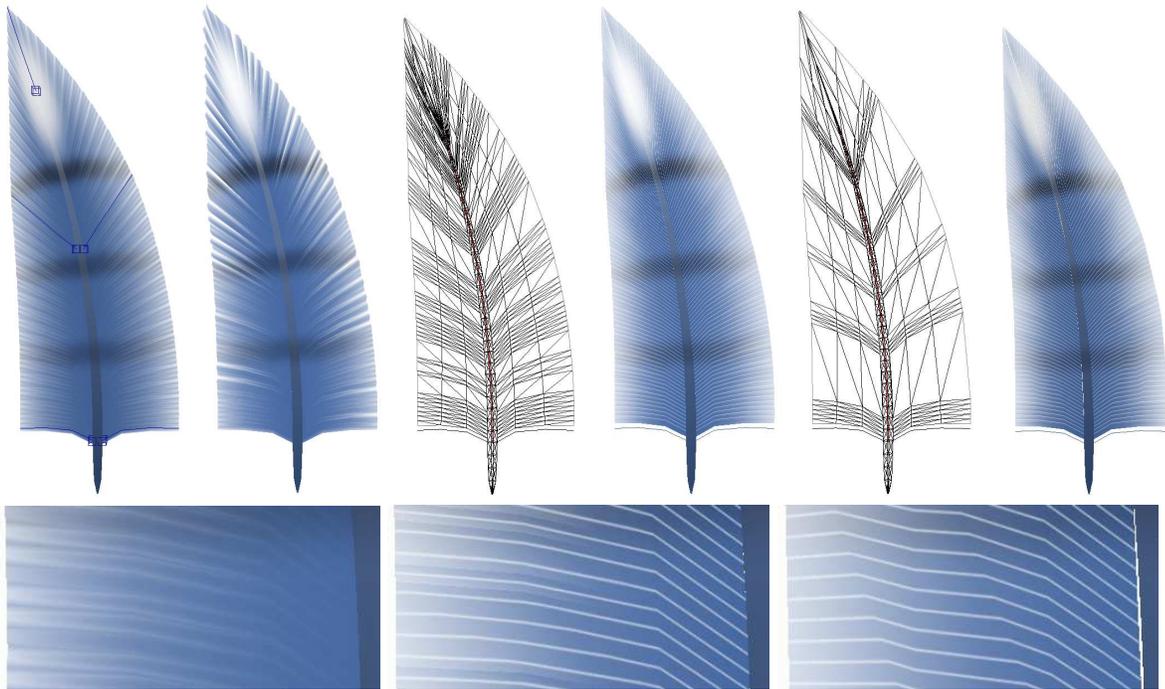


Figure 9.9: Left to Right: Feather with no splits and full geometry (key barbs shown in dark blue). Feather with random vane splits at full geometry (highest LOD – approx. 1660 triangles). Feather with half geometry. Feather with one-quarter geometry. **Bottom (l-r):** Magnification of feather vane, with full, half and quarter geometry.

to obtain. To overcome this restraint the interpolation function in the model could easily be replaced by some other interpolation function. In addition, if discontinuities in the barb length are desired either this can be accomplished with adjacent key barbs as described in Section 5.2.2 or with the use of optional silhouette data. Achieving discontinuities in barb spacing or angle are not possible without adding another parameter to the model. However, again this does not occur in real contour feathers and is very atypical even of other types of real feathers, and thus is not an issue for our model. Linear interpolation is also used to compute the control vertices for the barbs, but once again discontinuities in the barb shape, can be achieved using the adjacent key barb technique from Section 5.2.2. However, if many discontinuities in barb shape are desired, this can become quite tedious. Fortunately, this is atypical of natural feathers and thus, is not an issue given the scope of our problem.

The use of Bézier curves is another core aspect of our model. Bézier curves are used for the shape of the key barbs (and hence barbs) as well as for specifying the curvature of the rachis. Since virtually an unlimited number of control vertices can be used, the degree of the curve is unrestricted. Thus, highly complex curves can be constructed if needed. The Bézier formulation was chosen because of its intuitiveness

of the control vertices to the curves. A more general B-Spline could also have been used instead of Bézier curves, though they allow more local control than Bézier curves (especially higher order), they tend to be less intuitive for interactive use specific to feather modelling because the shape of the shaft and rachis are more easily specified as global curves, rather than a piecewise collection of local curves. For general feather curvature shapes, a Bézier curve is more than adequate, but there are some special cases, where more local control over the curve may be useful.

The use of any curve presents problems when defining length. A stylistic choice must be made whether to maintain length as a Euclidean distance between the endpoints of the curve or as the arc-length along the curve. This choice was needed in computing the length of the curve (barb) from the shape or silhouette data. If arc-length is used then it is assumed that the shape or silhouette data specified represents a flattened view of the feather, and thus as the internal curvatures of the feather are edited, the arc-length must be maintained. This means, not only will the resulting unflattened silhouette shape not match the flattened shape, but the desired curve at the desired scale may be impossible to achieve given the arc-length restriction and thus, the curve must be scaled. If Euclidean length between endpoints is used it is assumed that the shape data represents the unflattened silhouette of the feather, and thus as the curves are edited the arc-length or physical length of the barbs may shrink or grow. While using arc-length is a more physically plausible scenario, it highly restricts the design of the feather. Without an accurate initial arc-length specification there may be limitations on the scale or degree of curve created. As well, the shape or silhouette of the feather is usually described independently from the barb shape and thus using Euclidean length is more intuitive. As such, Euclidean distance was used in computing the barb length during the design of the feather, but during packing arc-length was used to simulate a plausibly correct bending of the feather.

In comparison to other individual feather models, our model is much more controllable and flexible than both the models of Dai et al. [DSC95] and Chen et al. [CXGS02]. Both of these previous works generate the barbs by growing or propagating a segment of the barb or rachis at a time, with the direction of the next segment defined by either a set of angles and polynomials (Dai et al.) or a parametric curve (Chen et al.). While the functional set of angles proposed by Dai et al. interpolate nicely, they are less flexible and rather difficult to control. The only barb curvatures that can be constructed are those defined by a quadratic polynomial. Our method is similar to Chen et al. in their use of a parametric curve to define the barb curvature, however they only allow the definition of a single left barb curve and single right barb curve. Thus, their model cannot construct feathers with eye-spots such as peacock feathers and many other feather

curvatures. Also with their model there is no variation in barb angle or barb spacing/density within a single feather. So again, any feathers with asymmetric vanes cannot be constructed and the shape of the feather is completely defined by the user supplied outline curve.

Comparing our feather model with what is known about the model from Sony Pictures Image-works [EAB⁺02], our model is both more flexible and simplistic. They state [EAB⁺02] they have hundreds of parameters to control their NURB patch based feathers. While it is assumed that many of these parameters control the lighting effects, this is still an order of magnitude more parameters than are used in our model which is approximately 15, plus key barbs and shape data for individual feathers in addition to the biasing vectors, parameterization and surface vectors for creating the coat. Also, using NURB patches for the surface of the feather makes having plumulaceous portions or down feathers impossible.

9.1.2 Feather Arrangement

The second component of our model is the feather arrangement component described in Chapter 6. The key attributes of this component are again flexibility without loss of control, as well as automation of the process. Placement and orientation of each individual feather is too tedious. As well, the use of multiple resolutions is critical especially when creating dense feather coats. This second component contained two main elements: follicle distribution and feather orientation. Three steps were used to construct the feather arrangement: follicle distribution, feather generation and feather direction and orientation (i.e. packing).

The first step, follicle distribution, uses a user-defined density value to randomly distribute follicles over a subset of polygons, representing a feather tract. The follicles are then iteratively relaxed to obtain a regular hexagonal or “stretched hexagonal” distribution using a set of user-defined biasing vectors. Adding the ability to place biasing vectors, adds flexibility to create non-linear distributions as well as ability to more closely approximate actual follicle distributions. Figure 9.10 show an artists illustration of the Dorsal tract of a Single Comb White Leghorn Chicken [LS72b, p.425] on the left and our follicle distribution on a set of polygons representing the Dorsal tract on the right. Note the similarity in the hexagonal arrangement of follicles. The image on the right also illustrates the “rows of follicles” similar to those on the left.

The need to approximate this “stretched hexagonal distribution” is based on the hypothesis that the density variation actually affects the resulting feather coat pattern and thus, the follicle distribution is important. Figure 9.11 exemplifies this importance. This figure shows three example distributions using the same type of feathers, the same individual feather patterns, and the same surface directions for the feathers.

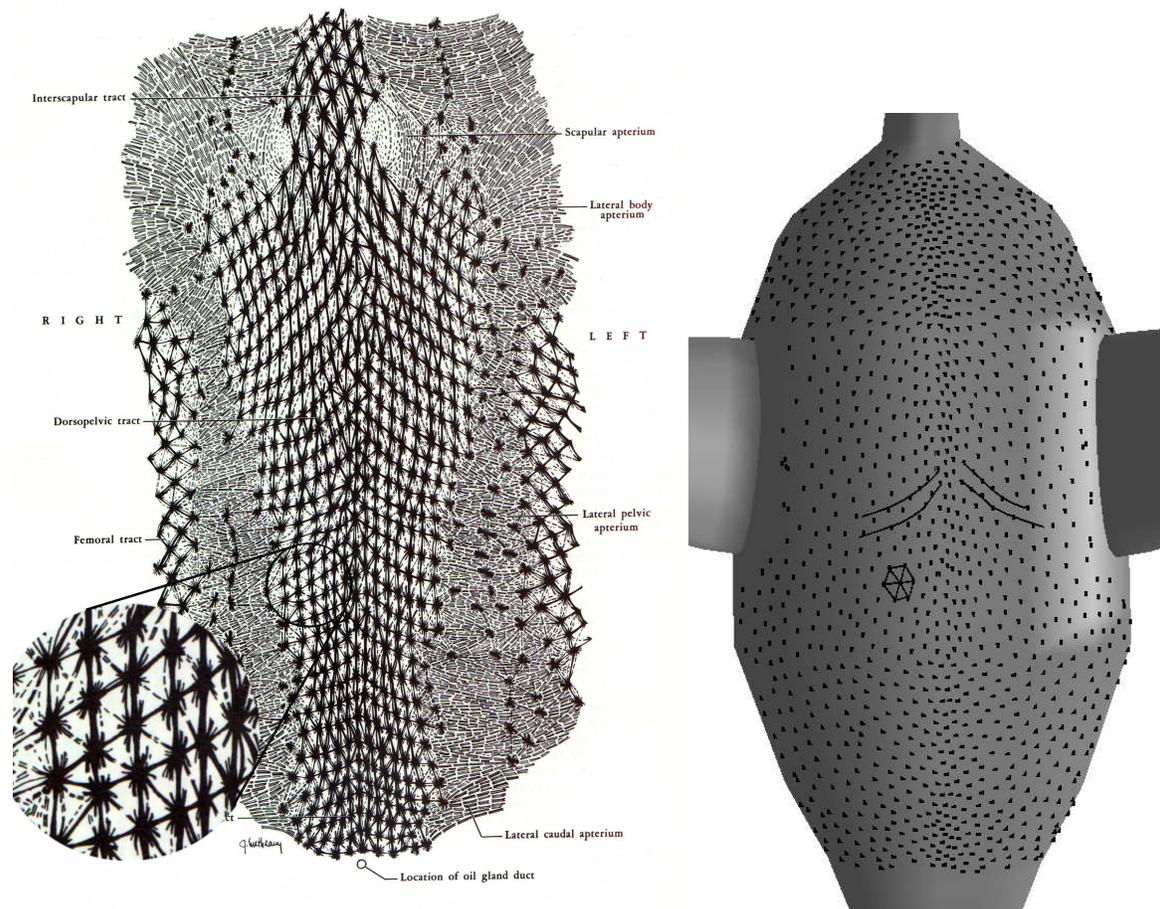


Figure 9.10: Simulating the hexagonal follicle distribution. **Left:** An illustration from Lucas and Stettenheim [LS72b, p.425] with an inset that shows the roughly hexagonal distribution of the follicles of the dorsal tract of a Single Comb White Leghorn Chicken. The connecting lines between the follicles represent follicular muscles. **Right:** Follicles generated using our technique. Note the similarities in the hexagonal distribution as well as the alignment of follicles into rows.

The only difference is the follicle distribution. The image on the far right has 60 follicles, and the other two have 57. The follicles near the edges have been slightly perturbed to define the convex hull for the feather interpolation. However, if we focus on the centre of each square where the distributions are most different and yet the type of feathers and feather directions are the same, there is a noticeable difference in feather coat pattern caused solely by the follicle distribution. The follicle distributions are shown at the top and the two key feathers positioned at the corners are shown on the bottom left and right. The rest of the feathers on the bottom show snapshots of the feather interpolation.

The processing time required to compute the final follicle distribution is dependent on the desired

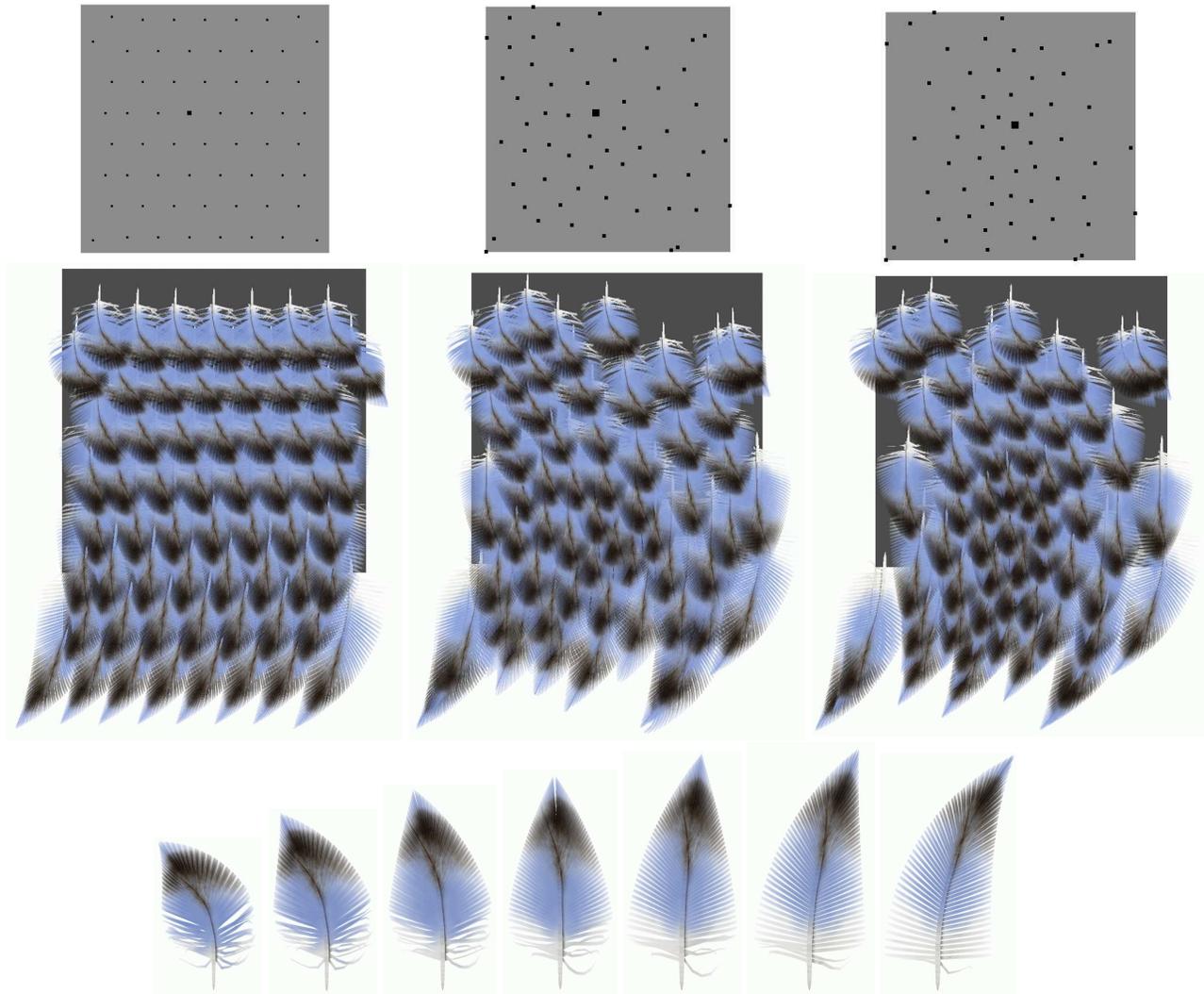


Figure 9.11: How differences in follicle distribution affect the overall feather coat pattern. **Left:** Follicles aligned in “grid-like” pattern. **Middle:** Follicles with regular hexagonal distribution. **Right:** Follicles with stretched hexagonal distribution. **Top:** Follicle distribution. **Bottom:** Two feather types used as key feathers shown on left and right. The rest of the feathers at the bottom are interpolated.

final distribution due to the iterative nature of the algorithm. Each iteration of the distribution algorithm traverses each follicle once, and thus is also dependent on the number of follicles. At each follicle the follicles are re-triangulated (Delaunay triangulation) to compute the new Voronoi regions and then each follicle is moved to the centre of its Voronoi region. To compute the Delaunay triangulation each edge is traversed and for each edge an undetermined number of edges may be affected. Thus, the Delaunay triangulation is $\mathcal{O}(m^2)$ where m is the number of edges. If n is the number of follicles an iteration to relax the follicles is $\mathcal{O}((n + 2)^2 + n)$. However, since the follicle distribution is produced offline, independent of the feather arrangement, this does not restrict the interactive modelling of the coat. Each collection of feathers in Figure 9.11 has approximately 60 feathers. The feather geometry is computed in real time with each scene containing approximately 100,000 polygons using full feather geometry. The packing of the feathers took approximately 2 seconds. With alpha blending this scene is rendered at approximately 3-5 fps on a PIII 800 GeForce 2 PC.

The second step of the feather arrangement component involved parameterizing the follicles and placing key feathers so that the in-between feathers could be interpolated as described in Chapter 5. This involved projecting a screen space grid/parameterization onto each tract, so that each follicle had a u,v interpolant coordinate. Once this is defined, it is fixed, since this ultimately defines the type of feather associated with this follicle. We will discuss the implications of this parameterization below and in Section 9.1.1 we showed the quality of the interpolated feathers.

The third step was to define the actual orientation and direction of the feathers. This involved the user defining surface direction vectors, which ultimately defined a coordinate system for each follicle, within which a packing algorithm is used to pack the feathers in a shingle-like fashion. The ordering of these layered feathers is defined by the same parameterization used for the feather interpolation. Alternately, a secondary parameterization could be imposed to define the ordering of the feathers, in a similar fashion as the follicle parameterization in step two.

Once an ordering and coordinate frame are defined at each follicle location, the feathers can then be packed. The objective in packing the feathers was to prevent the feathers from interpenetrating the surface and each other, since this generally does not occur in nature. The method for packing used the information that the feathers were constructed from interpolations of Bézier curves and that not only did the key barbs form a bounding volume for the feather, but the control vertices of the key barbs defined an absolute bounding region. Information about the inherent stiffness of the feather was also built into the

9.1. COMPONENTS OF THE MODEL

algorithm. This allowed the stiff shafts of the feathers to be packed without deformation and the more pliable barbs deform slightly to prevent interpenetration. Thus, each feather was traversed in bottom to top order (painterly) first maximally rotating the shaft in the follicle coordinate frame using a subset of the surrounding feather control vertices and then optionally elevating the barbs.

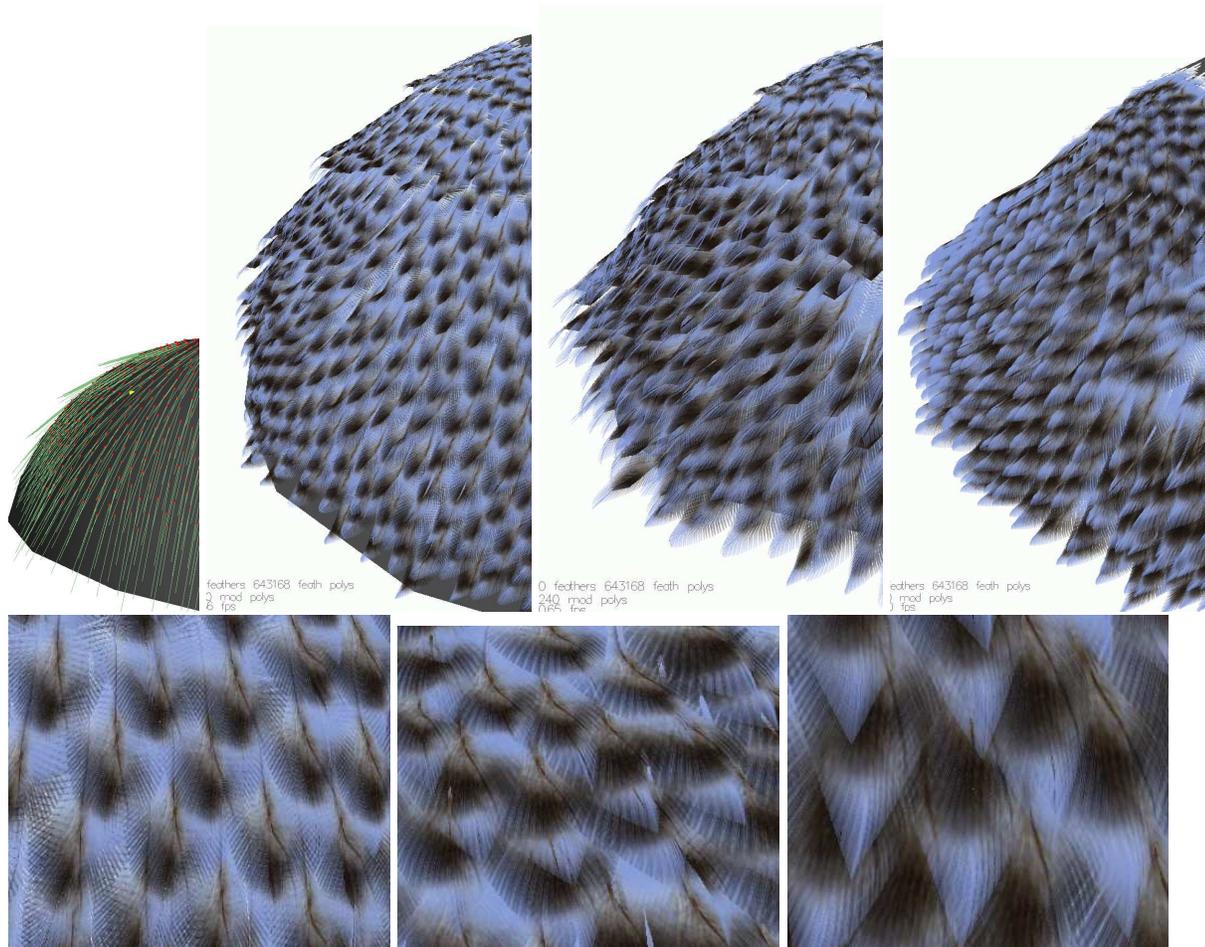


Figure 9.12: Feathered sphere comparing packing results with varying feather size to follicle density and feather size to surface feature size ratio. All three feathered spheres have the same feathers, surface direction vectors and approximate number of follicles. They vary in feather scale - smallest on the left to largest on the right. **Bottom:** Magnifications of various regions of the three spheres. Note size variation of feathers.

In Chapter 6, Figure 6.12 we showed an image of feathers, packed and unpacked. Not only did the unpacked feathers interpenetrate the model's surface, but they also interpenetrated each other. In the packed version, all of these interpenetrations were eliminated. In the case of the a planar surface, the packing is nice and tight as shown in the side view. With curved surfaces, the tightness is dependent on the density of the follicles, the size of the feathers in comparison to this density and the curvature and tessellation of the

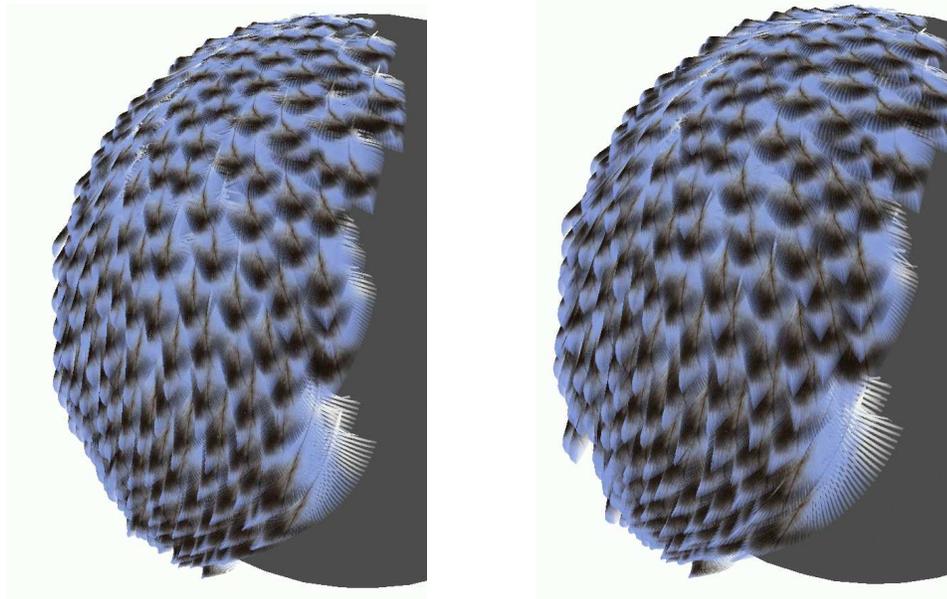


Figure 9.13: High resolution sphere (1600 polygons v.s. Figure 9.12 with 250). **Left:** Small feathers **Right:** Larger feathers. Note the smooth uniform contour and how the larger feathers are starting to bulk at the bottom.



Figure 9.14: Feathered sphere with multi-texturing. **Left:** Full geometry **Right:** Half-geometry. Magnified portions of each feather arrangement shown to their right.



Figure 9.15: Comparison between a feather pattern on a real budgie (left) and a synthesized feather pattern (centre) using the feather on the right, a regular hexagonal distribution of follicles and our feather coat methodology.

surface itself. Figure 9.12 shows a portion of a sphere with a set of follicles and the corresponding surface direction vectors and the same sphere feathered with the same feathers, simply at three different scales. The direction vectors are purposely drawn long to demonstrate the low tessellation of the underlying surface. The rightmost feathered sphere has feathers twice the length of the leftmost. The sphere in the middle is midway in length. Note how smaller feathers follow the surface more tightly. Hence there is a disjoint contour similar to the direction vectors - due to the low surface tessellation. The larger feathers hide these surface details, but also create more bulk, especially near the base of the sphere where the feathers are the largest. Below these images are magnifications of each scene. In Figure 9.13 we show the same type of feathers with similar density on a higher tessellated sphere (1600 polygons v.s. 250 polygons). On the left the feathers are smaller and to the right are larger feathers. Note how both sets of feathers now follow the smoother surface and similar bulking behaviour is starting at the base of larger feather sphere.

In Figure 9.12 there are 426 feathers (not all shown) comprising approximately 643,000 polygons at full geometry. With normal alpha blending (not multi-texturing) we are able to obtain approx. 1-2 fps on PIII 800 with a GeForce 3 graphics card. With the image on the left (smaller feathers) packing required approximately 20 seconds. The image on the right has larger feathers and required approximately 151 seconds for packing. The image in the middle had feathers whose length was mid-way between the left and right images. Packing for this middle image required approximately 66 seconds. The feather structures used are the same as those in Figure 9.11. The increase in packing time is proportional to the increase in the surface area of the feather. With an increase in feather surface area, the feather covers more underlying feathers, each with more candidate control vertices that must be considered in the packing process. This explains the non-linear increase in computation time.

Figure 9.14 shows the use of multi-texturing on the same feathered sphere at multiple levels of detail. On the left is the sphere at full geometric detail (approx. 643,000 polygons), rendered at approximately 0.9 fps. on a PIII 800 with a GeForce 3 graphics card. On the right is an image with approx 45% of the polygons (299,000) and is rendered at approximately 1.5 fps on the same machine. To the right of each image is a magnification for analysis of the multi-texturing. As is shown much of the detail is maintained even with less than half the geometric complexity.

Figure 9.15 compares the pattern of a real Budgie feather coat with one synthesized using our feather coat methodology. On the left is a image of a portion of the left wing of a budgie. In the centre is our

synthesized feather arrangement and on the right is the feather used to construct the arrangement. A dense hexagonal distribution of follicles was used. Note the similarities in feather pattern.

Arrangement Component Analysis and Scope

The main restriction of this component is the follicle parameterization. Projecting a grid from screen space to object space, can possibly create distortions in the parameterization, depending on the topology of the surface. As well, for some surface regions, parameterizing in this way is impossible without subdivision of the region. The projection must be unique and close to planar to minimize distortion in the parameter space. Devising the parameterization in screen space does however, allows the user to construct any uniform or non-uniform parameterization space desired. For instance if non-uniform interpolation of the feathers was desired in certain regions a local adjustment to the grid density would suffice to construct such non-linear interpolations. This manual grid adjustment, can also be used to absorb the projection distortions, however, would require some skill and time to construct

Of course if the surface already contains a parameterization and the parameterization is appropriate for the desired feather interpolation, this step can be avoided. As well, there is current research on automatically devising non-distorted parameterizations for polygonal meshes, especially for genus-0 [GGS03] meshes. Thus, a more sophisticated technique could easily be used in place of our rather simplistic method. All other user-defined data, such as biasing vectors, surface direction vectors etc. are manipulated directly on the surface, and thus are devoid of any distortion problems.

As already mentioned in Section 6.1 the results of the follicle distribution process are dependent on follicle density in comparison to surface feature size. The follicle distribution is however independent of the surface topology, since the algorithm works directly on the polygonal mesh.

One complexity of our model is the pipeline design. The follicles are placed first and then the parameterization is overlaid and the key feathers added to complete the feather coat design. This is done, because the feather coat cannot be constructed without each follicle having a set of parametric coordinates. However, once the feathers were in place, if there was a desire to move a follicle, there could be two outcomes: the new follicle can obtain both a new position and a new parametric location or just a new position. In the former case this means the interpolants used to construct the feather and consequentially the type of feather may change. In the latter case it means simply a location change. Either way this is more a development and design issue than one directly associated with the methodology for creating feather coats.

While our feather packing algorithm uses the control vertices of the key barbs and efficiently packs the feathers, it does not handle feathers that do not have key barbs (i.e. down). If the entire feather coat is composed of down, the feathers will simply point in the direction of the surface normal and no packing will occur. This is representative of the natal down of baby birds. However, if the feather coat has some down and some contour feathers, there is no current method for packing against the down feathers. The postnatal down that is commonly found on some types of birds often acts as an insulation layer and is not visible on the surface. This layer can easily be simulated by an offset from the original mesh, but is unaccounted for in our model. Since the focus of our model was on the visible feathers, this is not a limitation. The one problematic case occurs with contour feathers having large plumulaceous portions. This plumulaceous portion is not adjusted with the key barb elevation in the packing algorithm and in some cases may fall outside the bounding box formed by the key barbs. In this case either the plumulaceous portions can be packed one barb at a time as with many hair systems, or either ignored and not drawn if not visible (feather present on top) or left to its natural state otherwise (no feather present on top).

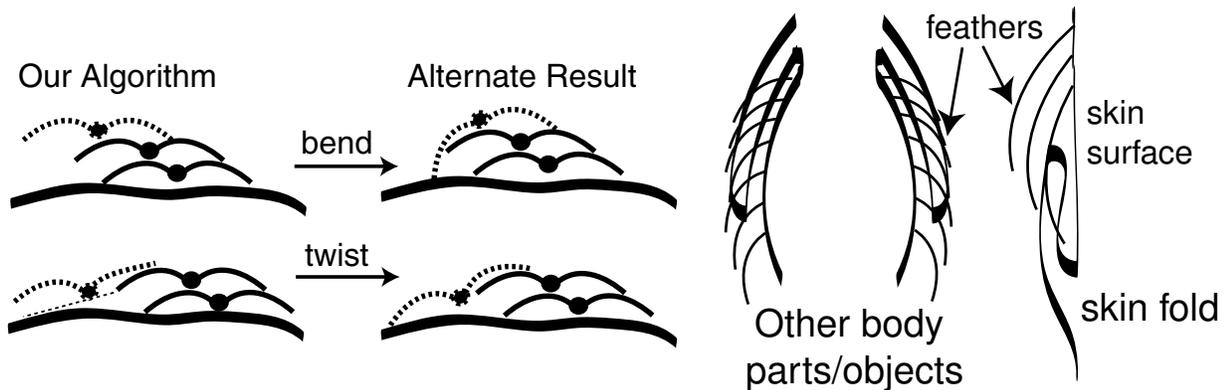


Figure 9.16: Cross-section of packed feathers. **Left:** Results from our algorithm. **Right:** An alternate plausible result.

Figure 9.17: Examples where the packing algorithm results in interpenetrations. **Left:** Other body parts or objects. **Right:** Skin folds.

As shown, our packing algorithm results in a plausible feather arrangement in which the feathers do not interpenetrate each other or the surface. However, there are two reasons why it may be less than optimal: vanes cannot gain curvature or twist and the control vertices are an upper bound on the bounding volume. First, once the shaft is maximally rotated, the barbs can be elevated, but never depressed. This means, that the feather vane can flatten or even take on a slightly reverse curvature, but can never gain curvature or twist. This is to prevent the feather from loosing its shape completely, as there is some stiffness to the barbs and

collectively the vane in real feathers. Thus in the situation shown in Figure 9.16, extra space beneath some feathers may occur, where a rotation of the feather coordinate frame in these situations may fix this problem. While twisting along the length of the shaft seems physically improbable, a twist of the feather within the follicle is more probable though physically restricted. This is also not part of our current model, but we have found it to be a rare occurrence. Second, as the control vertices are the upper limit on the bounding volume of the feather it is not an precise limit. Thus, there may be some extra “space” between the feathers. However, the only method that would fix this problem is to do complete collision detection using complete geometry and while this is computationally intensive, it may work at high levels of detail. However, either space or interpenetrations would occur at lower levels of detail, as the approximated feather geometry may not fit tightly together.

Finally, the packing algorithm does not handle packing against other objects or body parts including skin folds. It can handle convex as well as concave model surfaces by checking neighbouring polygons normals in addition to the surface direction vector, but it is highly inefficient to check every polygon or polygons of other models. Thus, currently there is no way to handle interpenetrations with polygons, that are not in a small region (feather size) of the original follicle, with other objects or with feathers attached to these surfaces. For example, if a large skin fold occurred, such that the skin folded over the region the feather covers, but does not overlap the follicle then the result of the packing algorithm may be that the feather interpenetrates the fold of the skin. This is because no checks are done against polygons far away (beyond the area of the feather) from the triangle the follicle is attached to as shown in Figure 9.17.

Comparing how we arrange feathers with other current work yields some similarities and differences. Previous work has distributed follicles using a force-repulsion based approach [CXGS02, Tur92]. Chen et al. [CXGS02] tailored this method to increase follicle density in regions of higher curvature, though did not attempt to produce stretched hexagonal patterns. The force-repulsion method produces comparable results to our own, though it can be unstable and is slightly harder to control. Only one previous work, Chen et al. [CXGS02] has attempted to detect and prevent collisions among feathers automatically. Pixar in *For the Birds* [Rob01] also did collision detection between feathers on adjacent objects, but it was done by manually placing bounding volumes. Collisions between feathers were not mentioned, and seemed to be irrelevant due to the surface shape, feather size and angle. Chen et al. [CXGS02] use recursive collision detection scheme for off-line collision detection and response. Their algorithm worked on simplified feather geometry, therefore it is presumed that there is also extra space in their final arrangement. They quote taking 30 minutes



Figure 9.18: Top: Photos of moulting male Eclectus Parrot from [Mos03]. Fully developed feather coat shown on far right. **Bottom:** Simulated moulting of similar coat using our feather coat morphogenesis methodology.

to compute collisions and orientation of 3500 feathers on a PIII 864MHz machine with a GeForce 3 graphics card. Comparing this with our medium size feather to follicle arrangement in Figure 9.12, 3500 feathers would take approximately 9 minutes with our algorithm on a similar machine. With smaller feathers it would take as little as 2.8 minutes and still have the feathers cover the surface in an overlapping manner.

9.1.3 Alteration Component

The alteration component of our methodology not only provides a means of simulating morphogenesis, but also validates the use of our methodology in a changing environment as shown in Chapter 8. We have already shown that our methodology is capable of both simulating feather coat morphogenesis as well as creating proper feather arrangements on moving surfaces (skin movement) in Chapter 8. In this section we will compare our morphogenesis results with those of real birds, as well as analyze the techniques used. Figure 9.18 shows photos of a male Eclectus Parrot moulting [Mos03], as well as a simulated moulting feather coat created with our methodology. On the top are three photos of the male Eclectus during moult and a fourth photo with fully developed feather coat. On the bottom, are two examples of simulated moult. The bottom left shows a region, and a magnification of the moult region using increasing u parameter moult. The bottom right shows a region and a magnification of that region with random moult. Comparing the top and bottom image moult regions bears some likeness to each other. The main difference in these images is the rendering of the feathers.

Alteration Component Analysis and Scope

There are only two minor complications with how we achieve feather coat morphogenesis in a dynamic environment: frame-to-frame packing and rendering of feather growth.

During animation of the skeleton and skin, the follicles move, the coordinate frame of the feather changes and the feathers are repacked. The feathers are repacked at each frame of the animation, to ensure no invalid feather arrangements (i.e. interpenetrations). However, not only is this time-consuming, but depending on the implementation of feather parameterization can lead to lack of coherence in feather arrangement across frames. With our current implementation, the follicles are not re-parametrized as they move. Therefore, large skin deformations in relation to the follicle density and feather size can lead to interpenetrations of feathers because the parameterization may no longer accurately reflect the ordering of the feathers. But, in most realistic movements, the skin does not undergo extreme deformation and therefore this does not tend to be a problem. Also, this ensures frame-to-frame coherence of the feather arrangement because the ordering is consistent. If extreme skin deformation is necessary, a re-parametrization of the follicle location would also be necessary to prevent interpenetrations, but then this leads to lack of coherence between frames.

As rendering is not the focus of this thesis, rendering of the feather during growth has not been completely addressed. The texture coordinates are assigned to the feather as if it was fully developed, only the substructures are arranged differently during growth. For some of the feather development, the feather substructures are contained within the feather shaft. Therefore the textures of the feather are not visible. Once the feathers open the texture is immediately visible as with real feathers. However, in the pin feather state the feather is covered by a sheath, which is more opaque and obscures the texture of the shaft and feathers. It also sometimes adds a blue or gray colouration to the shaft. This has not been addressed in our model.

No other work currently exists on feather coat morphogenesis. No one to the authors knowledge has attempted to model feather growth or feather coat moult, even though it greatly impacts the look of the resulting feather coat. As well, there has been little work on addressing modelling of feather coats during animation. Details of how the feather coat responds to animation have not been addressed by [CXGS02]. Sony Pictures Imageworks [EAB⁺02] addresses many of these complications introduced by movement of the skin model through the use of manual intervention. Finally, while styling has become an area of research with regards to hair, it is unexplored in the domain of feathers.

9.2 Discussion and Analysis

Integrating the three components for use in creating a feather coat is neither theoretically complicated nor developmentally trivial. The design of the components has made integration relatively simple, though there are some subtleties to implementing the integration. The focus of this thesis is on the functionality of the methodology for creating feather coats and feather coat morphogenesis, rather than the usability of the particular development framework in which the methodology has been implemented. Thus, the methodology itself is the contribution of this research and the functionality forms the basis for the validation of the contribution.

9.2.1 Integration Challenges

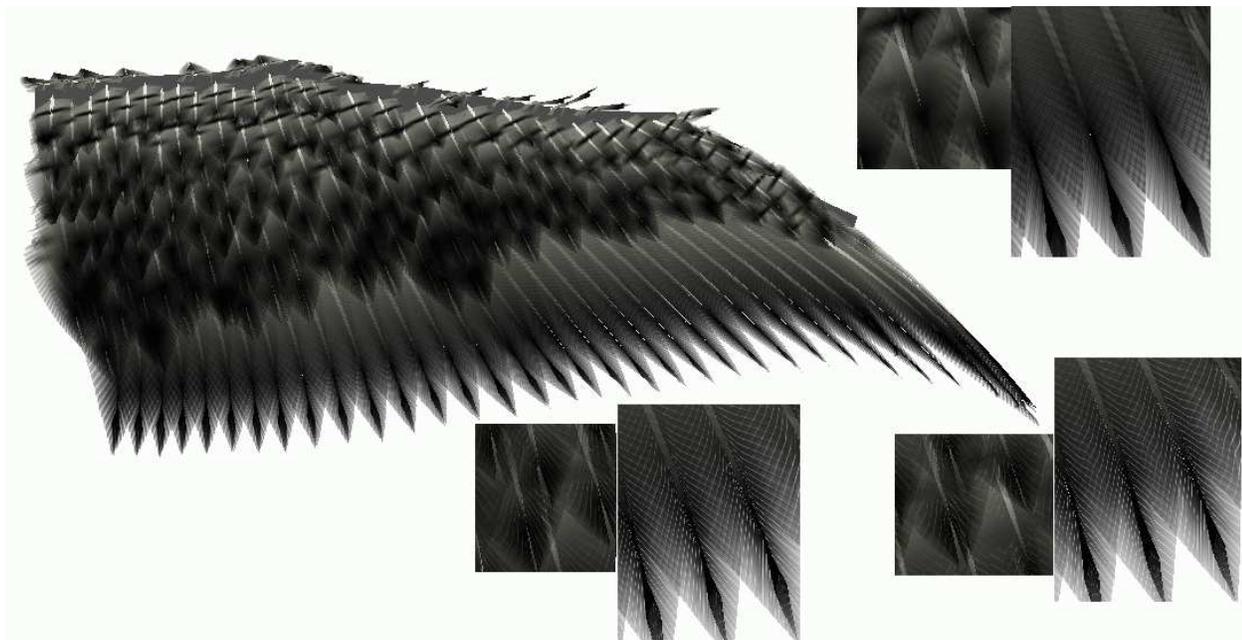


Figure 9.19: Our modelled wing with 210 feathers at a 100% geometry (120,000 polygons). In the top left are two magnifications of this same LOD. The bottom middle and right are two magnifications of 60% and 20% of the geometry respectively. A frame rate of 4.8/7.8/13 fps for 100/60/20 percent detail were obtained on a GeForce 3 graphics card.

The images of the wing shown in Figure 9.19 were constructed using two feather tracts as shown in Figure 8.6; one for the primary and secondary (large bottom feathers - comparable to the Alar tract in real birds), and a second slightly overlapping tract for the coverts (upper smaller feathers - comparable to the

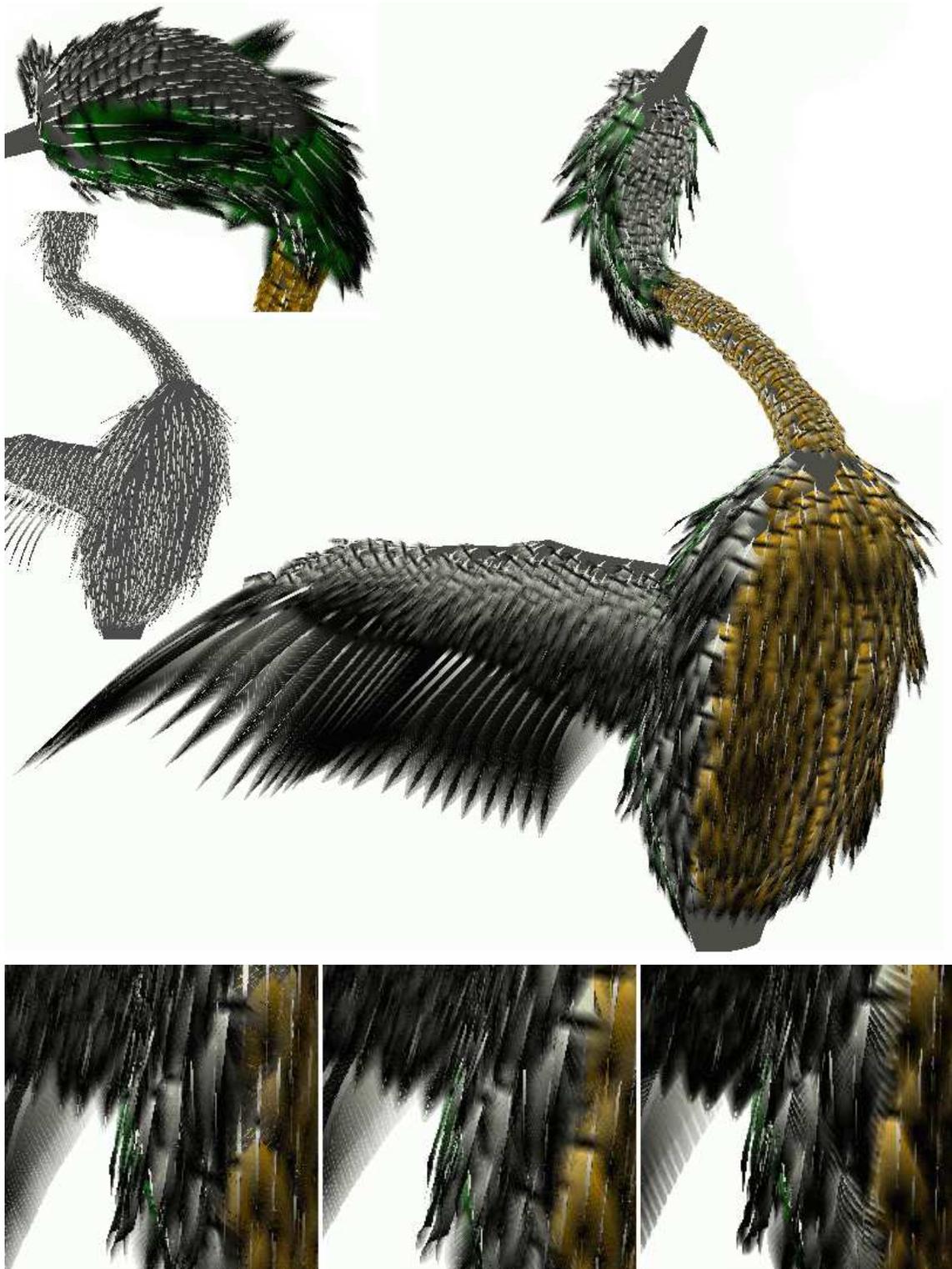


Figure 9.20: Top: A fully feathered bird with 3148 feathers at 60% of the geometry with 830,000 feather triangles and 3732 model triangles. On a GeForce 3 we obtained 0.69 fps. In the top corner is a close-up of the styled head and a inset of the follicle distribution on the body. Bottom: a close-up of the body/wing area at 20%, 60% and 100% of the geometry.

9.2. DISCUSSION AND ANALYSIS

Humeral tract). Six key feathers were placed on the boundary of the upper tract and three key feathers were placed along the lower tract. The feathers are in a packed state. We obtained a frame rate of 4.8 fps at 100% geometry, 7.8 fps at 60% geometry and 13 fps at 20% geometry on PIII 1 GHz with a GeForce 3 graphics card.

Figure 9.20 shows a fully feathered bird with 3148 feathers at 60% of the full geometry (830,000 triangles). The underlying model has 3732 triangles. An inset of the bird with only the shafts of the feather are drawn to show the density and direction of feathers used. The feather coat was packed in 35.1 seconds. With hardware multi-texturing we obtained a frame rate of 0.69 fps on a 1 GHz Pentium III with a GeForce 3 graphics card. An inset of a magnification of the head is shown in the top left corner. The feathers on the back and side of the head have been styled by specifying unnatural direction vectors and thus do not follow the normal contour of the head. The model was constructed using 18 tracts and 80 key feathers. The 18 tracts consisted of 5 for the head, 3 for the neck, 3 for each wing and 4 for the body. The feathers on the head were defined using four key feathers for each of the top, bottom and upper neck tracts and five key feathers were used to defined the feathers on each of the side head tracts. The feathers on the neck were specified using four key feathers for each of the three tracts. Each wing was specified with six key feathers on the two upper wing tracts and three key feathers were used for the lower wing (flight) tract. Four key feathers were used for each of the body tracts. Using 20% of the geometry of the feather coat created 276,000 feather triangles and ran at 1.35 fps on the same machine. At 100% of the full geometry 2.2 million feather triangles were created and ran at 0.31 fps. Below the image of the full bird are three magnifications of the area where the wing joins the body at varying levels of detail. Left to right are the same area at 20%, 60% and 100% of the geometry. Note the maintenance of detail, feather shape and reflectance. In certain regions there are bald patches where no feathers/follicles exist. In these areas a denser follicle distribution may be required. In nature areas like the one under the wing have less dense contour feathers (similar to our tract density), however often down feathers are present in these areas to hide the skin. Also we must note how smaller more dense feathers naturally pack more closely as those shown on the neck. In some regions on the body stiff flat feathers can be seen protruding slightly from the feather coat. We must note that since they are not single feathers that the packing is correct and that this protrusion is caused by the extreme coarseness of the underlying model (i.e. sharp edge) and a large feather size relative to the sharp feature edge of the model. This affect is similar to the results shown in Figure 9.12.

Unifying the three components into a single methodology involves specifying only one more detail;

how to handle interaction between tracts. Since feather tracts can be disjoint or overlap, there are a few ways of handling this issue. There are two possible complications to address across tracts: follicle distribution and follicle parameterization. Generally in real birds, there is an apteria region of no contour feathers in-between the feather tracts. This means in general the contour feather tracts are disjoint and across tract follicle distribution is no longer a problem. If however, the tracts are not disjoint, the polygons that contain follicles from multiple tracts may have clusters or clumps of follicles. This can be desirable if the different tracts represent different types of feathers. However, if the tracts are the same feather type a more uniform distribution is typically desired. This can be accomplished by running the follicle distribution algorithm a second time, on only polygons of multiple tracts. This was not needed for our examples, as we maintained apteria region representation and had two sets of tracts: one for down feathers and one for contour feathers.

Parameterizing the follicles across tracts is slightly more challenging. This is required not only to compute the interpolated feathers from the key feathers, but also to define the packing for the ordering across tracts. We have chosen to handle the feather interpolation by maintaining that the tract divisions are formed to define regions of differing feather types and thus the boundary of the feather tracts will always have key feathers. If an overlapping tract exists the same key feathers will also be located on this tract. Across tract packing is handled by defining a global ordering, either by ordering the tracts or the individual follicles using projection to a body axis vector. This is required for determining which feathers to pack against at the tract boundaries.

9.2.2 Rendering and LOD Challenges

Our level of detail (LOD) or rendering algorithm maintains the shape and look of the feather smoothly while reducing geometric detail as shown in Chapter 7. However, there are two special cases where the results can be less smooth than desired: plumulaceous feathers, and dramatically curved barbs. While combining geometry of adjacent barbs looks satisfactory for higher levels of detail, at very low levels of detail the combined triangle strips can cause “popping” as the levels of detail are transitioned for plumulaceous feathers since they are defined as randomness in the barb curvatures. This is due to the lack of coherence between adjacent barbs. Also, at lower levels of detail an alpha texture is repeated to simulate the transparency between the individual barbs. As long as the barb is represented by a sufficient number of vertices, dramatically curved barbs can be maintained, but once curves are coarsely approximated, the simulated barb straightens and so too does the alpha map. This results in straightening of dramatically curved barbs at low levels of detail.

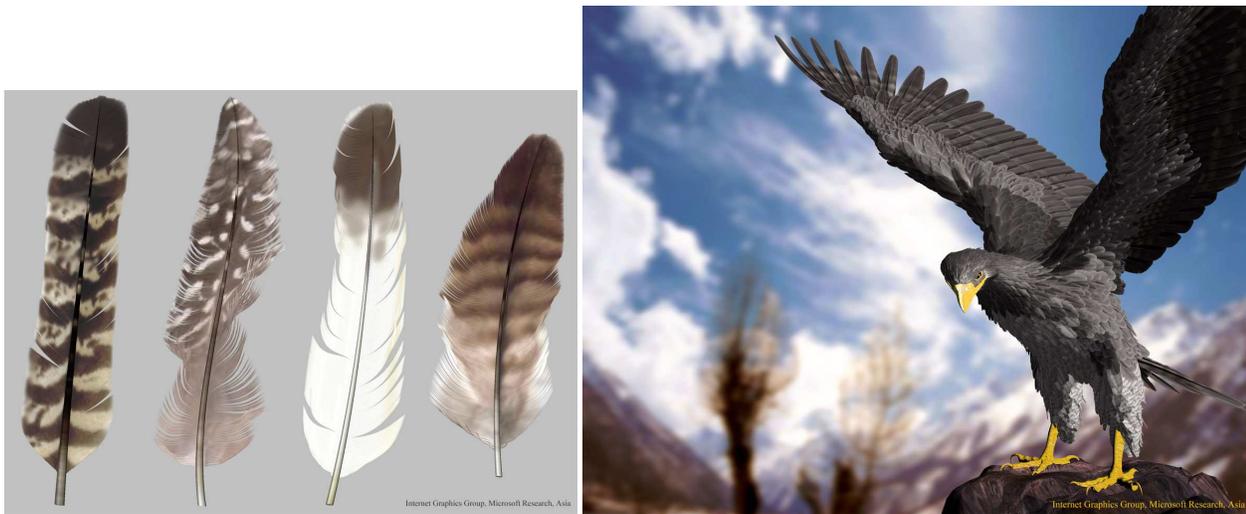


Figure 9.21: Two images from Chen et al.'s [CXGS02] work on modelling and rendering feathers. **Left:** Some example feathers. **Right:** A feathered Eagle. These images were obtained from the authors and used with their permission.

However, since both of these special cases occur at very low levels of detail, usually the viewer is sufficiently far away not to be able to discern these details. The silhouette and texture which are discernible at these levels of detail are maintained. Therefore, we do not deem this to be an inadequacy of the methodology.

9.2.3 Comparison with Other Models

Comparing the integrated feather morphogenesis methodology with other models of feather coat morphogenesis is impossible because there are none. We can compare our model of feather coats with the concurrent research of Chen et al. [CXGS02] and Sony Pictures Imageworks [EAB⁺02] on modelling feather coats. As already stated in Section 9.1.1 our model is more flexible than both of these models. Not only can we model feather types other than contour feathers, but we can also model feathers of various ages. Controlling the feather structure is also more intuitive with our model biologically parameterized model. Sony's model contains "hundreds of parameters" [EAB⁺02, p. 59] and Chen et al.'s model is based on the less-intuitive L-systems.

Our model is more precise and efficient. Our model positions the follicles more precisely than either of the above models which we have shown to have a significant impact on the feather coat pattern. Not only does it allow for the creation of follicle distributions similar to Chen et al.'s and Sony's work, but it is more controllable allowing for the creation of the stretched hexagonal distributions. While there is little known

about how Sony’s model handles feather-to-feather collisions, the efficiency of our model exceeds that of Chen et al.’s. Since Chen et al. uses raw collision detection, there is the potential for their model to result in more precise packings of feathers, but they do state a simplified geometry is used which means there is some imprecision involved. They also do not try to account for bending or flattening of the feathers as they are packed unlike with the barb adjustment in our model. The built-in multiple-level-of-detail aspect to our methodology is also an efficiency advantage. Again Chen et al. mention a simplified geometry for collision detection, but their rendering model does not seem to scale to lower resolutions.

However, the rendering method used in both of these models is superior in some ways to ours. Sony’s model uses some of their “hundreds of parameters” to account for more accurate transparency, translucency, back-lighting and various other optical and lighting effects which are not present in our model. Chen et al. [CXGS02] use a precomputed Bidirectional Texture Function or BTF to achieve more detail in closer views of the feathers as shown in Figure 9.21. However, either because of this or the lack of level of detail the computation required for rendering is much greater than ours.

9.2.4 Contribution and Validation

The main contribution of this research is the feather coat morphogenesis methodology. Currently, there are no other methods for simulating feather coat morphogenesis. Since this research began, two other methods have developed for modelling feather coats, both with comparable results. However, as previously stated, our method is more flexible, controllable and in many aspects more efficient, as well as includes the morphogenesis aspect.

We have shown that our methodology is capable of modelling a range of feather types and ages, semi-automatically generating feather collections and plausible arrangements of feathers as well as providing a basis for modelling changes in the feather coat over time (feather coat morphogenesis). We have shown that our methodology is flexible and fairly efficient in comparison to other concurrently produced methods for feather modelling. We have shown that our “parameter specific” model is more flexible and controllable and that the use of our LOD approach adds efficiency. As well, we have shown that the methodology responds appropriately in a changing environment (animation of the skin surface). Therefore our thesis statement has been effectively supported: *modelling feather coat morphogenesis using a multi-resolution methodology that is feather parameter specific will best facilitate the creation of “a realistic looking” feather coat for the purposes of animating the follicle surface or “skin”*.

9.3 Summary

This chapter reviewed the appropriateness of feather coat morphogenesis methodology presented in Chapters 5 – 8. It discussed in detail the contributions, capabilities, flexibilities and constraints of each of the three components: feather structure, feather arrangement and feather alteration as well as the methodology as a whole. The results of each component were detailed and compared with real feathers and feather coats and other current work. Any scope and restrictions of the methodology were clearly outlined. The methodology was validated in terms of the thesis statement and the contributions of this research were clearly outlined.

Conclusions and Future Work

Modelling of feathers and feather coats was until very recently a largely unexplored topic in computer graphics. With this thesis not only has a strong justification for exploring the intricacies of modelling feathers and feather coats been provided, as also shown by other recent research, but that a basis for a solution to this problem has been provided. Modelling feathers, feather coats and most importantly feather coat morphogenesis, is as demonstrated in this thesis, a very involved problem.

This thesis presented a methodology for modelling feather coat morphogenesis, in particular for creating “realistic-looking” feather coats for the purposes of animating the skin. The methodology proposed and presented was a multi-resolution parameterization, specific to feathers and feather coats. This methodology was then validated to both handle animation of the skin and changes in the feather coat to produce results comparable to real feathers and feather coats. This thesis defined a particular methodology in terms of a set of procedures used to model feather coat morphogenesis. The focus of this thesis is the methodology or process for modelling the feather coat and its morphogenesis, rather than the system implementation of the methodology. Validating the methodology includes evaluating the capabilities, scope and results of the methodology, rather than the usability of a particular implementation of the methodology. Therefore, the main contribution of this thesis is the methodology and the validation of the methodology for feather coat morphogenesis.

To evaluate the methodology in the context of modelling natural phenomena, four basic principles (see Chapter 1) were identified: tailorability, level of detail, stochastic elements and dynamics. As the intent was to model a vastly complex, non-repetitive, changing surface, a balance between automation and

control was needed. With too much control, a large set of parameters become overwhelming causing the design of every feather detail to be far too tedious. With too much automation, the desired result may become unattainable or unpredictable given the input parameters. Secondly, incorporating level-of-detail aspects into the model, offered a natural way of reducing complexity when unneeded in order to improve performance of the results. Thirdly, using stochastic elements allowed for the simulation of outside influences, such as splits in the feather vane and plumulaceous feather barbs. However, care was taken not to overuse stochastic behaviour, especially where predictable results were expected. Finally, designing the model to incorporate development of the feathers allowed for natural modelling of the structural development of the feather coat known as *morphogenesis*.

In modelling a realistic feather coat, decisions were made about what aspects of the highly complex real feather coat had the largest impact on the resulting appearance. The decision to model feather coat morphogenesis instead of static feather coats was based on the hypothesis that structural changes of the feather coat have a significant affect on the appearance of the feather coat. In this thesis, three more aspects that were hypothesized to have a significant affect on the appearance: feather structure, follicle distribution and feather orientation. Each aspect was shown not only to create results comparable with real feather coats, but also to have a significant impact on the final appearance of the feather coat and feather coat pattern.

First, a parameterization for individual feathers (see Chapter 5), based on the biological substructures of real feathers was presented. As a result, this parameterization contained parameters specific to modelling feathers and feather coats. The parameterization consisted of a small manageable number of parameters, that allowed for the creation of a wide-variety of feather structure types with smooth interpolation between the different types. It also accommodated modelling feathers at various stages of development and smoothly interpolating between the different ages. The resulting feather structures were compared with structures of real feathers and the interpolation between the structures were shown to be consistent with the continuum of feather types found in real birds. As well, the sequencing of the parameters used to simulate growth also produced results comparable with real feather growth.

Second, a process for distributing follicles and automatically computing the final arrangement of the feathers was also presented (see Chapter 6). User-defined follicle density, surface direction vectors and a parameterized polygonal mesh subdivided into feather tracts, were used to control the the density distribution of the follicles on the surface. The mesh-based algorithm automatically maintained the pseudo-hexagonal distribution of follicles found in real feather coats. The resulting follicle distributions were compared with

real follicle distributions and shown to produce similar results. The impact of the follicle distribution on the pattern of the resulting feather coat was analyzed and shown to be a significant aspect of the feather coat appearance.

Third, a method for automatically computing the resulting feather coat arrangement was presented. Using the surface parameterization, the user-defined surface vectors and the normal to the surface at each follicle a coordinate frame was defined and used to *pack* the feathers in a layered fashion. The feathers were *packed* using a subset of the control vertices of previously packed feathers. Using this subset of points as a height field an initial position of the feather was determined. From this initial position the barbs of the newly packed feather were adjusted to prevent interpenetrations. By restricting packing or feather angles and modifying surface direction vectors the user was able to style the feather coat. The resulting feather arrangements were compared with real feather arrangements and were shown to produce plausible realistic feather coat results. Again, it was shown how changes in the feather arrangement not only changed the overall feather pattern, but changed both the area covered and the silhouette shape of the feather arrangement.

Finally, in addition to showing that the proposed methodology was capable of modelling these three attributes, it was also important to show it could both produce changes in the feather coat over time, and respond appropriately to changes in the skin surface. Using the parameterization to specify various stages of feather development and a user-defined moulting sequence, the development and replacement of the feather coat was simulated. The moulting results were shown not only to be similar to real feather coats, but also to have a large impact on the resulting appearance of the feather coat. Being able to animate the resulting model is often desirable in computer graphics. Thus, showing that the methodology responds appropriately under skeletal animation validates the methodology's flexibility and applicability. Using skeletal movement and the corresponding skin movement it was shown that both the follicle density responded appropriately to stretching and compression of the skin, and that the associated feathers moved plausibly over each other. A basic model for rendering the feathers and feather coat was also provided. This model used an anisotropic reflection model, alpha blending and multi-texturing to both simulate the presence of barbules and incorporate acquired and simulated feather textures. The rendering method also added detail to lower geometric levels of feather representations.

10.1 Main Contribution

The computer graphics problem addressed in this thesis is a methodology for defining and managing (i.e. creations, editing and control) of an intricate, geometrically complex, diverse, changing surface in the context of feather coat modelling. The solution presented was a feather parameter-specific, multi-resolution feather coat morphogenesis methodology for creating “realistic looking” feather coats for the purposes of animating the follicle surface or “skin”.

As mentioned previously, the main contribution of this work is the methodology for feather coat morphogenesis. While no other such methodologies exist and only a very recent few for non-changing feather coats, the significant aspects of this entirely novel methodology are first the feather parameterization including its incorporation of growth and second the packing algorithm. Incorporating growth into the parameterization of the feathers and ensuring a smooth interpolation between types and ages of feathers made modelling the rest of the feather coat and its morphogenesis simpler. Secondly, the parameterization for the feathers provided an implicit bounding volume that was used to reduce the complexity of collision detection and collision resolution (packing) of the feather coat in an efficient manner novel to the graphics community.

The particular implementation of the methodology both allowed for testing and validation of the methodology, and produced results for visual comparison with real feathers and feather coats, but itself was not part of the thesis contribution. In order for this methodology to be used as a tool for the public to create feather coats and feather coat morphogenesis, proper implementation of the methodology would be required. This would include studying how people use which parameters and in what combinations they may want to change them to create various results. While designing a parameterization and set of controls is never independent of their use, this thesis showed that the biological-based parameterization is capable of creating a wide range of feathers and feather coat designs and therefore should not be a limiting factor in the development of an implementation for this purpose.

10.2 Future Work

While it was shown that feather structure, follicle distribution, feather orientation and feather coat morphogenesis all have a significant impact on the resulting appearance of the feather coat, there are other aspects of the feather coat that also affect its appearance. The most significant of these is the rendering of the feathers

and the incorporation of various optical effects. This is one area for future work. Not only would building a reflectance model based on feather subsurface properties be useful, but also incorporating feather-to-feather and feather-to-surface shadowing as well as subtleties such as shaft shadowing on the barbs could also affect the appearance of the resulting coat. Other areas of future work may include:

- using an automatic parameterization method for parameterizing the model's surface
- investigating and developing a pattern synthesis algorithm based on the substructures of the feather model and the transport and concentrations of pigments defined by ornithologists.
- designing a set of interactive tools for styling the feather coat
- incorporating post-design editing details such as movement of follicles
- incorporating packing of down/plumulaceous feathers or use of a down layer
- use of non-linear interpolation for feather interpolation and generation
- allowing feather twist or follicle rotation and enhanced physical simulation (i.e. friction between feathers, flattening of feathers under weight or computed force).
- collision detection with other body part parts, objects and skin folds
- incorporating a reduced geometric representation for plumulaceous feathers at very low levels of detail
- more automatic modelling from images

10.3 Conclusions

This particular methodology was designed to create realistic-looking feathers and feather coats, including the changes that a feather coat undergoes due to moulting. It is hoped that this methodology may be useful to animators and character designers for film production, teachers and instructors of ornithology and paleontology, with advances in graphics hardware - game developers and with precise data and inclusion of pattern synthesis possibly ornithological researchers. It's also hopeful that the graphics community can use aspects of this methodology as a basis for the design of other subject-specific methodologies with similar aspects such as complex scenes and large amount of changing, geometric detail. Finally, as the first methodology for feather coat morphogenesis, this research provides a basis for other methodologies in this area.

10.3.1 Supplemental Material

Note: Please see <http://www.cs.ubc.ca/~streit/Thesis.html> for supplemental material.

10.3. CONCLUSIONS

Glossary

afterfeather	The feather-like structure that branches from the superior umbilicus. Often resembles, but is smaller than the main feather, although in some cases it is nearly the same size. Also called hypopennae, hyporachis aftershaft.
anterior	Situated or located near the front. On an animal relating to or situated near the head.
apteria	A featherless region or a region devoid of any contour feathers. See also pterylae .
attenuate	Describing shape where the object is more narrow than long and only tapers to one end (distally).
barbicels	The structures that branch outwardly from the barbules. These structures are typically only present on larger contour or flight feathers. Also called hooks , hooklets or hamuli .
blastema	A small mass of living substance capable of growth.
BRDF	Bidirectional Reflection Distribution Function. Used to describe the resulting illumination at a surface location given the incoming and outgoing light and viewing directions.
BTF	Bidirectional Texture Function. Used to increase rendering performance and results - assuming a limited set of light and viewing directions.
caudal	see posterior .
cephalad	see anterior .
distal	Direction away from the midline of the body or body segment.
dorsal	Located on the upper surface. On an animal relating to or situated near the back.
epithelial	An animal cellular tissue covering the external and internal surfaces of the body.
feather coat	The collection of feathers on a bird's body at a particular time.
feather sheath	The outer most layer of the feather during growth. A thin opaque layer with a blue, grey or white tinge.

feather tract	See pterylae .
frontal	The plane which divides the body that is perpendicular to the transverse and saggital planes.
hamuli	See barbicels .
hooklets	See barbicels .
inferior umbilicus	Opening in the calamus through which blood supply enters the feather during growth.
keratin	A fibrous protein that is found in the outer region of skin and in feathers, hair, nails, hooves etc. It is resistant to bacterial decay and even once growth is complete.
lanceolate	Describing shape where the object is more narrow than long and tappers to both ends.
medial	Describing a comparative relationship of locations within the frontal plane that are closer to the midline.
midline	A line oriented along the main axis of the body such that a saggital plane placed on this line would divide the body into two equal bi-lateral halves.
monopodial	Having or involving the formation of off-shoots from a main axis.
morphogenesis	The evolutionary development of the form of an organism. The development of form in an organism during growth. See also morphology .
morphology	A branch of biology concerned with form and structure of organisms.
moult	The cyclical or periodic loss and replacement of the plumage.
NPR	Non-photorealistic rendering. Rendering in a style that produces non-photorealistic looking images. This style is often used to convey an artist-like or impressionistic effect.
obovate	Describing shape - upside-down egg-shaped.
ovate	Describing shape - egg-shaped.
pennaceous	Stiff, flat portion of feather or barb. Barbicels are present causing the barbs or barbules to interlock.
peripheral	Describing a comparative relationship of locations outside the frontal plane that are further from the midline.
pitch	Directional change or rotation along an axis (within the transverse plane) connecting the wing tips when the bird or aircraft is in flight position.

pith	Spongy tissue (air-filled cells) formed from the keratinization of long epithelial cells.
plumage	The set of feathers acquired after a particular moult. See also feather coat .
plumulaceous	Loose, “fuzzy” portion of the feather or barb. Barbs or barbules are non-interlocking. Usually no barbicels are present.
posterior	Situated behind or near the rear. On an animal relating to or situated near the buttocks.
proximal	Direction toward the midline of the body or body segment.
pterology	The study of Pterylosis. Also named pterylography, pterology, ptilology.
pterylae	Regions of contour feathers, usually having a well-defined ordering. The eight main pterylae regions are capital, spinal or dorsal, humeral, ventral, caudal, alar, femoral and crural. See also apteria , pterylosis .
pterylosis	The arrangement of contour feathers in orderly groupings of feather tract or pterylae . Also called ptilosis.
rectrices (sing. rectrix)	Large flight feathers found on the tails of birds.
remiges (sing. remex)	Large flight feathers found on the wings of birds.
roll	Directional change or rotation along an axis (within the saggital plane) connecting the head and tail when the bird is flight position.
feather rudiment	The start of feather development.
saggital	A plane oriented along the main axis of the body dividing it bi-laterally into right and left parts.
spatulate	Describing shape - short broad tip.
superior umbilicus	Opening where the rachis and calamus meet on the shaft, from which the afterfeather grows.
transverse	A plane oriented perpendicular to the main axis of the body dividing it into posterior or anterior parts.
tropism	The involuntary response of an organism either toward or away from external stimulus.
ventral	Located on the lower surface. On an animal - opposite the back.
yaw	Directional change or rotation along an axis perpendicular to both the saggital and transverse plane or pitch and roll axes.

Non-Computer Science References

- [Bar81] J. B. L. Bard. A model for generating aspects of zebra and other mammalian coat patterns. *Journal of Theoretical Biology*, 93(2):363–385, 1981. 36
- [Cle70] Mary H. Clench. Variability in body pterylosis with special reference to the genus passer. *The Auk*, 86:650–691, 1970. 19, 21
- [Com37] Lawrence V. Compton. The pterylosis of the falconiformes with special attention to the taxonomic position of the osprey. *University of California Publications in Zoology*, 42(3):173–211, 1937. 19
- [CPPR87] G. Cocho, R. Perez-Pascual, and J. L. Rius. Discrete systems, cell-cell interactions and color pattern in animals - conflicting dynamics and pattern formation. *Journal of Theoretical Biology*, 125(4):419–447, 1987. 37
- [Ede60] M. Eden. A two-dimensional growth process. *In Proceedings of the Fourth Berkeley Symposium on Mathematics, Statistics and Probability*, 4:223–239, 1960. 36
- [Gil95] Frank B. Gill. *Ornithology*. W.H. Freeman and Company, New York, 1995. 9, 15
- [Got93] M. E. Gottlieb. The vt model: A deterministic model of angiogenesis. *IEEE Transactions on Biomedical Engineering*, 1993. 37
- [Har93] L. Harrison. *Kinetic Theory of Living Pattern*. University Press, New York, 1993. i, 36
- [Lin68] A. Lindenmayer. Mathematical models for cellular interaction in development, parts i and ii. *Journal of Theoretical Biology*, 18:280–315, 1968. 37
- [LS72a] Alfred M. Lucas and Peter R. Stettenheim. *Avian Anatomy Integument, Volume I*. U.S. Government Printing Office, Washington D.C., 1972. 2, 10, 11, 12, 13, 15, 17, 19, 21, 23, 25, 27, 105, 107
- [LS72b] Alfred M. Lucas and Peter R. Stettenheim. *Avian Anatomy Integument, Volume II*. U.S. Government Printing Office, Washington D.C., 1972. 18, 21, 22, 24, 25, 126, 127
- [Mea86] P. Meakin. A new model for biological pattern formation. *Journal of Theoretical Biology*, 118:101–113, 1986. 36
- [Mei82] H. Meinhardt. *Models of Biological Pattern Formation*. Academic Press Inc., New York, 1982. 36
- [Mei95] H. Meinhardt. *The Algorithmic Beauty of Seashells*. Springer-Verlag, 1995. 38
- [Mil28] A.H Miller. The molts of the loggerhead shrike, *lanius ludoviciana linnaeus*. *University of California Publications in Zoology*, 30:393–417, 1928. 18
- [MK87a] H. Meinhardt and M. Klinger. A model for pattern formation on the shells of molluscs. *Journal of Theoretical Biology*, 126:63–89, 1987. 36, 38

- [MK87b] H. Meinhardt and M. Klinger. Pattern formation by coupled oscillations: The pigmentation patterns on shells of molluscs. *Lecture Notes in Biomathematics*, 71:184–198, 1987. 36, 38
- [Mos03] Dean Moser. Shades of red and green. http://www.shadeselectus.com/Moulting_Picts.htm, 2003. 136
- [Mur81] J. D. Murray. On pattern formation mechanisms for lepidopteran wing patterns and mammalian coat markings. *Philosophical Transactions of the Royal Society of London B*, 295:473–496, 1981. 36
- [Mur93] J. D. Murray. *Mathematical Biology*. Springer-Verlag, New York, 1993. 36
- [NM85] B. N. Nagorcka and J. D. Murray. The role of reaction-diffusion system in the initiation of primary hair follicles. *Journal of Theoretical Biology*, 114:243–272, 1985. 77
- [Ode81] G. M. Odell. The mechanical basis of morphogenesis: I. epithelial folding and invagination. *Developmental Biology*, 85:446–462, 1981. 36
- [Ons01] Inte Onsman. The myth of the tyndall effect in blue bird feathers. <http://www.euronet.nl/users/hnl/tyndall.htm>, 2001. 22
- [oPA01] Department of Physics and Astronomy. Patterns in nature light and optics. <http://www.acept.asu.edu/PiN/rdg/readings.shtml>, 2001. 22
- [Ost83] G. F. Oster. Mechanochemistry and morphogenesis. *Biological Structures and Coupled Flows*, pages 417–443, 1983. 36
- [PL93] Noble S. Procter and Patrick J. Lynch. *Manual of Ornithology - Avian Structure and Function*. Yale University Press, London, 1993. 8, 10, 26
- [Sav95] D. Savic. Model of pattern formation in animal coatings. *Journal of Theoretical Biology*, 172:299–303, 1995. 36
- [SM90] L. J. Shaw and J. D. Murray. Analysis of a model of complex skin patterns. *SIAM Journal of Applied Mathematics*, pages 628–648, April 1990. 36, 77
- [Tho61] D’Arcy Wentworth Thompson. *On Growth and Form*. Cambridge at the University Press, New York, 1961. 36
- [Tur52] A. M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London B*, 237:37–72, 1952. 36
- [Ula62] S. Ulam. On some mathematical properties connected with patterns of growth of figures. *In Proceedings of Symposia on Applied Mathematics*, 14:215–224, 62. 37
- [Voi66] A. A. Voitkevich. *The Feather and Plumage of Birds*. October House Inc., New York, 1966. 12, 20
- [vTB76] Josselyn van Tyne and Andrew J. Berger. *Fundamentals of Ornithology*. John Wiley and Sons Inc., New York, 1976. 14, 17, 20, 21
- [WB88] J.C Welty and Luis Baptista. *The Life of Birds*. Saunder College Publishing, New York, 1988. 3, 21, 23, 24
- [WO90] M. Weliky and G. Oster. The mechanical basis of cell rearrangement. *Development*, 109:373–386, 1990. 36
- [WS83] T. Witten and L. Sander. Diffusion-limited aggregation. *Physics Rev. B*, 27:5686–5697, 1983. 36
- [You84] D. A. Young. A local activator-inhibitor model of vertebrate skin patterns. *Mathematical Biosciences*, 72:51–58, 1984. 37

Computer Science References

- [AK84] M. Aono and T. L. Kunii. Botanical tree image generation. *IEEE Computer Graphics & Applications*, 4(5):10–34, May 1984. 38
- [AK88] James Arvo and David Kirk. Modelling plants with environment-sensitive automata. *In Proceedings of Ausgraph*, 1988. 38
- [AM95] Markoto Ando and S. Morishima. Expression and motion control of hair using fast collision detection methods. *In Image Analysis Applications and Computer Graphics*. Springer Verlag, 1995. 70
- [APS00] Michael Ashikhmin, Simon Premoze, and Peter Shirley. A microfacet-based brdf generator. *Proceedings of SIGGRAPH 2000*, pages 65–74, July 2000. ISBN 1-58113-208-5. 35
- [Ara97] Mur Arad. Isometric texture mapping for free-form surfaces. *Computer Graphics Forum*, 16(5):247–256, 1997. ISSN 1067-7055. 32
- [Bid95] H. B. Bidasaria. A new method for modelling of hair and grass type textures. *ACM-SIGMOD International Conference on Management of Data*, pages 109–113, 1995. 40
- [BKRS86] E. A. Bier and Jr. K. R. Sloan. Two part texture mappings. *IEEE Computer Graphics & Applications*, 6(9):40–53, September 1986. 32
- [Bli78] James F. Blinn. Simulation of wrinkled surfaces. *Computer Graphics (Proceedings of SIGGRAPH 78)*, 12(3):286–292, August 1978. Held in Atlanta, Georgia. 31
- [BM93] Barry G. Becker and Nelson L. Max. Smooth transitions between bump rendering algorithms. *Proceedings of SIGGRAPH 93*, pages 183–190, August 1993. ISBN 0-201-58889-7. Held in Anaheim, California. 31, 34
- [BN76] J. F. Blinn and M. E. Newell. Texture and reflection in computer generated images. *Communications of the ACM*, 19:542–546, 1976. 30
- [Bon97] Jeremy S. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. *Proceedings of SIGGRAPH 97*, pages 361–368, August 1997. ISBN 0-89791-896-7. Held in Los Angeles, California. 32
- [BR97] G. V. G. Baranoski and J. G. Rokne. An algorithmic reflectance and transmittance model for plant tissue. *Computer Graphics Forum*, 16(3):141–150, August 1997. ISSN 1067-7055. 35
- [Bru98] Armin Bruderlin. A method to generate wet and broken-up animal fur. *Proceedings of the seventh Pacific Conference on Computer Graphics and Applications*, 1998. 41, 48, 113
- [BVIG91] Chakib Bennis, Jean-Marc Vézien, Gérard Iglésias, and André Gagalowicz. Piecewise surface flattening for non-distorted texture mapping. *Computer Graphics (Proceedings of SIGGRAPH 91)*, 25(4):237–246, July 1991. ISBN 0-201-56291-X. Held in Las Vegas, Nevada. 32

- [Cal96] Patrick Callet. Pertinent data for modelling pigmented materials in realistic rendering. *Computer Graphics Forum*, 15(2):119–128, 1996. ISSN 0167-7055. 34
- [CMS87] Brian Cabral, Nelson Max, and Rebecca Springmeyer. Bidirectional reflection functions from surface bump maps. *Computer Graphics (Proceedings of SIGGRAPH 87)*, 21(4):273–281, July 1987. Held in Anaheim, California. 34
- [Coo84] Robert L. Cook. Shade trees. *Computer Graphics (Proceedings of SIGGRAPH 84)*, 18(3):223–231, July 1984. Held in Minneapolis, Minnesota. 31
- [CR95] Peter Chambers and Alyn Rockwood. Visualization of solid reaction-diffusion systems. *IEEE Computer Graphics and Applications*, pages 7–11, 1995. 38
- [Cro84] Franklin C. Crow. Summed-area tables for texture mapping. *Computer Graphics (Proceedings of SIGGRAPH 84)*, 18(3):207–212, July 1984. Held in Minneapolis, Minnesota. 32
- [CT82] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics*, 1(1):7–24, January 1982. 33
- [CXGS02] Y. Chen, Y. Xu, B. Guo, and H. Shum. Modelling and rendering of realistic feathers. In *Computer Graphics (Proceedings of SIGGRAPH 2002)*, pages 630–636, 2002. 44, 45, 46, 50, 60, 69, 70, 72, 79, 87, 125, 135, 137, 142, 143
- [dBvKOS98] M. de Berg, M. van Kreveld, M. Overmars, and O. Scharwarzkopf. *Computational Geometry - Algorithms and Applications*. Springer-Verlag, New York, 1998. 80
- [DEL⁺99] Julie Dorsey, Alan Edelman, Justin Legakis, Henrik Wann Jensen, and Hans K ohling Pedersen. Modeling and rendering of weathered stone. *Proceedings of SIGGRAPH 99*, pages 225–234, August 1999. ISBN 0-20148-560-5. Held in Los Angeles, California. 32
- [DG94] J.-M. Dischler and D. Ghazanfarpour. A geometrical based method for highly complex structured textures generation. *Computer Graphics Forum*, 14(4):203–216, October 1994. 32, 52, 98
- [DHL⁺98] Oliver Deussen, Patrick Hanrahan, Bernd Lintermann, Radom r Mech, Matt Pharr, and Przemyslaw Prusinkiewicz. Realistic modeling and rendering of plant ecosystems. *Proceedings of SIGGRAPH 98*, pages 275–286, July 1998. ISBN 0-89791-999-8. Held in Orlando, Florida. 3, 39, 47, 110
- [dREF⁺88] 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):151–158, August 1988. Held in Atlanta, Georgia. 38
- [DSC95] Wen-Kai Dai, Zen-Chung Shih, and Ruei-Chuan Chang. Synthesizing feather textures in galliformes. *Computer Graphics Forum*, 14(3):407–420, August 1995. ISSN 1067-7055. 42, 60, 61, 70, 125
- [DTKT93] 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. Held in Oxford, UK. 2, 40, 41, 48, 70, 86, 87, 113
- [EAB⁺02] R. Engle, E. Armstrong, R. Bredow, J. Chen, R. Peitzman, and J Redd. Stuart Little 2: Let the feathers fly. *Siggraph Course #32*, 2002. 44, 50, 70, 126, 137, 142
- [Ede98] Herbert Edelsbrunner. Shape reconstruction with Delaunay complex. *Lecture Notes in Computer Science*, 1380:119–132, 1998. 81
- [EL99] A. A. Efros and T. K. Leung. A framework for interactive texture mapping. *International Conference on Computer Vision*, 2(Sept):1033–1038, 1999. 32

- [FFC82] A. Fournier, D. Fussell, and L. Carpenter. Computer rendering of stochastic models. *Communications of the ACM*, 25(6):371–384, June 1982. 4
- [FLCB95] Kurt Fleischer, David Laidlaw, Bena Currin, and Alan Barr. Cellular texture generation. *Proceedings of SIGGRAPH 95*, pages 239–248, August 1995. ISBN 0-201-84776-0. Held in Los Angeles, California. 32, 39
- [FMP92] Deborah R. Fowler, Hans Meinhardt, and Przemyslaw Prusinkiewicz. Modeling seashells. *Computer Graphics (Proceedings of SIGGRAPH 92)*, 26(2):379–388, July 1992. ISBN 0-201-51585-7. Held in Chicago, Illinois. 32, 38
- [Fou89] Alain Fournier. The modelling of natural phenomena. *Graphics Interface '89*, pages 191–202, June 1989. 4
- [FPB92] Deborah Fowler, Przemyslaw Prusinkiewicz, and Johannes Battjes. A collision-based model of spiral phyllotaxis. *SIGGRAPH '92*, 26(2):361–368, 1992. 39
- [FPdB90a] F. David Fracchia, Przemyslaw Prusinkiewicz, and Martin J. M. de Boer. Animation of the development of multicellular structures. *Computer Animation '90 (Second workshop on Computer Animation)*, pages 3–19, April 1990. 38
- [FPdB90b] F. David Fracchia, Przemyslaw Prusinkiewicz, and Martin J. M. de Boer. Visualization of the development of multicellular structures. *Graphics Interface '90*, pages 267–277, May 1990. 38
- [FR86] Alain Fournier and William T. Reeves. A simple model of ocean waves. *Computer Graphics (Proceedings of SIGGRAPH 86)*, 20(4):75–84, August 1986. Held in Dallas, Texas. 29
- [Fra95] F. D. Fracchia. Integrating lineage and interaction for the visualization of cellular structures. *5th International Workshop on Graph Grammars and their Application to Computer Science*, 1995. 39
- [FW01] Cristiano Franco and Marcelo Walter. Modeling the structure of feathers. In *SIBGRAPI - XIV Brazilian Symposium on Computer Graphics and Image Processing*, 2001. 44
- [GGS03] C. Gotsman, X. Gu, and A. Sheffer. Fundamentals of spherical parameterization for 3d meshes. In *ACM Transactions on Graphics Proceedings of SIGGRAPH 2003*, pages 358–363, 2003. 84, 133
- [Gla95] Andrew S. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann Publishers Inc., San Fransico, 1995. 33, 34, 41
- [Gol97] Dan B. Goldman. Fake fur rendering. *Proceedings of SIGGRAPH 97*, pages 127–134, August 1997. ISBN 0-89791-896-7. Held in Los Angeles, California. 5, 41, 70, 77
- [Gre89] 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. Held in Boston, Massachusetts. 38
- [Gre91] Ned Greene. Detailing tree skeletons with voxel automata. *SIGGRAPH '91 Course Notes on Photorealistic Volume Modelling and Rendering Techniques*, 1991. 38
- [GRS96] Eduard Gröller, René T. Rau, and Wolfgang Straßer. Modeling textiles as three dimensional textures. *Eurographics Rendering Workshop 1996*, pages 205–214, June 1996. ISBN 3-211-82883-4. Held in Porto, Portugal. 31
- [GW92] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, New York, 1992. 121

- [GW97] Allen Van Gelder and Jane Wilhelms. An interactive fur modeling technique. *Graphics Interface '97*, pages 181–188, May 1997. ISBN 0-9695338-6-1 ISSN 0713-5424. 5, 41, 61, 70, 77, 86
- [HB95] David J. Heeger and James R. Bergen. Pyramid-based texture analysis/synthesis. *Proceedings of SIGGRAPH 95*, pages 229–238, August 1995. ISBN 0-201-84776-0. Held in Los Angeles, California. 32
- [HB00] Donald House and David Breen. *Cloth Modeling and Animation*. A K Peters, Natick, Massachusetts, 2000. 87
- [HDKS00] Wolfgang Heidrich, Katja Daubert, Jan Kautz, and Hans-Peter Seidel. Illuminating micro geometry based on precomputed visibility. *Proceedings of SIGGRAPH 2000*, pages 455–464, July 2000. ISBN 1-58113-208-5. 35
- [Hec86] Paul S. Heckbert. Survey of texture mapping. *IEEE Computer Graphics & Applications*, 6(11):56–67, November 1986. 31
- [HH90] Pat Hanrahan and Paul E. Haeberli. Direct wysiwyg painting and texturing on 3d shapes. *Computer Graphics (Proceedings of SIGGRAPH 90)*, 24(4):215–223, August 1990. ISBN 0-201-50933-4. Held in Dallas, Texas. 32
- [HK93] Pat Hanrahan and Wolfgang Krueger. Reflection from layered surfaces due to subsurface scattering. *Proceedings of SIGGRAPH 93*, pages 165–174, August 1993. ISBN 0-201-58889-7. Held in Anaheim, California. 34, 48
- [HM92] Chet S. Haase and Gary W. Meyer. Modeling pigmented materials for realistic image synthesis. *ACM Transactions on Graphics*, 11(4):305–335, October 1992. ISSN 0730-0301. 34, 48
- [HMT00] Sunil Hadap and Nadia Magnenat-Thalmann. Interactive hair styler based on fluid flow. *Eurographics Computer Animation and Simulation Workshop*, 2000. 41, 48, 85, 86, 87, 113
- [Hol94] M. Holton. Strands, gravity, and botanical tree imagery. *Computer Graphics Forum*, 13(1):57–67, January 1994. 38
- [HP96] Mark Hammel and Przemyslaw Prusinkiewicz. Visualization of developmental processes by extrusion in space-time. *Graphics Interface '96*, pages 246–258, May 1996. ISBN 0-9695338-5-3. 38
- [HRD95] Mark Hammel, William Remphrey, and Campbell Davidson. Simulating the development of *fraxinus pennsylvanica* shoots using l-systems. *Proceedings of the Sixth Western Computer Graphics Symposium*, pages 49–58, 1995. Held in Banff, Alberta 20-22 March 1995. 38
- [HS99] Wolfgang Heidrich and Hans-Peter Seidel. Realistic, hardware-accelerated shading and lighting. *Proceedings of SIGGRAPH 99*, pages 171–178, August 1999. ISBN 0-20148-560-5. Held in Los Angeles, California. 101
- [HTSG91] Xiao D. He, Kenneth E. Torrance, François X. Sillion, and Donald P. Greenberg. A comprehensive physical model for light reflection. *Computer Graphics (Proceedings of SIGGRAPH 91)*, 25(4):175–186, July 1991. ISBN 0-201-56291-X. Held in Las Vegas, Nevada. 34
- [iAUK92] 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. ISBN 0-201-51585-7. Held in Chicago, Illinois. 2, 5, 40, 41, 70, 77
- [IKVZ00] 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. 41, 48, 113

- [Kaa92] J. Kaandorp. *Modelling growth forms of biological objects using fractals*. PhD Thesis, University of Amsterdam, Amsterdam, 1992. 38
- [Kaj85] James T. Kajiya. Anisotropic reflection models. *Computer Graphics (Proceedings of SIGGRAPH 85)*, 19(3):15–21, July 1985. Held in San Francisco, California. 34
- [KH00] C.K. Koh and Z. Huang. Real-time animation of human hair modeled in strips. *Eurographics Computer Animation and Simulation Workshop*, 2000. 41, 87, 113
- [KK89] James T. Kajiya and Timothy L. Kay. Rendering fur with three dimensional textures. *Computer Graphics (Proceedings of SIGGRAPH 89)*, 23(3):271–280, July 1989. Held in Boston, Massachusetts. 31, 32, 34, 40, 41, 77
- [KMN⁺99] Michael A. Kowalski, Lee Markosian, J. D. Northrup, Lubomir Bourdev, Ronen Barzel, Loring S. Holden, and John Hughes. Art-based rendering of fur, grass, and trees. *Proceedings of SIGGRAPH 99*, pages 433–438, August 1999. ISBN 0-20148-560-5. Held in Los Angeles, California. 42
- [KN00] Waiming Kong and Masayuki Nakajima. Hair rendering by jittering and pseudo shadow. *Computer Graphics International 2000*, June 2000. Held in Geneva, Switzerland. 42
- [KS00] Jan Kautz and Hans-Peter Seidel. Towards interactive bump mapping with anisotropic shift-variant brdfs. *2000 SIGGRAPH / Eurographics Workshop on Graphics Hardware*, pages 51–58, August 2000. 101
- [Lar92] Gregory J. Ward Larson. Measuring and modeling anisotropic reflection. *Computer Graphics (Proceedings of SIGGRAPH 92)*, 26(2):265–272, July 1992. ISBN 0-201-51585-7. Held in Chicago, Illinois. 33, 34
- [LD96] Bernd Lintermann and Oliver Deussen. Interactive modelling and animation of branching botanical structures. *Eurographics Computer Animation and Simulation Workshop*, 1996. 39, 48
- [Len00] Jerome Lengyel. Real-time hair. *Rendering Techniques 2000: 11th Eurographics Workshop on Rendering*, pages 243–256, June 2000. ISBN 3-211-83535-0. 42
- [Lew89] John-Peter Lewis. Algorithms for solid noise synthesis. *Computer Graphics (Proceedings of SIGGRAPH 89)*, 23(3):263–270, July 1989. Held in Boston, Massachusetts. 31
- [LL00] Greg Leibon and David Letscher. Delaunay triangulations and voronoi diagrams for riemannian manifolds. In *Proceedings of the 16th Annual Symposium on Computational Geometry (SCG-00)*, pages 341–349, N. Y., June 12–14 2000. ACM Press. 81
- [Llo82] S. Lloyd. Least square quantization in PCM. *IEEE Transactions on Information Theory*, 28:129–137, 1982. 79
- [LM94] Peter Litwinowicz and Gavin Miller. Efficient techniques for interactive texture placement. *Proceedings of SIGGRAPH 94*, pages 119–122, July 1994. ISBN 0-89791-667-0. Held in Orlando, Florida. 32
- [LM98] Bruno Lévy and Jean-Laurent Mallet. Non-distorted texture mapping for sheared triangulated meshes. *Proceedings of SIGGRAPH 98*, pages 343–352, July 1998. ISBN 0-89791-999-8. Held in Orlando, Florida. 32
- [LR79] A. Lindenmayer and G. Rosenberg. Parallel generation of maps: Developmental systems for cell layers. *Lecture Notes in Computer Science; Graph grammars and their application to computer science; First International Workshop*, 73:301–316, 1979. 37

- [LTT91] Andre LeBlanc, Rusell Turner, and Daniel Thalmann. Rendering hair using pixel blending shadow buffers. *The Journal of Visualization and Computer Animation*, 2(3):92–97, 1991. 41, 42, 70
- [LV00] Tom Lokovic and Eric Veach. Deep shadow maps. *Proceedings of SIGGRAPH 2000*, pages 385–392, July 2000. ISBN 1-58113-208-5. 42
- [Mil88] Gavin S. P. Miller. From wire-frames to furry animals. *Graphics Interface '88*, pages 138–145, June 1988. 40, 70
- [MJC03] S. Marschner, H. Jensen, and M. Cammarano. Light scattering from human hair fibers. In *ACM Transactions on Graphics Proceedings of SIGGRAPH 2003*, pages 780–791, 2003. 35, 42, 72, 102
- [MKKI00] Xiaoyang Mao, Makoto Kikukawa, Kouichi Kashio, and Atsumi Imamiya. Automatic generation of hair texture with line integral convolution. *Proceedings of the IEEE International Conference on Information Visualization*, 2000. 40
- [MN00] A. Meyer and F. Neyret. Multiscale shaders for the efficient realistic rendering of pine-trees. In *Graphics Interface*, pages 137–144, May 2000. 99
- [MPBM03] Wojciech Matusik, Hanspeter Pfister, Matt Brown, and Leonard McMillan. A data-driven reflectance model. In *ACM Transactions on Graphics Proceedings of SIGGRAPH 2003*, pages 759–769, 2003. 35
- [MYV93] Jérôme Maillot, Hussein Yahia, and Anne Verroust. Interactive texture mapping. *Proceedings of SIGGRAPH 93*, pages 27–34, August 1993. ISBN 0-201-58889-7. Held in Anaheim, California. 32
- [Ney95a] Fabrice Neyret. Animated texels. *Computer Animation and Simulation '95*, pages 97–103, September 1995. ISBN 3-211-82738-2. 31
- [Ney95b] Fabrice Neyret. A general and multiscale model for volumetric textures. *Graphics Interface '95*, pages 83–91, May 1995. ISBN 0-9695338-4-5. 31
- [Ney96] Fabrice Neyret. Synthesizing verdant landscapes using volumetric textures. *Eurographics Rendering Workshop 1996*, pages 215–224, June 1996. ISBN 3-211-82883-4. Held in Porto, Portugal. 31
- [Ney98] Fabrice Neyret. Modeling, animating, and rendering complex scenes using volumetric textures. *IEEE Transactions on Visualization and Computer Graphics*, 4(1):55–70, January-March 1998. ISSN 1077-2626. 31
- [Nis98] Tomoyuki Nishita. Light scattering models for the realistic rendering of natural scenes. *Eurographics Rendering Workshop 1998*, pages 1–10, June 1998. ISBN 3-211-83213-0. Held in Vienna, Austria. 29, 34
- [PB00] Dan Piponi and George D. Borshukov. Seamless texture mapping of subdivision surfaces by model pelting and texture blending. *Proceedings of SIGGRAPH 2000*, pages 471–478, July 2000. ISBN 1-58113-208-5. 32
- [Pea85] Darwyn R. Peachey. Solid texturing of complex surfaces. *Computer Graphics (Proceedings of SIGGRAPH 85)*, 19(3):279–286, July 1985. Held in San Francisco, California. 31, 32
- [Ped96] Hans K hling Pedersen. A framework for interactive texturing operations on curved surfaces. *Proceedings of SIGGRAPH 96*, pages 295–302, August 1996. ISBN 0-201-94800-1. Held in New Orleans, Louisiana. 32
- [Per85] Ken Perlin. An image synthesizer. *Computer Graphics (Proceedings of SIGGRAPH 85)*, 19(3):287–296, July 1985. Held in San Francisco, California. 31, 32

- [PF90] Pierre Poulin and Alain Fournier. A model for anisotropic reflection. *Computer Graphics (Proceedings of SIGGRAPH 90)*, 24(4):273–282, August 1990. ISBN 0-201-50933-4. Held in Dallas, Texas. 31, 34, 95, 99, 102
- [PFH00] Emil Praun, Adam Finkelstein, and Hugues Hoppe. Lapped textures. *Proceedings of SIGGRAPH 2000*, pages 465–470, July 2000. ISBN 1-58113-208-5. 32
- [PH89] Ken Perlin and Eric M. Hoffert. Hypertexture. *Computer Graphics (Proceedings of SIGGRAPH 89)*, 23(3):253–262, July 1989. Held in Boston, Massachusetts. 31, 40
- [PH00] Matt Pharr and Pat Hanrahan. Monte carlo evaluation of non-linear scattering equations for subsurface reflection. *Proceedings of SIGGRAPH 2000*, pages 75–84, July 2000. ISBN 1-58113-208-5. 35
- [PHM93] Przemyslaw Prusinkiewicz, Mark S. Hammel, and Eric Mjolsness. Animation of plant development. *Proceedings of SIGGRAPH 93*, pages 351–360, August 1993. ISBN 0-201-58889-7. Held in Anaheim, California. 38
- [PJM94] Przemyslaw Prusinkiewicz, Mark James, and Radomír Mech. Synthetic topiary. *Proceedings of SIGGRAPH 94*, pages 351–358, July 1994. ISBN 0-89791-667-0. Held in Orlando, Florida. 38
- [PL90] Przemyslaw Prusinkiewicz and Aristid Lindenmayer. *The Algorithmic Beauty of Plants*. Springer-Verlag, 1990. 3, 29, 38, 59, 60, 70
- [PL98] R. Paget and I. Longstaff. Texture synthesis via a noncausal nonparametric multiscale markov random field. *IEEE Transactions on Image Processing*, 7(6):925–931, 1998. 32
- [PLH88] Przemyslaw Prusinkiewicz, Aristid Lindenmayer, and James Hanan. Developmental models of herbaceous plants for computer imagery purposes. *Computer Graphics (Proceedings of SIGGRAPH 88)*, 22(4):141–150, August 1988. Held in Atlanta, Georgia. 38
- [Pru93] Przemyslaw Prusinkiewicz. Modelling and visualization of biological structures. *Graphics Interface '93*, pages 128–137, May 1993. Held in Toronto, Ontario, Canada. 36, 37, 38
- [RB85] 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. Held in San Francisco, California. 38, 40
- [RC96] Alyn Rockwood and Peter Chambers. *Interactive Curves and Surfaces*. Morgan Kaufmann Publishers Inc., California, 1996. 64
- [RCT91] 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. 2, 40, 41, 70, 77, 113
- [Ree83] William T. Reeves. Particle systems - a technique for modeling a class of fuzzy objects. *Computer Graphics (Proceedings of SIGGRAPH 83)*, 17(3):359–376, July 1983. Held in Detroit, Michigan. 29, 70
- [RH96] J. Rankin and R. Hall. A simple naturalistic hair model. *Computer Graphics*, 30(1):5 – 9, Jan. 1996. 70
- [Rob01] Barbara Robertson. Birds of a feather. *Computer Graphics World*, pages 22–27, January 2001. 43, 135
- [RW99] Balajee Ramakrishnananda and Kok Cheong Wong. Animating bird flight using aerodynamics. *The Visual Computer*, 15(10):494–508, 1999. ISSN 0178-2789. 7

- [Sch91] Andreas Schilling. A new simple and efficient anti-aliasing with subpixel masks. *Computer Graphics (Proceedings of SIGGRAPH 91)*, 25(4):133–141, July 1991. ISBN 0-201-56291-X. Held in Las Vegas, Nevada. 32
- [Sch94] Christophe Schlick. A survey of shading and reflectance models. *Computer Graphics Forum*, 13(2):121–131, June 1994. 33
- [Sec02] A. Secord. Weighted voronoi stippling. In *2nd International Symposium on Non-Photorealistic Animation and Rendering*, 2002. 83
- [SF95] Jos Stam and Eugene Fiume. Depicting fire and other gaseous phenomena using diffusion processes. *Proceedings of SIGGRAPH 95*, pages 129–136, August 1995. ISBN 0-201-84776-0. Held in Los Angeles, California. 29
- [SFDC00] Yinlong Sun, F. David Fracchia, Mark S. Drew, and Thomas W. Calvert. Rendering iridescent colors of optical disks. *Rendering Techniques 2000: 11th Eurographics Workshop on Rendering*, pages 341–352, June 2000. ISBN 3-211-83535-0. 34
- [SG94] Zen-Chung Shih and Hurng-Dar Guo. The modeling and animation of human hair. In *Pacific Graphics*, pages 215–228, Beijing China, 1994. 70
- [SGM97] 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. 35, 48
- [SH02] L. Streit and W. Heidrich. A biologically-parameterized feather model. *Computer Graphics Forum (Eurographics 2002)*, 21(3):565–573, 2002. 84
- [Ski89] Robert Skinner. Modeling hair with structured particle systems. *Master's Thesis*, June 1989. 70
- [Smi84] Alvy Ray Smith. Plants, fractals and formal languages. *Computer Graphics (Proceedings of SIGGRAPH 84)*, 18(3):1–10, July 1984. Held in Minneapolis, Minnesota. 35, 38
- [Sta99] Jos Stam. Diffraction shaders. *Proceedings of SIGGRAPH 99*, pages 101–110, August 1999. ISBN 0-20148-560-5. Held in Los Angeles, California. 34
- [Str90] Paul S. Strauss. A realistic lighting model for computer animators. *IEEE Computer Graphics & Applications*, 10(6):56–64, November 1990. 33
- [SZH97] Detlev Stalling, Malte Zöckler, and Hans-Christian Hege. Fast Display of Illuminated Field Lines. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):118–128, April 1997. 101
- [Tay92] C. E. Taylor. Fleshing out. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial Life II*, pages 25–38. Addison-Wesley, 1992. 36
- [TM87] Tommaso Toffoli and Norman Margolous. *Cellular Automata Machines: A new environment for modelling*. Cambridge at the University Press, Cambridge, 1987. 37
- [TOS⁺03] Norimichi Tsumura, Nobutoshi Ojima, Kayoko Sato, Mitsuhiro Shiraishi, Hideto Shimizu, Hirohide Nabeshima, Syuuichi Akazaki, Kimihiko Hori, and Yoichi Miyake. Image-based skin color and texture analysis/synthesis by extracting hemoglobin and melanin information in the skin. In *ACM Transactions on Graphics Proceedings of SIGGRAPH 2003*, pages 770–779, 2003. 35
- [Tur91] Greg Turk. Generating textures for arbitrary surfaces using reaction-diffusion. *Computer Graphics (Proceedings of SIGGRAPH 91)*, 25(4):289–298, July 1991. ISBN 0-201-56291-X. Held in Las Vegas, Nevada. 32, 38, 78

- [Tur92] Greg Turk. Re-tiling polygonal surfaces. *Computer Graphics (Proceedings of SIGGRAPH 92)*, 26(2):55–64, July 1992. ISBN 0-201-51585-7. Held in Chicago, Illinois. 47, 78, 135
- [WAT92] Stephen H. Westin, James R. Arvo, and Kenneth E. Torrance. Predicting reflectance functions from complex surfaces. *Computer Graphics (Proceedings of SIGGRAPH 92)*, 26(2):255–264, July 1992. ISBN 0-201-51585-7. Held in Chicago, Illinois. 34
- [WF97] Marcelo Walter and Alain Fournier. Growing and animating polygonal models of animals. *Computer Graphics Forum*, 16(3):151–158, August 1997. ISSN 1067-7055. 32
- [WFR98] Marcelo Walter, Alain Fournier, and Mark Reimers. Clonal mosaic model for the synthesis of mammalian coat patterns. *Graphics Interface '98*, pages 82–91, June 1998. ISBN 0-9695338-6-1. 32, 38
- [WH94] Andrew P. Witkin and Paul S. Heckbert. Using particles to sample and control implicit surfaces. *Proceedings of SIGGRAPH 94*, pages 269–278, July 1994. ISBN 0-89791-667-0. Held in Orlando, Florida. 44
- [Wil83] Lance Williams. Pyramidal parametrics. *Computer Graphics (Proceedings of SIGGRAPH 83)*, 17(3):1–11, July 1983. Held in Detroit, Michigan. 32
- [WK91] Andrew Witkin and Michael Kass. Reaction-diffusion textures. *Computer Graphics (Proceedings of SIGGRAPH 91)*, 25(4):299–308, July 1991. ISBN 0-201-56291-X. Held in Las Vegas, Nevada. 32
- [WL00] Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. *Proceedings of SIGGRAPH 2000*, pages 479–488, July 2000. ISBN 1-58113-208-5. 32
- [WNH97] Tien-Tsin Wong, Wai-Yin Ng, and Pheng-Ann Heng. A geometry dependent texture generation framework for simulating surface imperfections. *Eurographics Rendering Workshop 1997*, pages 139–150, June 1997. ISBN 3-211-83001-4. Held in St. Etienne, France. 32, 98
- [Wor96] Steven P. Worley. A cellular texture basis function. *Proceedings of SIGGRAPH 96*, pages 291–294, August 1996. ISBN 0-201-94800-1. Held in New Orleans, Louisiana. 32
- [WP95] Jason Weber and Joseph Penn. Creation and rendering of realistic trees. *Proceedings of SIGGRAPH 95*, pages 119–128, August 1995. ISBN 0-201-84776-0. Held in Los Angeles, California. 39
- [WS92] Yasuhiko Watanabe and Yasuhito Suenaga. A trigonal prism-based method for hair image generation. *IEEE Computer Graphics & Applications*, 12(1):47–53, January 1992. 2, 5, 40, 41, 48, 70, 77, 86, 87, 113

Index

- alpha blending, 98
- anterior, 10
- apteria, 9
- Bézier curves, 63–64, 92, 96
- barbicels, 12
- barbule
 - pennaceous, 12
 - plumulaceous, 12
 - rachidal, 12
 - stylet, 12
 - substructure, 12–13
- biochromes, 21–22
- biological models, 35–39
- birdieness, 43
- blastema, 25
- BRDF, 31
- BTF, 46
- caudal, 10
- cephalad, 10
- conclusions, 145–149
- definitive down, 14
- Delaunay Triangulation, 80–81
- distal, 10
- dorsal, 10
- endogenous, 36
- epithelial, 12
- exogenous, 36
- feather
 - direction, 86–87
 - models, 42–47
 - arrangement, 111–114
 - barb
 - elevation, 89–90
 - colour and pattern, 21–24, 72
 - colour changes, 23–24
 - counts, 21
 - curvature, 15
 - development, 18
 - direction, 20
 - distribution, 16
 - contour feathers, 17
 - non-contour feathers, 16–17
 - growth, 25–26, 69, 105–107
 - interpolation, 62–69, 84
 - moult, 26–27, 69
 - orientation, 86–87
 - parameterization, 59–62
 - pigment location, 24
 - regeneration, 24–27, 69, 107–111
 - rendering, 44, 95, 103
 - shaft
 - rotation, 88–89
 - shape, 15
 - sheath, 12
 - size, 21
 - structure, 13
 - substructure
 - afterfeather, 11, 13
 - barbicels, 12–13
 - barbs, 12
 - barbules, 12–13
 - calamus, 12
 - rachis, 12
 - vane, 12
 - tract, 9, 18–20
 - types, 66–69
 - bristles, 11, 15
 - contour, 11–13
 - down, 11, 14
 - ear coverts, 14
 - filoplumes, 11, 14
 - flight, 11, 14
 - plumulaceous, 11
 - powder down, 14

- powderdown, 11
- semplume, 11, 14
- feather coat, 9
 - arrangement, 75
 - morphogenesis, 39
 - morphogenesis, 39
 - regeneration, 26–27
 - rendering, 103
- follicle distribution, 77–84
- frontal, 10
- fur models, 40–42
- fur orientation, 86
- future work, 148–149

- Galliformes, 42
- geometry, 70–72

- hair models, 40–42
- hair orientation, 86

- inferior umbilicus, 12

- keratin, 25

- L-systems, 38, 45, 142
 - context sensitive, 38
 - differential, 38
 - IL-systems, 37
 - OL-systems, 37
 - parametric, 38
- level of detail, 30–35, 51–52, 98–102, 141–142
- lineage, 36
- Lloyd's Algorithm, 79–82

- mathematical models, 35–37
- medial, 10
- midline, 10
- monopodial, 60
- morphogenesis, 9
- morphogens, 36
- moult, 9

- natal down, 14
- NPR, 43

- packing algorithm, 85–92
- pennaceous, 12
- peripheral, 10

- pigments, 21–22
- pitch, 10
- pith, 12
- plant models, 39–40
- plumage, 9
- plumulaceous, 12
- posterior, 10
- problem
 - assumptions, 51
 - components, 54–56, 117–137
 - contribution, 56, 143
 - integration, 138–141
 - scope and goals, 49–54
- proximal, 10
- pterlography, 9
- pterylae, 9

- reaction diffusion, 36
- rectrices, 14
- reflection models, 33–35
 - anisotropic, 98–101
 - analytical, 33–34
 - non-analytical, 34–35
- remiges, 14
- roll, 10
- rudiment, 25

- saggital, 10
- schemochromes, 22
- spatial models, 36–37
- structural models, 37
- superior umbilicus, 12

- terms
 - anatomical, 10
 - avian, 9–10
- texture
 - Hypertexture, 31
 - mapping, 30, 32, 96–97
 - maps, 31–32
 - texel, 31
 - texel mapping, 31
 - volumetric, 31
- transverse, 10

- ventral, 10

- wing shape, 20–21

yaw, 10