
JCFeather Doc

Author: JerryKon

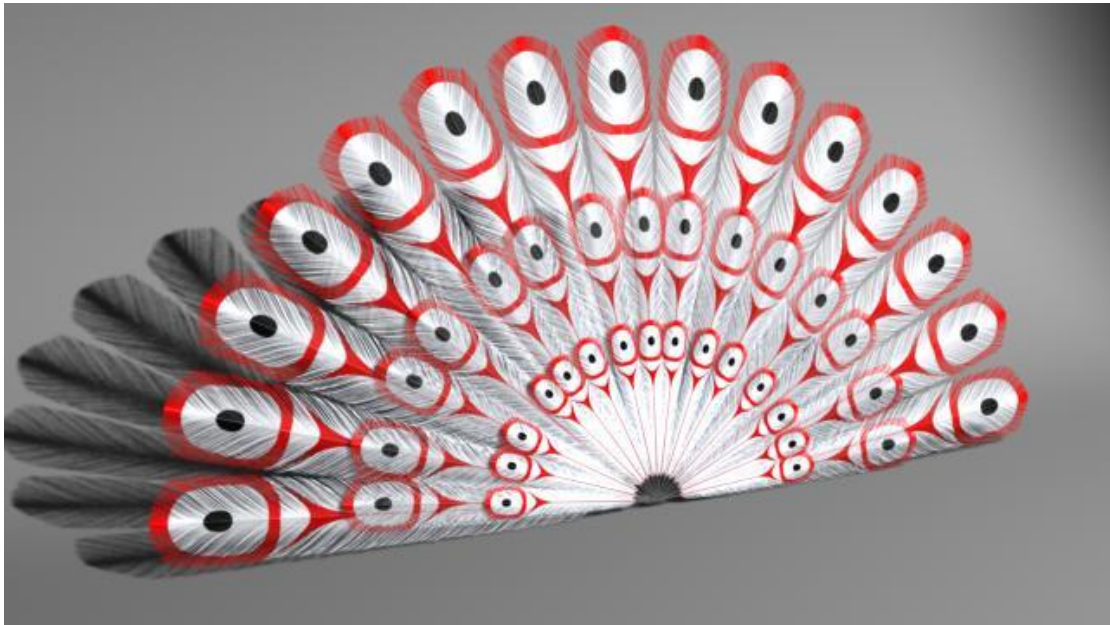
Website: <http://www.jerrykon.com>

Chapter 1 Introduction.....	
Chapter 2 Getting Started.....	
2.1 Surface Feather.....	
2.2 Vertices Feather.....	
2.3 Component Feather.....	
2.4 ComponentArray Feather.....	
2.5 Standard Feather.....	
2.6 Feather Library.....	
2.7 Feather Coat.....	
Chapter 3 Tools UI.....	
3.1 jcFeather Library.....	
3.2 jcFeather Tools.....	
3.3 jc Grow Feather.....	
3.4 Hair Gen From Geometry.....	
3.5 Create Multi jcComponent.....	
Chapter 4 Technique Detail In JCFeather.....	
3.1 The Feather World In JCFeather.....	
3.2 Behind The Barbules Shape.....	
3.3 Rendering JCFeather.....	
Chapter 5 Menus Reference.....	

Chapter 6 Nodes Reference.....	
Chapter 7 Commands Reference.....	
Chapter 8 JCFeather SDK.....	
Chapter 9 Current Limitations And Future Work.....	

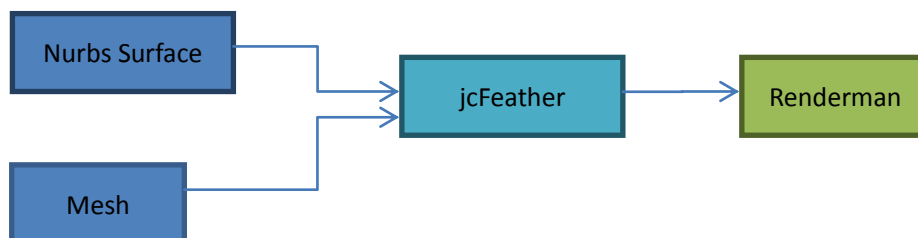
Chapter 1

Introduction



jcFeather is a plugin for Autodesk Maya. It is designed to make feathers for creatures such as birds, eagles. It includes shaping individual feather, creating feather coat and finally rendering them with renderman Ricurves.

Go to <http://www.jerrykon.com/jcFeather.html> for videos and more information.



The above graph is the basic workflow of jcFeather.

Free version jcFeather2.8.5:

With free version 2.8.5, you can create only 3 jcFeather nodes and 3 jcPreFeather nodes.

How to install jcFeather (purchase jcFeather):

1. Get getHostID.exe from <http://www.jerrykon.com/jcFeather.html>
2. Run getHostID.exe. If success, you will get a jcFeather.hid file in C:/(make sure you run getHostID.exe as administrator).
3. Send jcFeather.hid file to jerrychinese@163.com to get a license.
4. Buy jcFeather on <http://www.jerrykon.com/jcFeather.html> and get the license.
5. Click jcFeather.msi to install.
6. Put the license file jcFeather.lic in C:/flexlm/(if it doesn't exist, create one) .

License type:

License is locked to your machine and is not transferrable.

What's new in jcFeather 2.8.5:

- 1, Render feather with textured polygons in 3delight and Arnold.

What's new in jcFeather 2.8.0:

1, jcPreGuide , jcPreFeather and jcPreController have been redesigned. A feather coat is much easier to make. You can also paint the direction and size of feather instance mesh. Notice that those three nodes are redesigning, so they are not compatible with previous version.

2, You can export small feather cache data from Maya and use jcFeatherLib sdk to generate feathers in your c++ code. So you can develop geometry shader(for example, in mr) for your render. Sdk in Maya is also available, if you do not want to export jcFeather cache to disk files, you may like writing a new Maya plugin to get feather curves in memory and render them out.

3, A dso plugin was written for 3delight. Now you can render feathers in memory. Each jcFeather node is rendered in its proxy bounding box. So jcFeather renders more efficiently.

What's new in jcFeather 2.3.2:

- 1, Add jclnstanncerToMesh mel command, convert particle instance to a combined mesh.

- 2, Use new feather gap generation method. The old gap attributes are obsolete.
- 3, Fix the bug: render is stopped when no jcFeather is renderable.

What's new in jcFeather 2.3.0:

- 1, Convert jcFeather to curves.
- 2, Fix the bug: wrong barbule rotation direction in standard jcFeather type.

What's new in jcFeather 2.2:

- 1, Feather generation is now multi thread enabled.
- 2, Add standard jcFeather.
- 3, More smooth feather rachis curve.
- 4, Add and remove a few attributes.
- 5, Bugs fixes.

What's new in jcFeather 2.0:

1. Add jcPreGuid, jcPreFeather, jcPreController nodes to make a flexible and easily controllable feather mesh.
2. Export selected jcFeather with "-s 1" in command "jcWriteFeather".
3. Work with maya renderlayer.

Update in jcFeather 1.6.5:

1. Optimize memory usage when disconnecting attributes with jcFeather node.
2. The vertex group attribute in *jcFeatherComponent* and *jcFeatherComponentArray* node can be number decreased, e.g. 5:0 not just 0:5.
3. The key barbule display is not correct when *surface feather* attribute turned on.
4. Update shelf tool Create Multi jcFeatherComponent procedure.

What's new in jcFeather 1.6:

1. Use jcFeatherComponent and jcFeatherComponentArray to get mesh information for feather making. These two new nodes collect mesh info and output the data to jcFeather node to make jcFeather.
2. Remove mesh vertices set feather making.

What's new in jcFeather 1.5:

1. Add *proxyboundingbox* to jcFeather node for delaying read archive in renderman.
2. Add *uniformWidth* to jcFeather to export feather curves more efficiently.
3. Add u and v parameters to control feather based on body uv.
4. Add mainColor parameters to make each feather in jcFeather different from each other in color.
5. Add textureProcedure parameters to make each feather in jcFeather different in barbules color texture.
6. Convert jcFeather to mesh in *surface feather* mode.

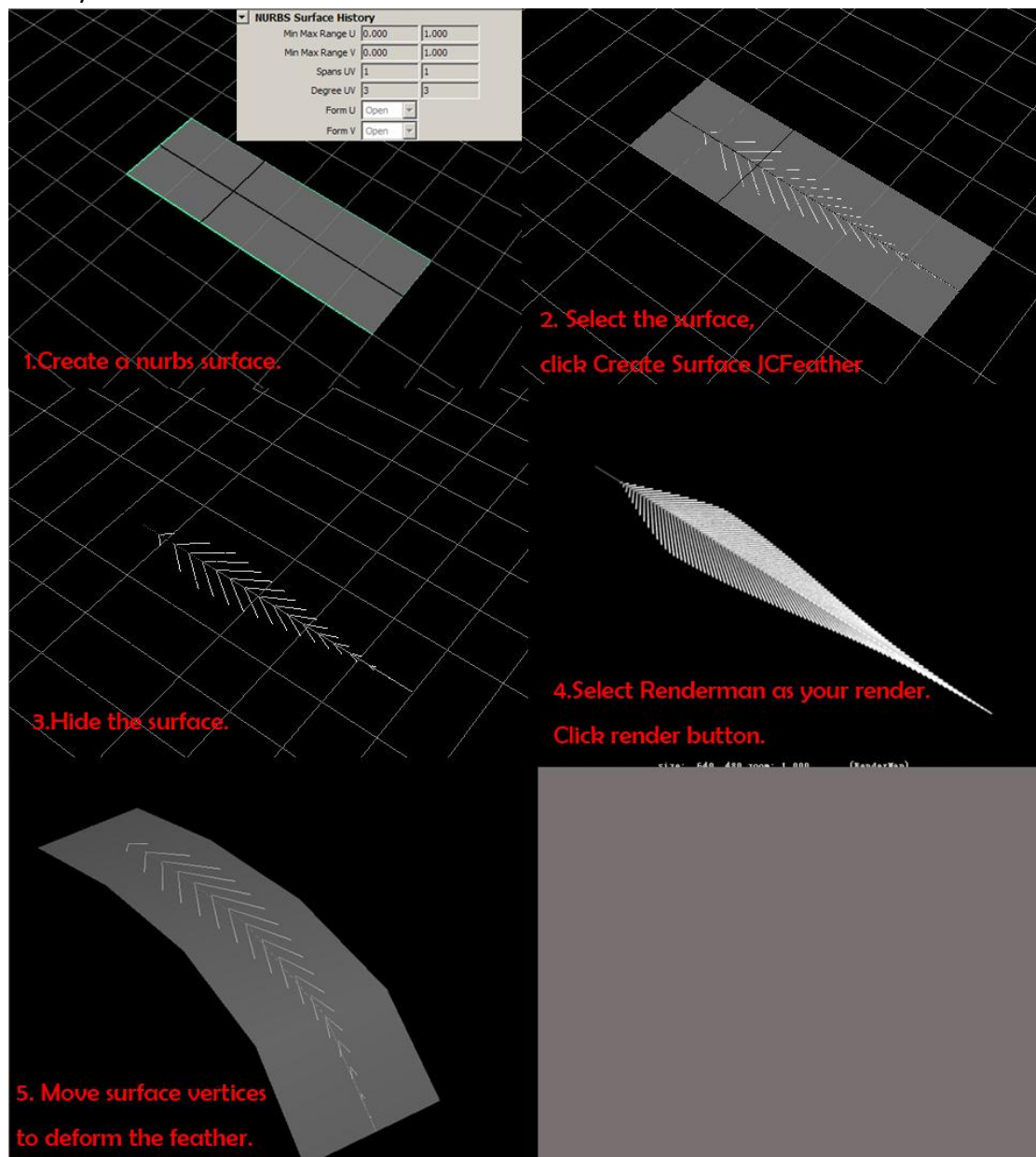
7. Optimize the interpolation algrithom in jcFeatherSystem. Add *power* parameters.
8. Ouput jcFeather to rib files with more efficient mel cmd.
9. Add *dealyReadArchive* to jcFeatherGlobals.
10. Add more motionblur parameters to jcFeatherGlobals to get better motion blur effects.

Chapter 2

Getting Started

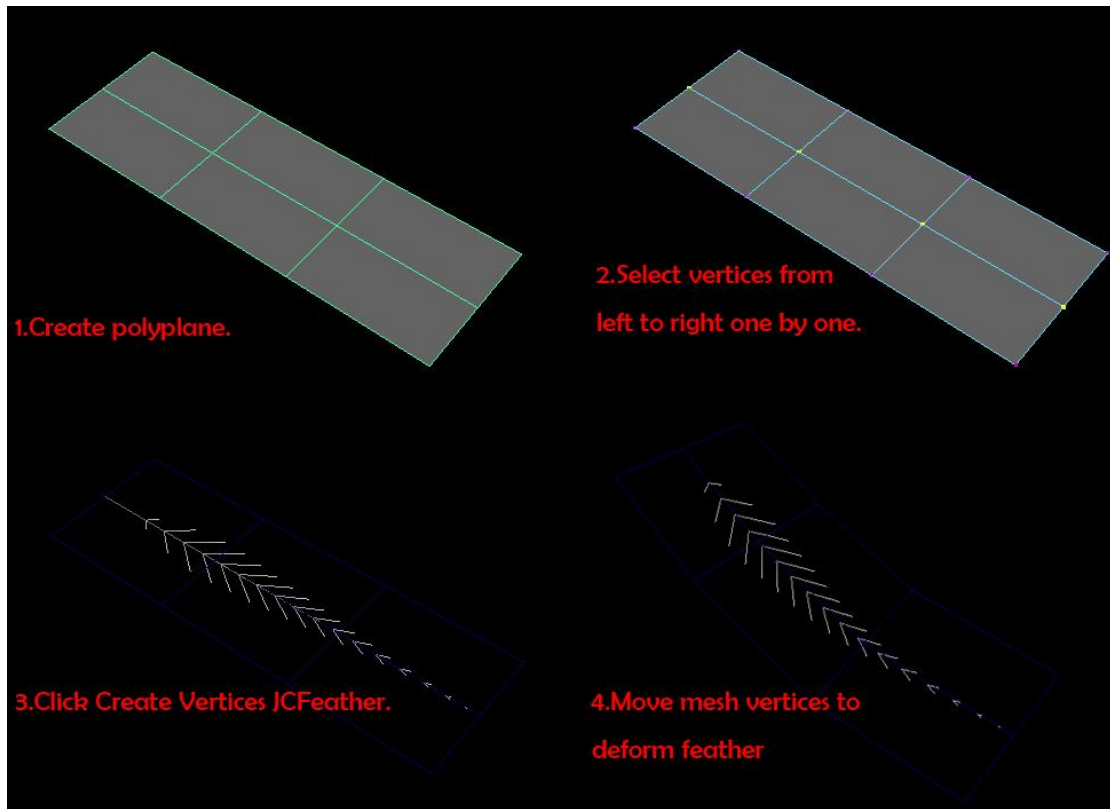
2.1 Surface Feather:

1. Create a nurbs surface and make sure its uv ranges 0 to 1.
2. Select the nurbs surface and click menu *Create Surface JCFeather*.
3. Hide the nurbs surface and select renderman as your render, then click render button and you will get the simplest feather.
4. To animate the feather, you can move the surface or animate the surface with tools in maya. For example, you can bind the surface to joints then add some keyframes.



2.2 Vertices Feather:

1. Create a poly plane mesh, set subdivision width 3, height 2.
2. Select vertices of the middle edges. Select the vertex one by one.
3. Click *Create Vertices JCFeather*. Then you get a feather.
4. To animate the feather, you can deform the poly mesh or move the mesh.

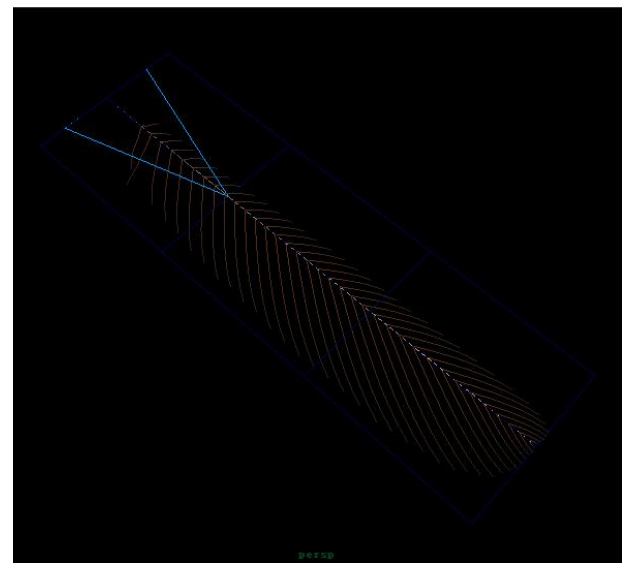
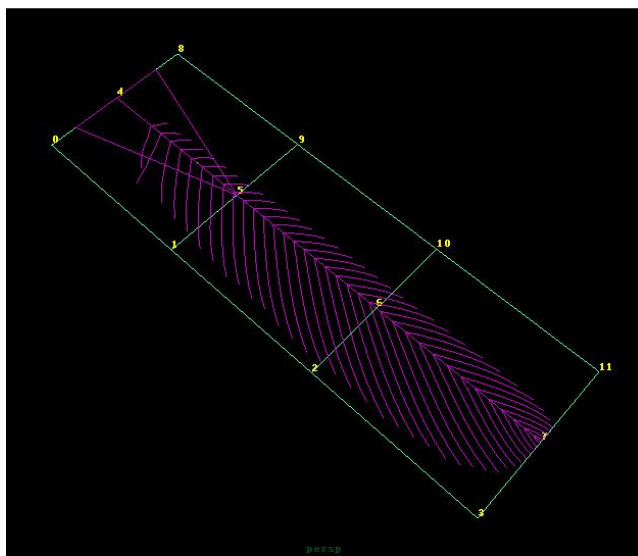


2.3 Component Feather:

Component Feather will use poly vertices to make feather. JCFeather use the node *jcFeatherComponent* to collect information from mesh. *jcFeatherComponent* uses the input vertices id to make a feather rachis template and output the data to *jcFeather* to make feather.

When you create Vertices Feather, a *jcFeatherComponent* node will be created automatically.

In the below images, the light blue locator is *jcFeatherComponent*. The node use vertices 4 5 6 7 as the rachis template points.

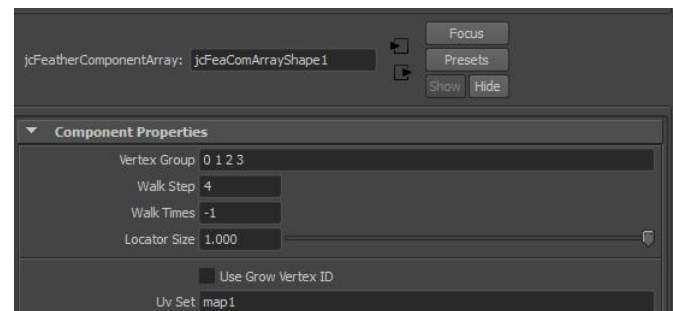
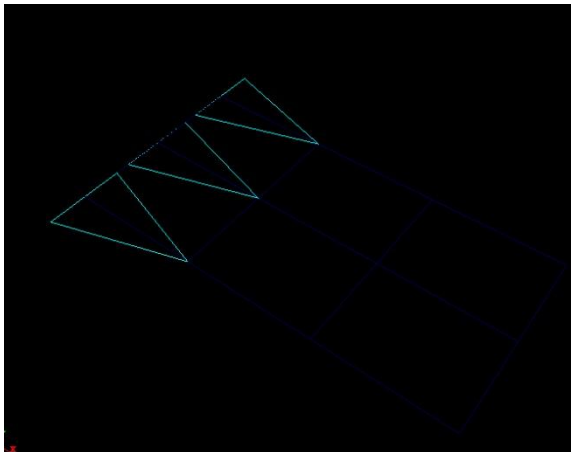


1. Create a poly plane mesh, set subdivision width 3, height 2.
2. To create a *jcFeatherComponent*, you need select mesh vertices one by one and then click menu *Create Vertices Component*. In the above image the vertices are 4 5 6 7.
3. Then you can select the *jcFeatherComponent* node in the outliner and then click menu *Create Component JCFeather* to create a *jcFeather*.

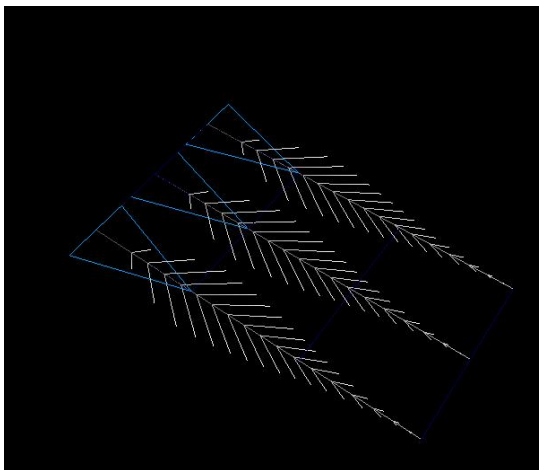
2.4 ComponentArray Feather:

ComponentArray Feather is only an array type of Component Feather. jcFeather uses *jcFeatherComponentArray* node to do such things. This node outputs multiple rachis templates to jcFeather.

1. Create a poly plane mesh, set subdivision width 3, height 2.
2. Select the mesh and click menu Create Mesh Component Array , then you will get a new node called *jcFeaComArray#* in outliner.
3. Select *jcFeaComArray#* node. Open the attribute editor, set VertexGroup attribute 0 1 2 3, set Walk Step 4. Then you will get this.



4. Select *jcFeaComArray#* node and click menu *Create Component Array Feather*. Then you will get 3 feathers like this.



Note: It is a good start to begin with the jcFeather Library. You can create a feather using any way above. Then select your jcFeather node and assign the preset feather shape in jcFeather Library. Get more details in 2.5.

2.5 Standard jcFeather

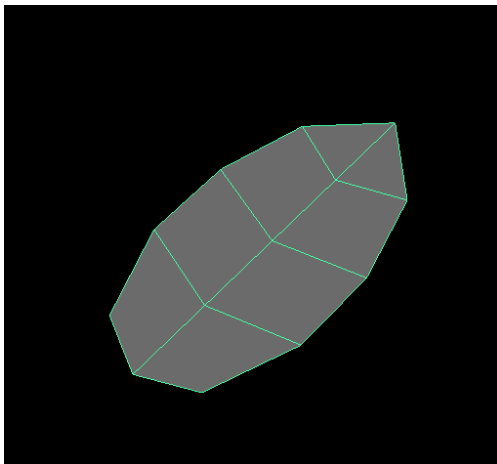
Standard jcFeather is supposed to be used with jcFeather template mesh. You can

create the template mesh in *JCFeather Library* by clicking icon

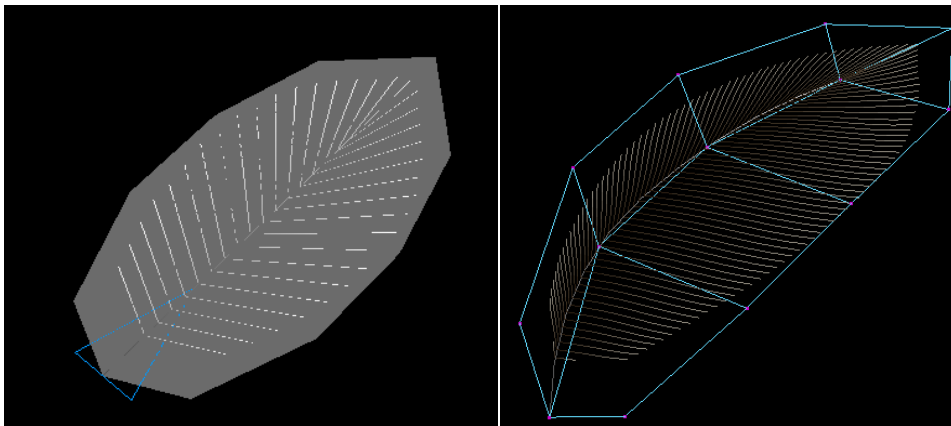


Example 1: Simple standard feather.

1, Select a template mesh.

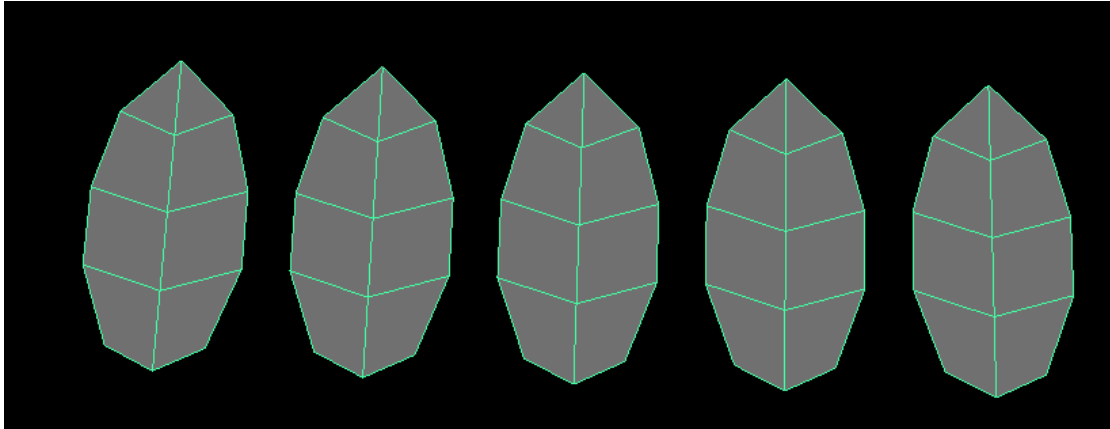


2, Click menu *Create Standard JCFeather* to create a standard mesh. And move mesh vertices to get your own feather shape.

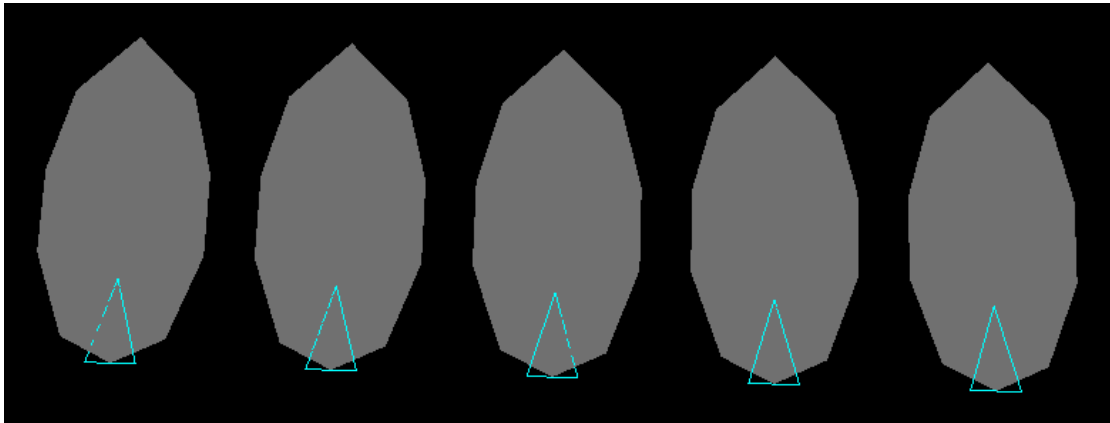


Example 2: Standard jcFeather Array

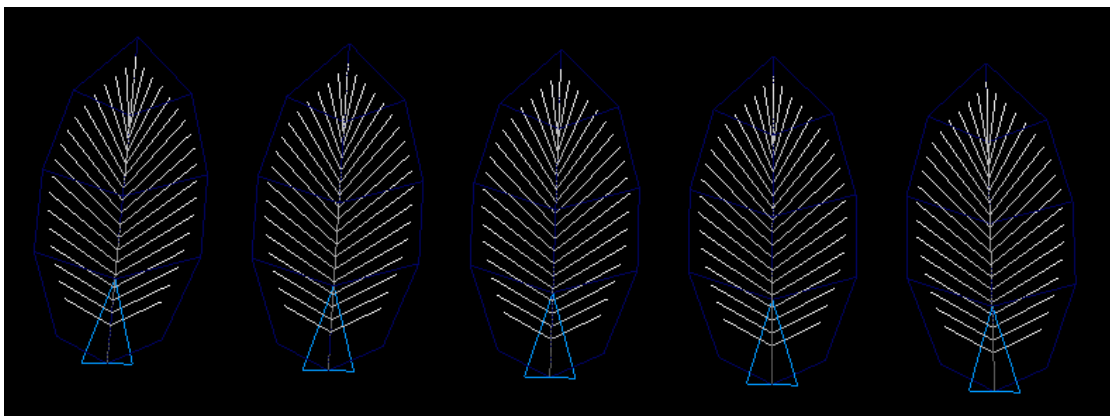
1, Create several template meshes, and combine them.



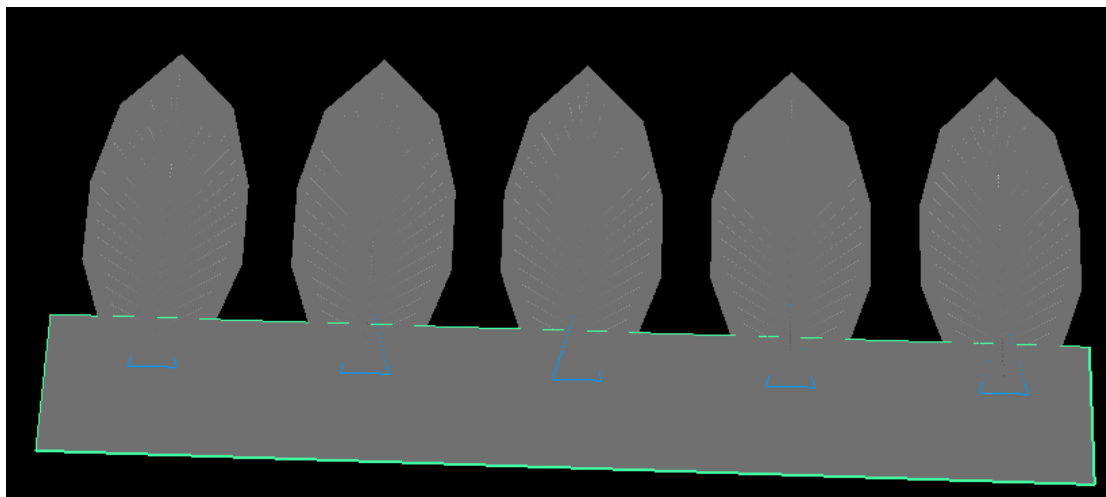
2, Select the new combined mesh, click *Create Standard Component Array*.



3, Select the new jcFeatherComponentArray node and click *Create Component Array JCFeather*.



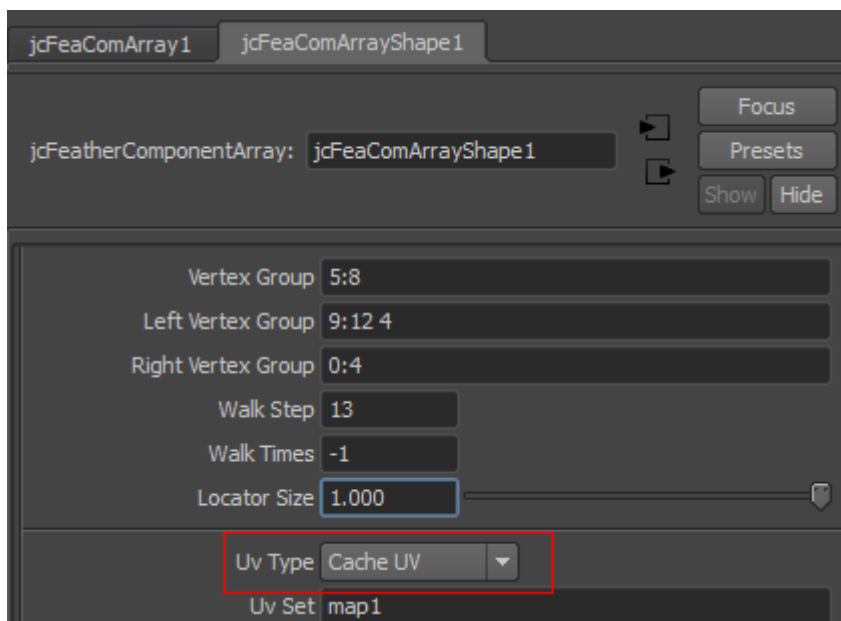
4, Create a new nurbsSurface at the root of feather like this.



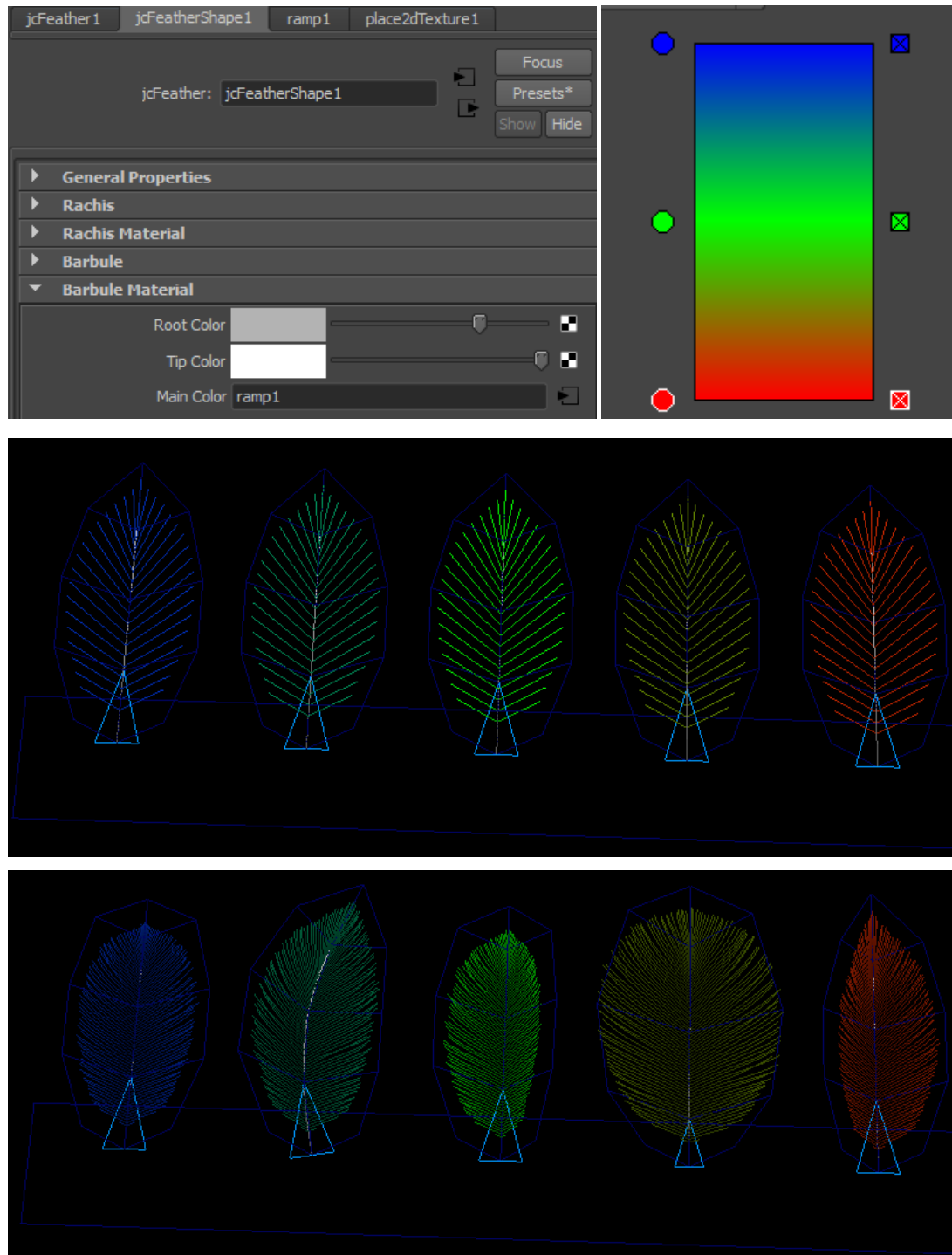
5, Now, we will transfer the uv value of the nurbssurface to the feather. Select the nurbssurface and the jcFeatherComponentArray node(triangle locator), click the



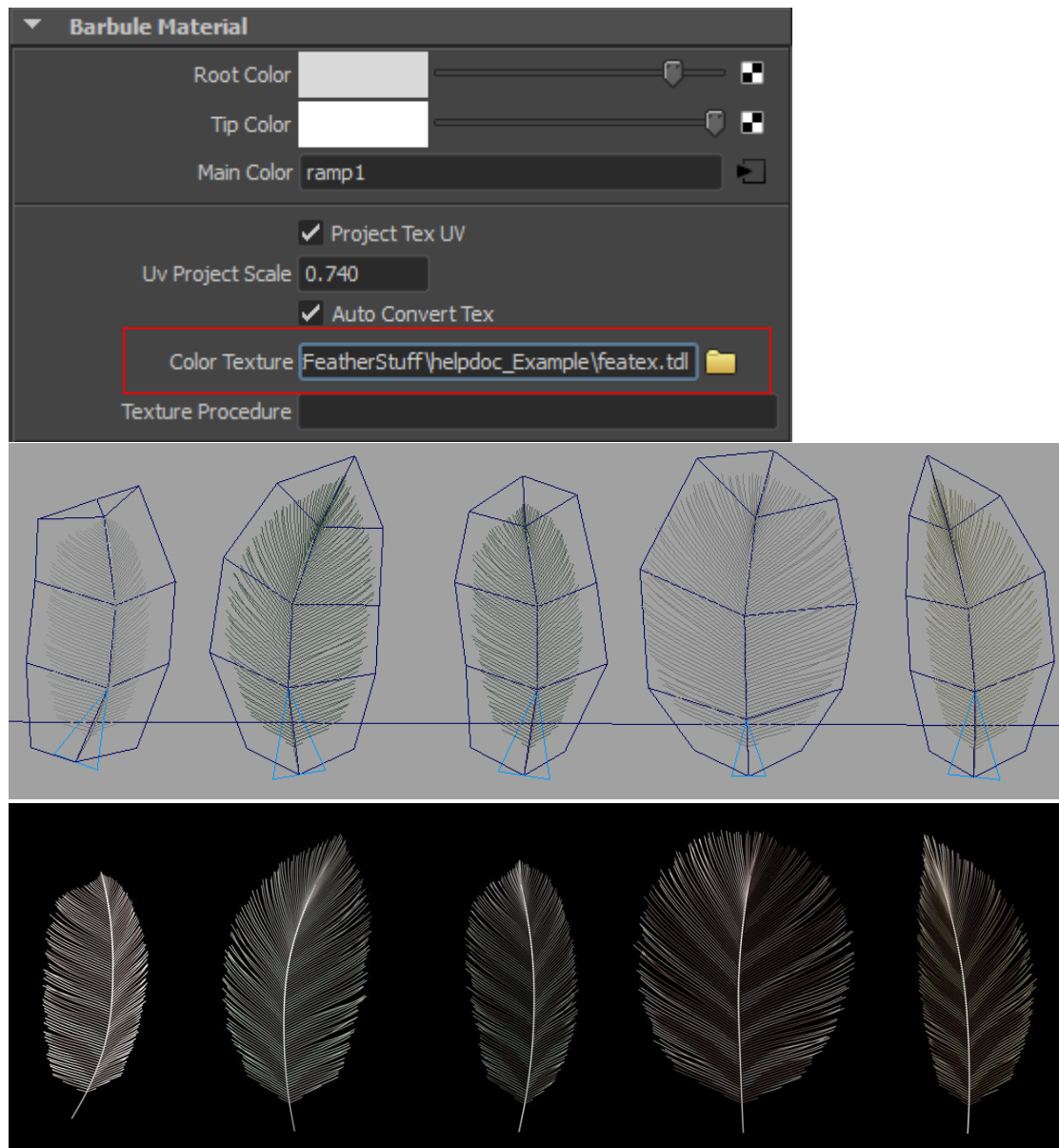
shelf icon on the *JCFeather Shelf*. Select the triangle locator change the attribute *UVType* to *CacheUV*.



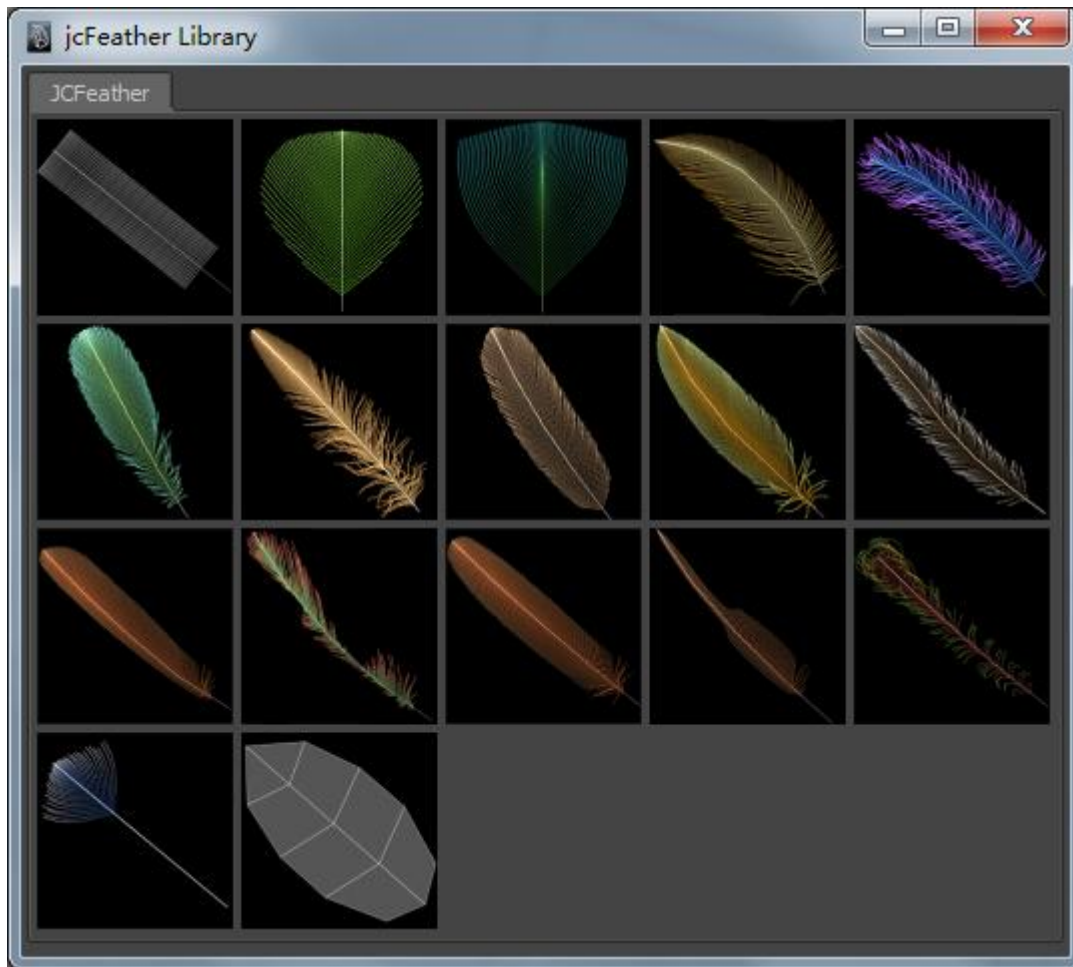
6, Select the jcFeather node, connect a ramp texture to the *main color* attribute. Then you can change the color of the ramp to make different colors for each feather.



7, Change the mesh shape and browse the texture featex.tdl to the feather texture attribute.



2.6 Feather Library:



Once you've installed jcFeather. It's recommended to Create JCFeather Shelf just by clicking the menu Create JCFeather Shelf at the bottom of JCFeather menu.

Click the jcFeather Library shelf button on the JCFeather shelf to open the jcFeather Library UI. Select a jcFeather node and then click the picture button in the ui to assign the preset feather shape.



The last icon is used to import a standard jcFeather template mesh.

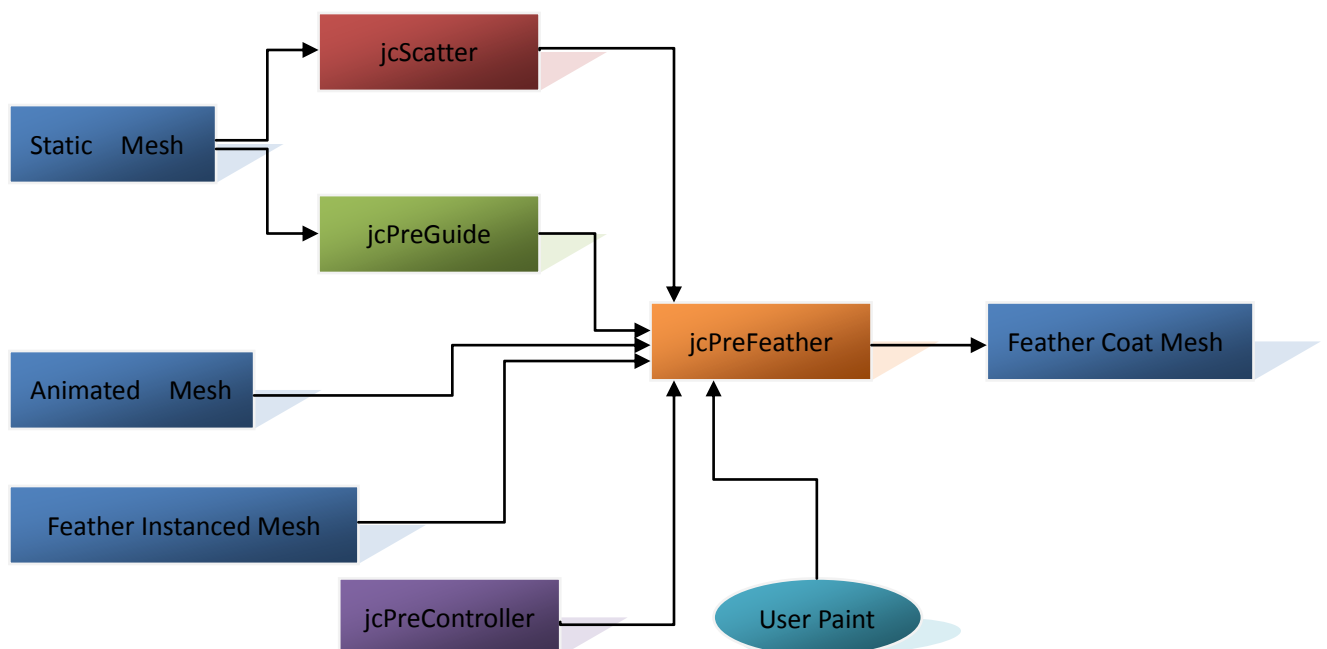
Note: The feather shape in this library is not compatible with standard feather.

Because the shape of standard jcFeather is affected by both the jcFeather attribute and the template mesh topology. So you need a little more work to get the right result.

2.7 Feather Coat:

There are two methods to make feather coat with jcFeather plugin.

1. You can create lots of nurbs surfaces or meshes , scatter them on the body of the bird and create jcFeather on each single surface or mesh. Finally you will get a feather coat.
2. Use jcScatter, jcPreGuide, jcPreFeather to make a feather mesh coat and use jcPreController as rigging controller. Here is the workflow of feather coat making with this method.

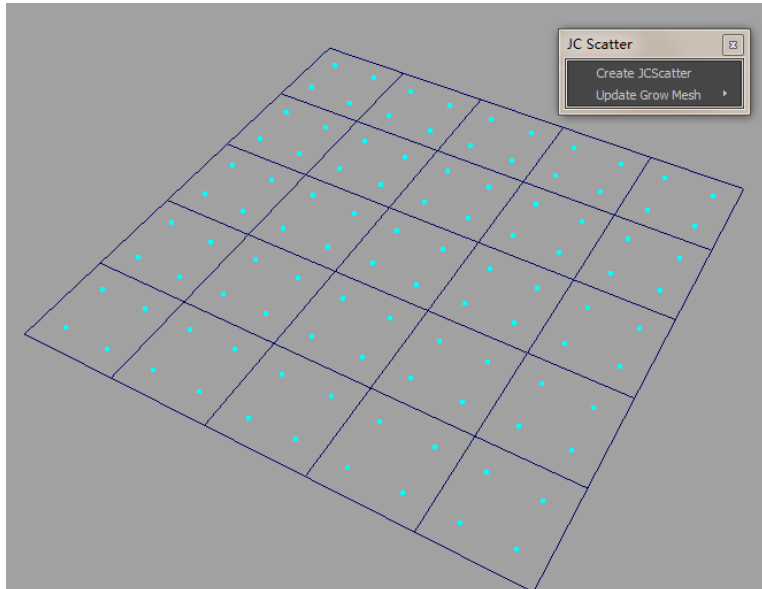


Description:

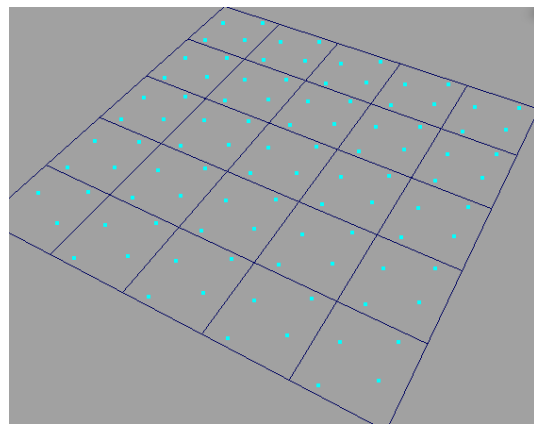
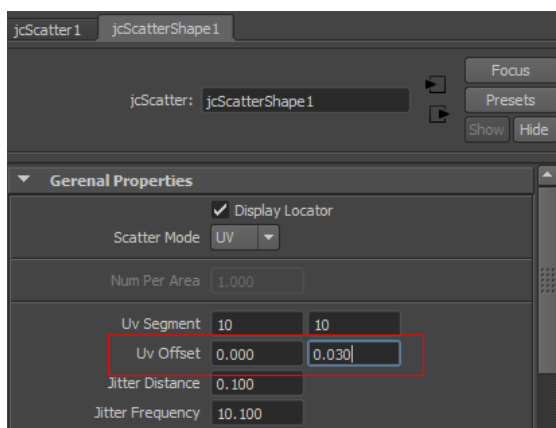
Static Mesh and **Animated Mesh** must have the same topology. **jcScatter** and **jcPreGuide** use the **Static Mesh** to compute static data for **jcPreFeather**. **jcPreFeather** instance lots of **Feather Template Mesh** on **Animated Mesh** to output a **Feather Coat Mesh**. If the size and direction of feather is not good, you can use **User Paint** to paint the direction and size. **jcPreController** is a rigging controller to interactively control the rotation and size of feather. A **jcPreFeather** may have lots of **jcPreControllers**. Each **jcPreController** has a weight map for its own control range. You can render the **Feather Coat Mesh** or use it to make jcFeather.

Example : How to make a feather coat

- 1, Make a polyPlane and select the mesh, click menu Create JcScatter.

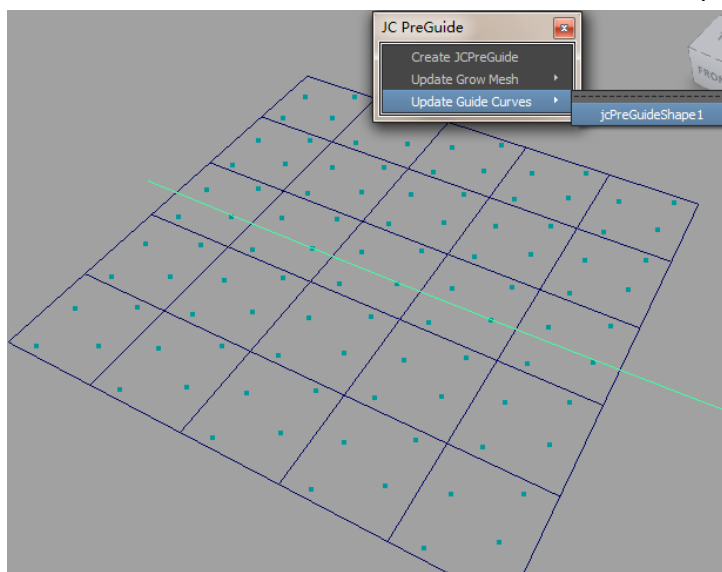


2, Select jcScatter1 node set the attribute as following. The grow point position will be like this.

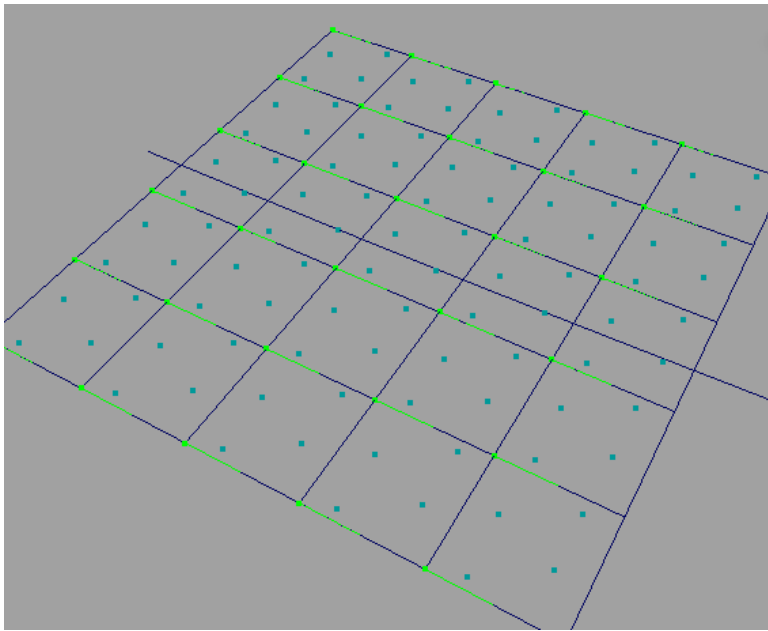


3, Select the polyPlane mesh and click Create JCPreGuide menu.

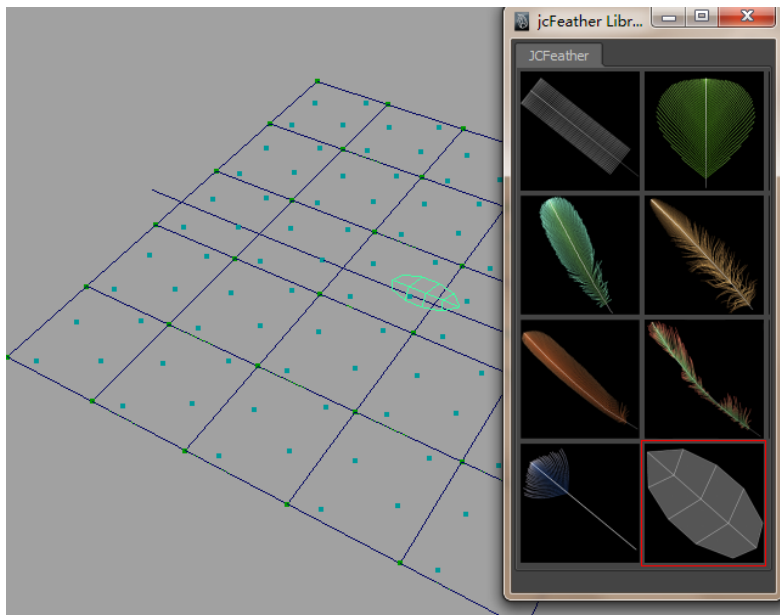
4, Create a nurbs curve like this. And click the menu Update Guide Curves.



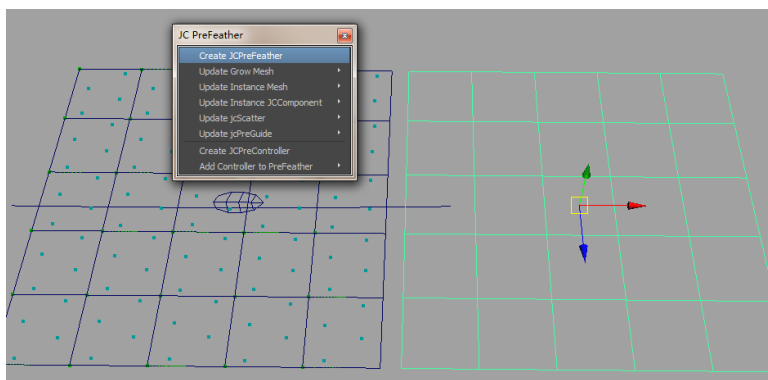
4, The green lines locator is the jcPreGuid locator.



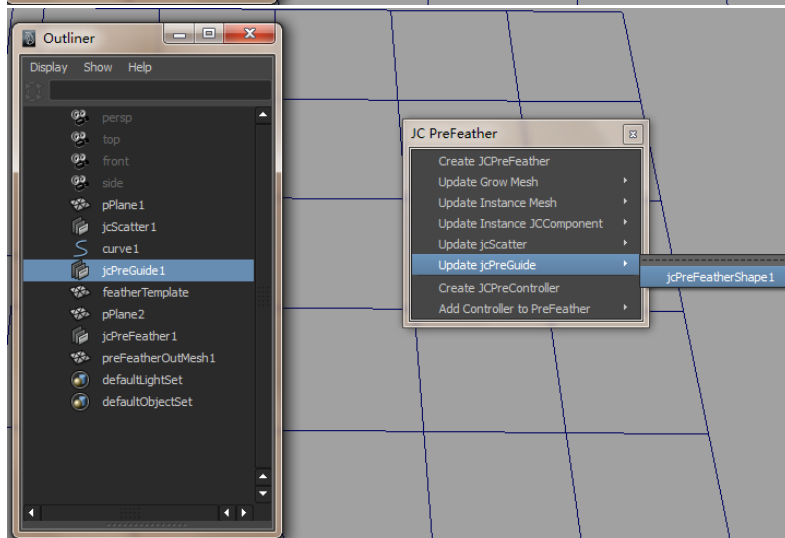
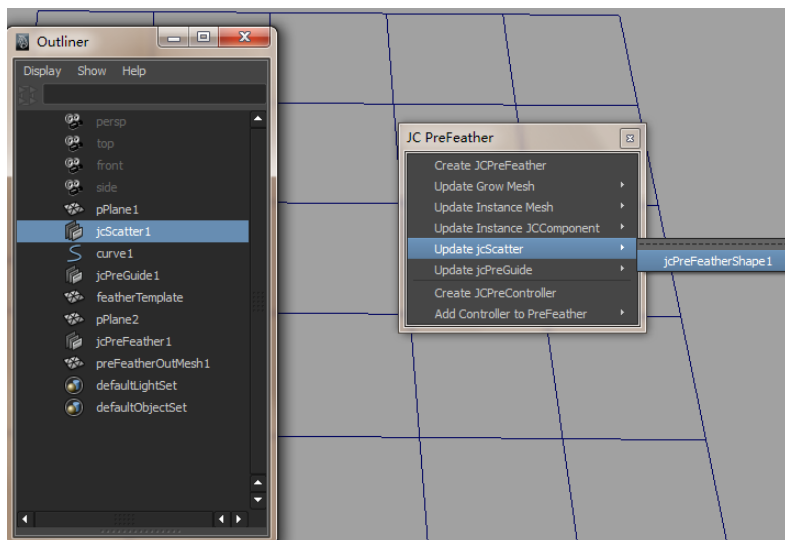
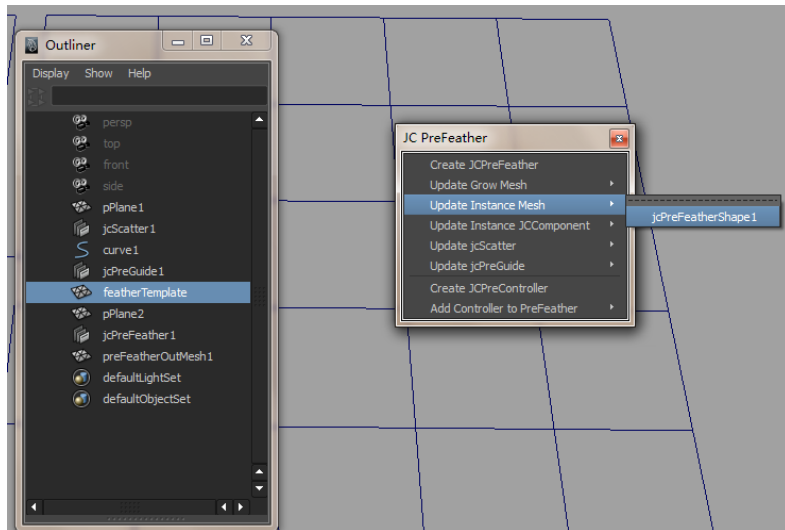
5, Use jcFeather Library to import a template feather mesh.



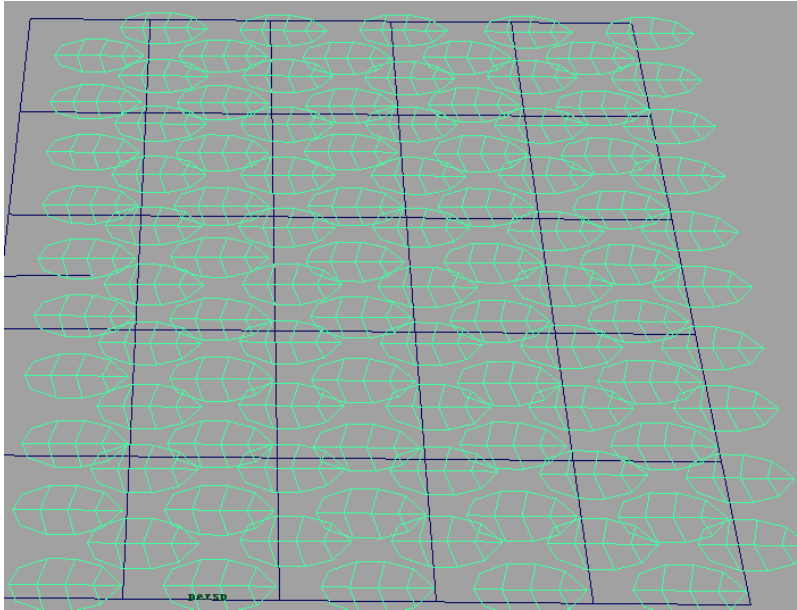
6, Duplicate the polyPlane and move it to a new location. Select pPlane2, click menu Create JCPreFeather.



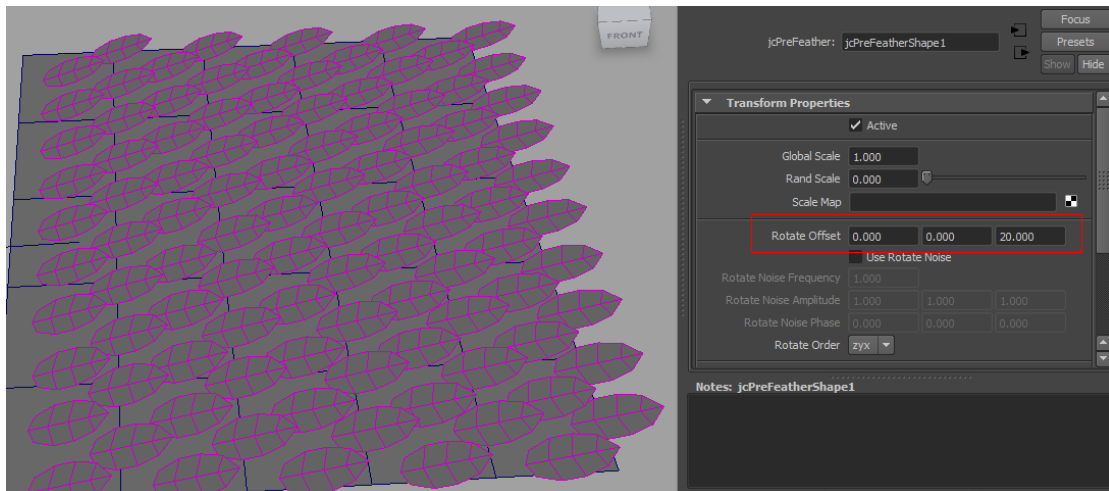
7, Select featherTemplate mesh click menu Update Instance Mesh. Select jcScatter1, click Update jcScatter. Select jcPreGuide1, click Update jcPreGuide.



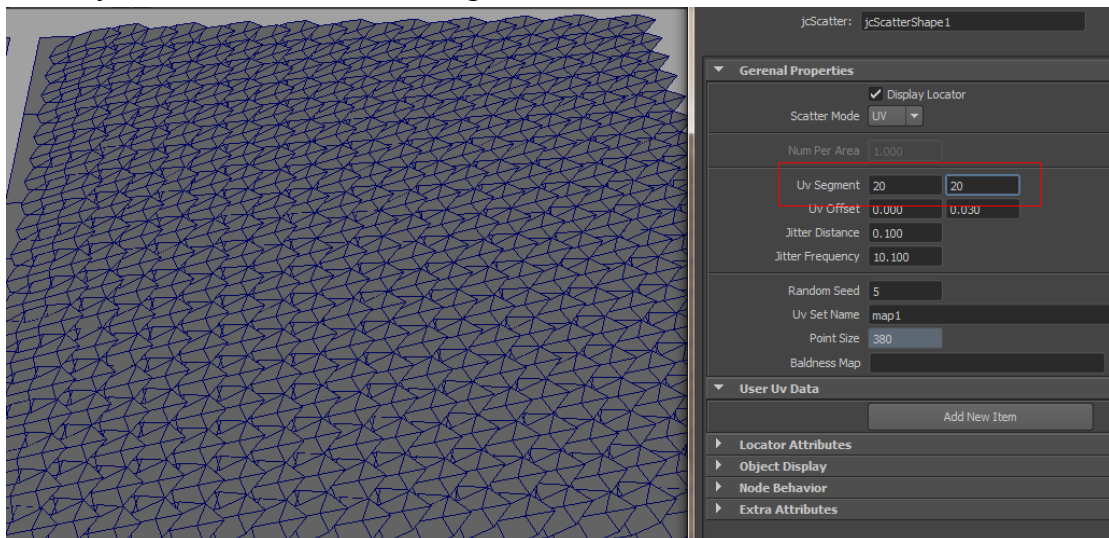
8, Finally, you will get a feather coat mesh.




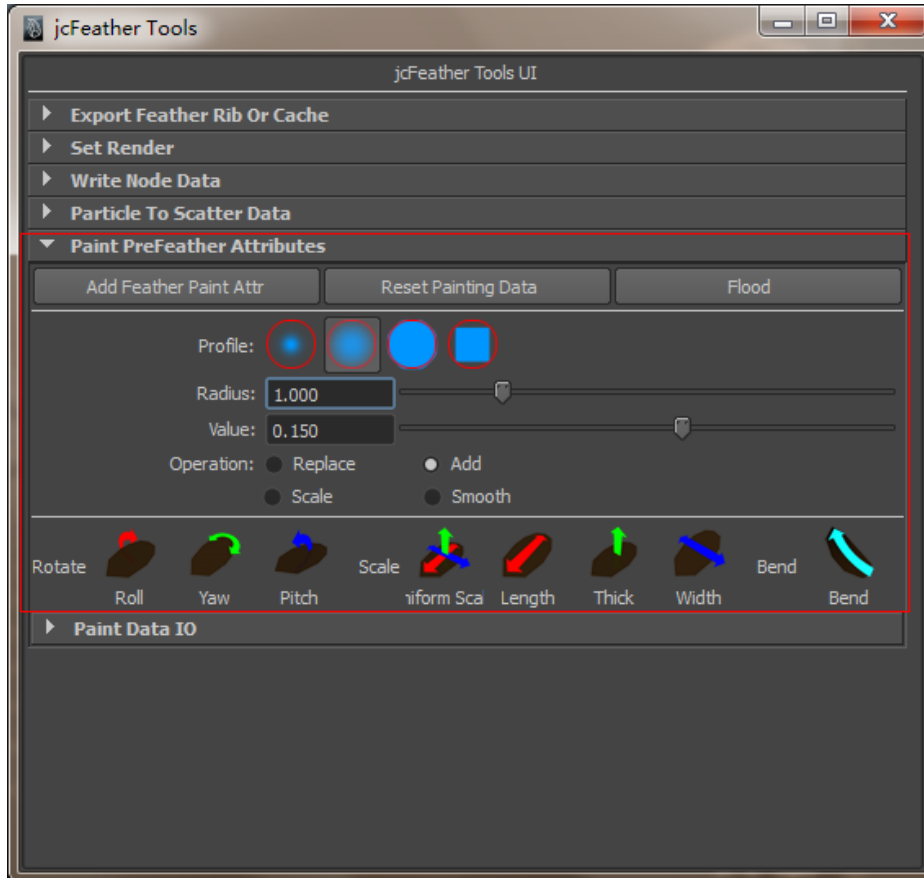
9, Set jcPreFeather1's attribute.



10, Set jcScatter node's attribute to get more feather mesh.



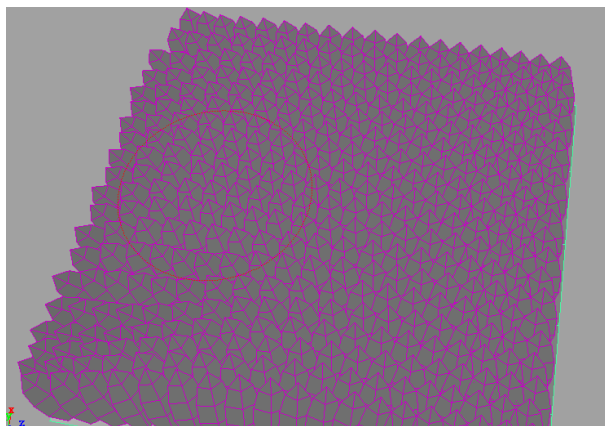
11, Click shelf tool . Use the paint brush to paint the feather direction and size.

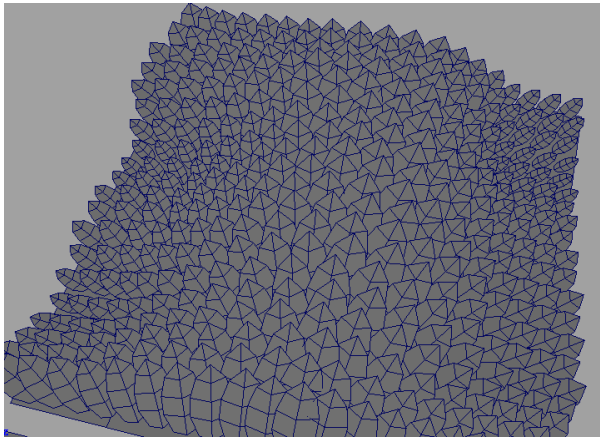


12, Select feather coat mesh, click the icon button Yaw to paint the rotate yaw attribute. Change the Profile, Radius, Value, Operation value as you need.

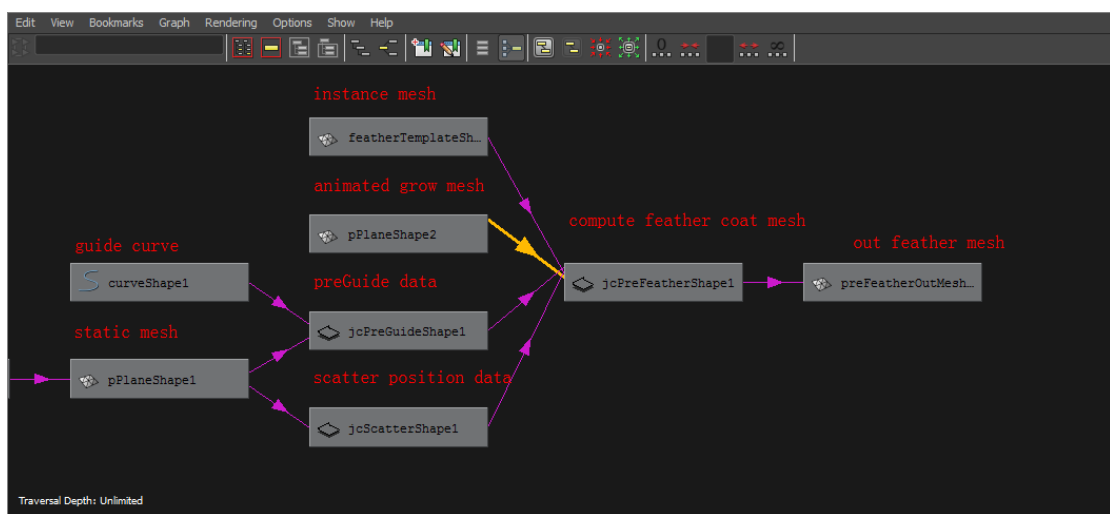


13, Paint on the grow mesh in the viewport. Repeat the step 13-14 to get the final mesh. You can paint feather's rotate attribute Roll, Yaw and Pitch, scale attribute Uniform Scale, Length, Thick, Width and Bend.

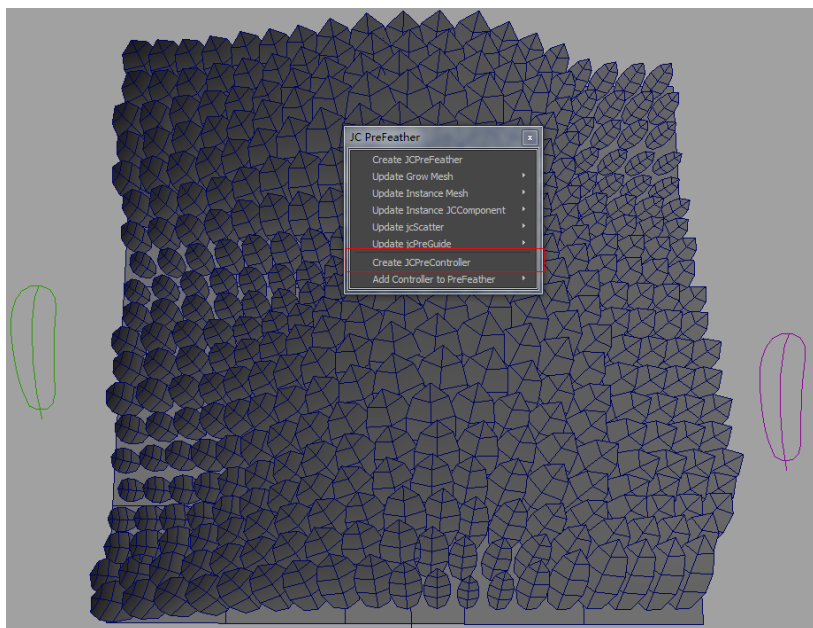




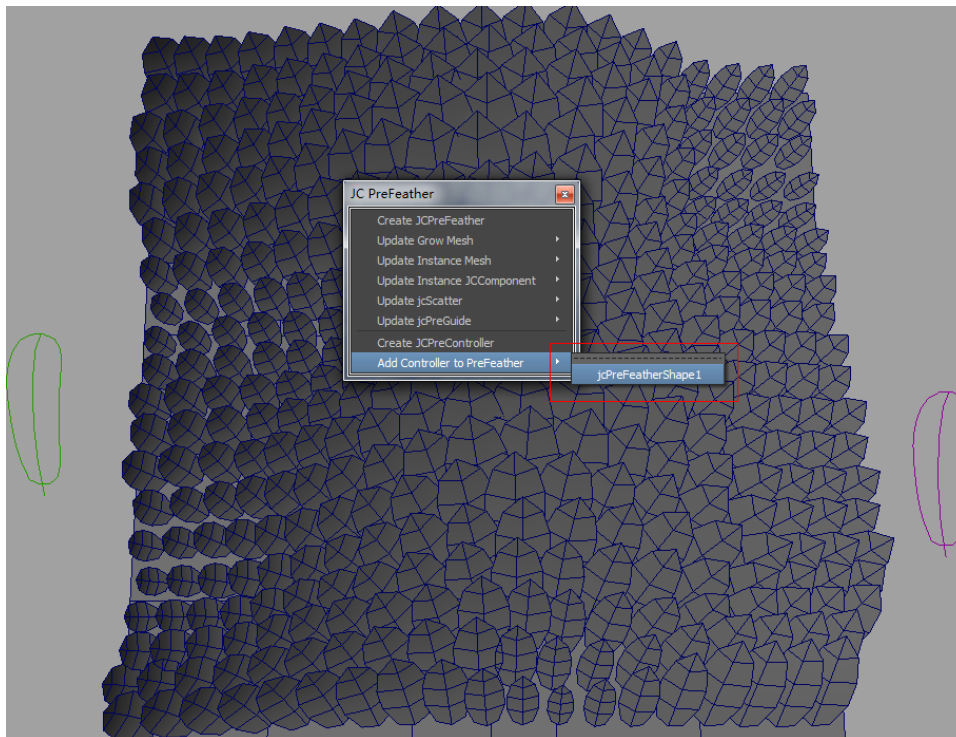
14, Take a look at the HyperGraph node connections.



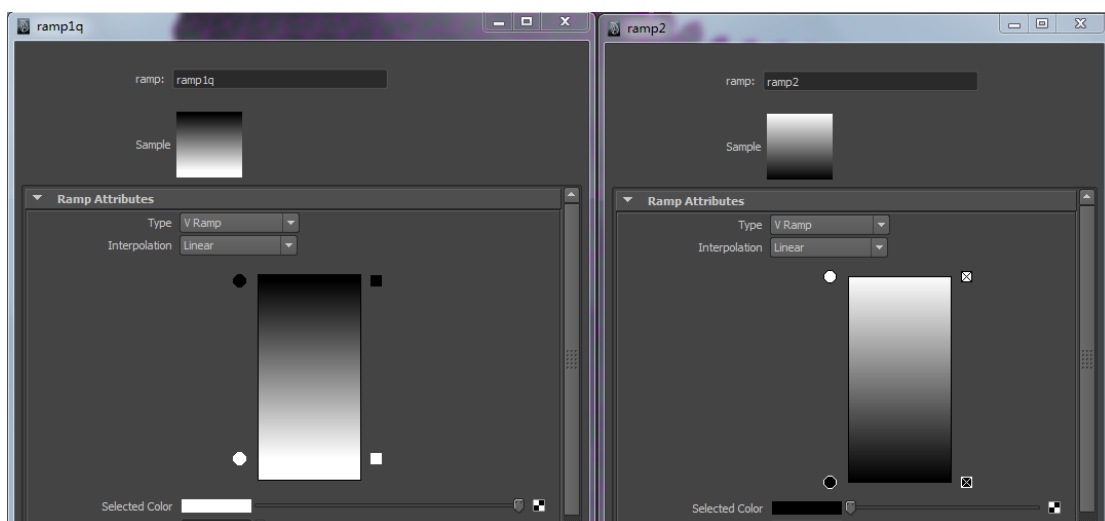
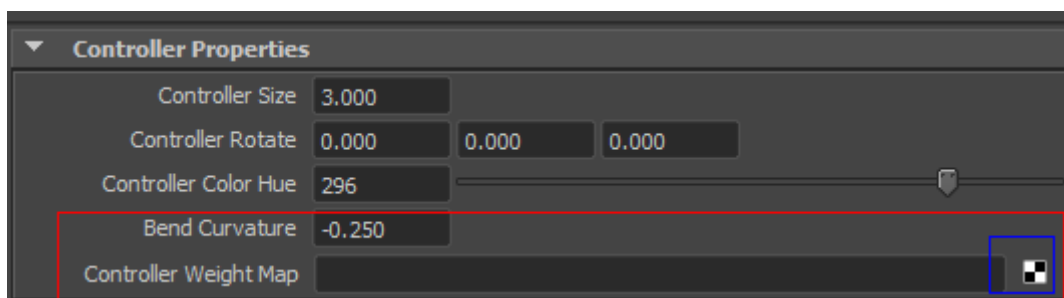
15, Create 2 jcPreControllers with the menu below. And change the color to green and red.



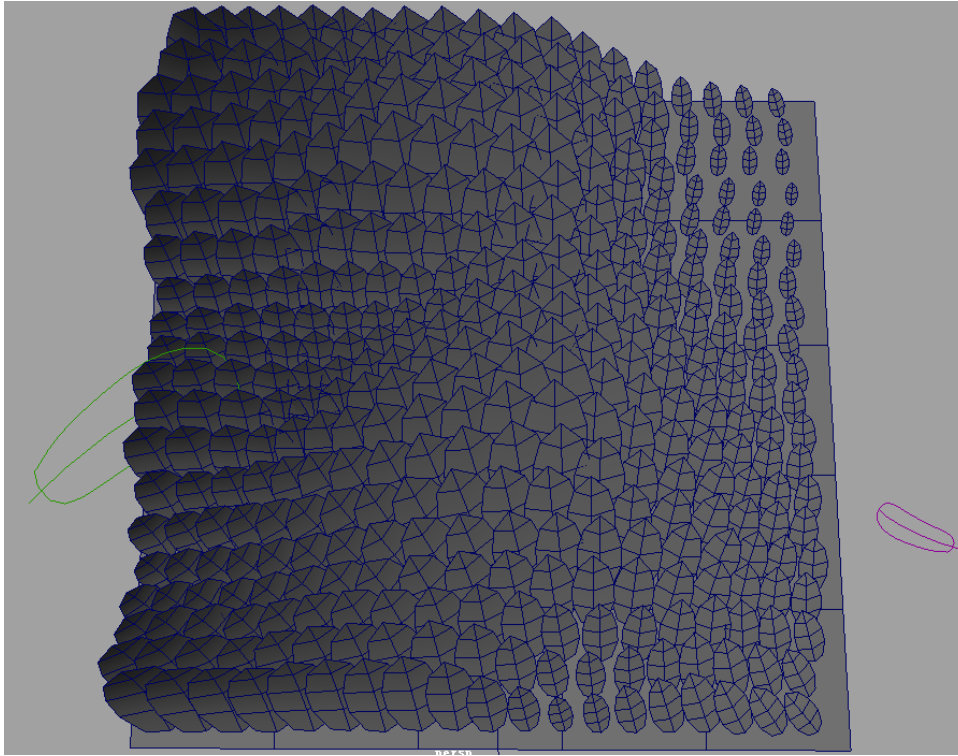
16, Select a controller, add the controller to jcPreFeather node.



17, Select jcPreController node and create a ramp texture node for each precontroller node by clicking here.



18, Rotate and scale the controller.



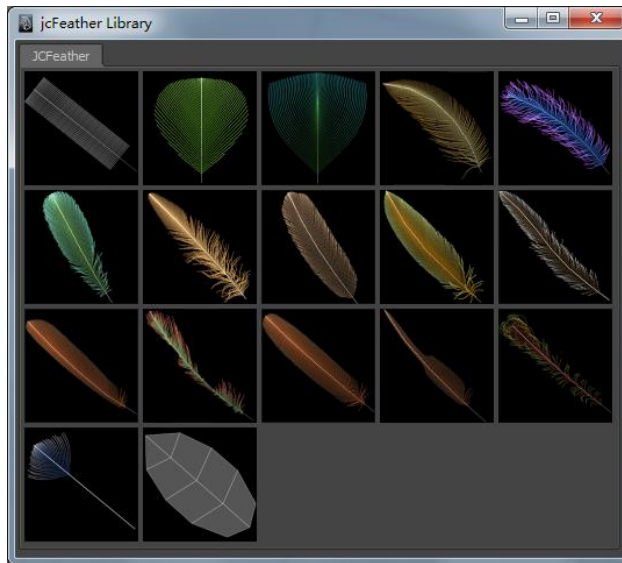
Chapter 3

Tools UI

3.1 jcFeather Library

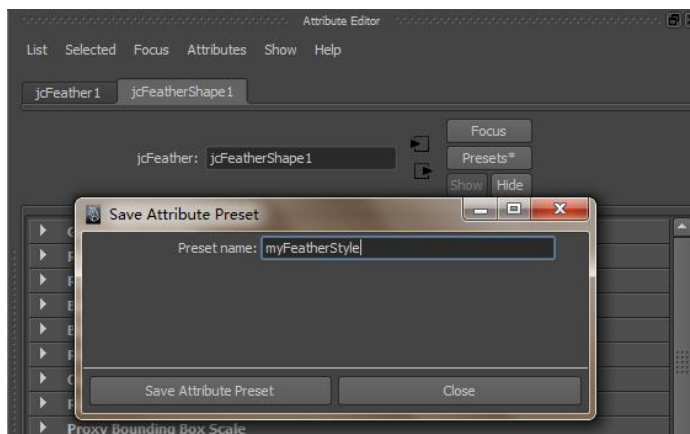


jcFeather Library collects many different kinds of feather style. Select a jcFeather node in your scene and click any icon to apply the preset style you like.



Also you can add your own feather style to this library.
For example.

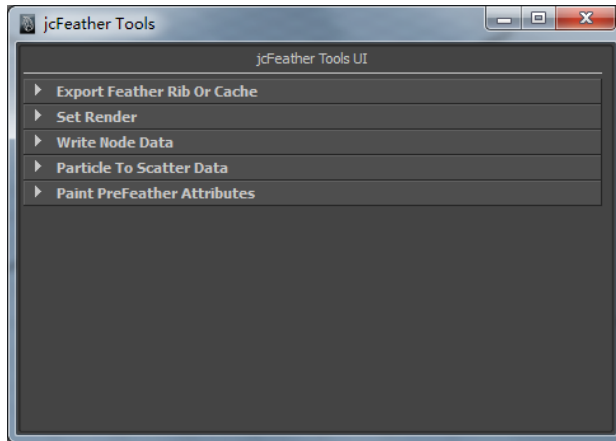
1. After saving your jcFeather preset named *myFeatherStyle*. You will get a *myFeatherStyle.mel* in the directory
C:\Users\~\Documents\maya\2011-x64\presets\attrPresets\jcFeather.
2. Copy the file *myFeatherStyle.mel* to *jcFeatherPath\scripts\attrPresets\jcFeather*
3. Make a png icon 200*200 named with the same name *myFeatherStyle.png*.
4. Copy the png file to *jcFeatherPath\icons*.
5. Reopen the jcFeather Library, you will get your style icon in the library.



3.2 *jcFeather Tools*



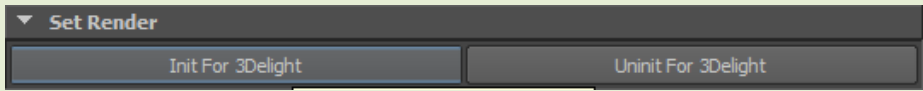
With *jcFeather Tools*, you can export jcFeather rib files and init your 3Delight for rendering. Write some node data out to files. Add particle positions to jcScatter node. Paint the feather coat mesh of jcPreFeather node.



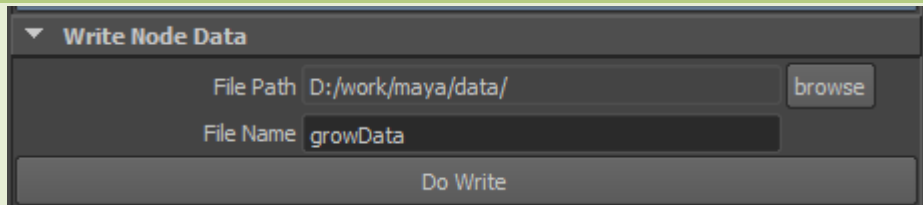
3.2.1

UI	
Intro	Export jcFeather data.
Export jcFeather	Out rib file directory.
Feather Name	Rib file name.
Frame Range	Frame range.
Use Selected	If on, only the selected jcFeather will be exported.
Out Type	The out data type of the feather. Currently only rmanRib is supported.
Export Feather Data	Do export.

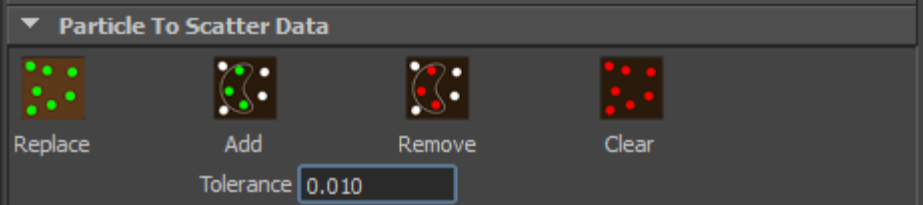
3.2.2

UI	
Init For 3Delight	Select a full renderpass to initialize it for later rendering.
Uninit For 3Delight	Remove the initial settings of selected renderpass for jcFeather.

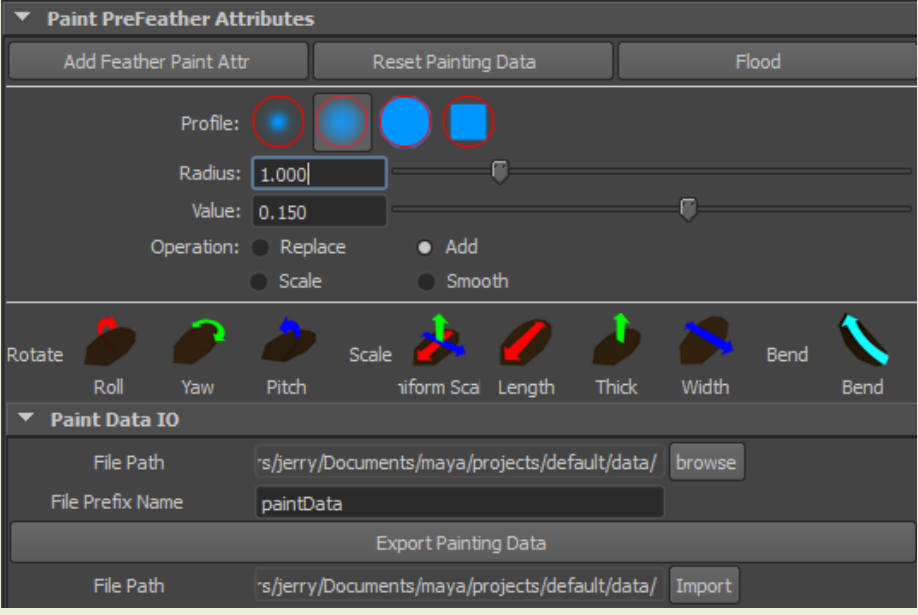
3.2.3

UI	
Intro	Export the jcScatter, jcPreGuide, jcFeatherComponentArray node data out. jcScatter and jcPreGuide data can be used in jcPreFeather node.

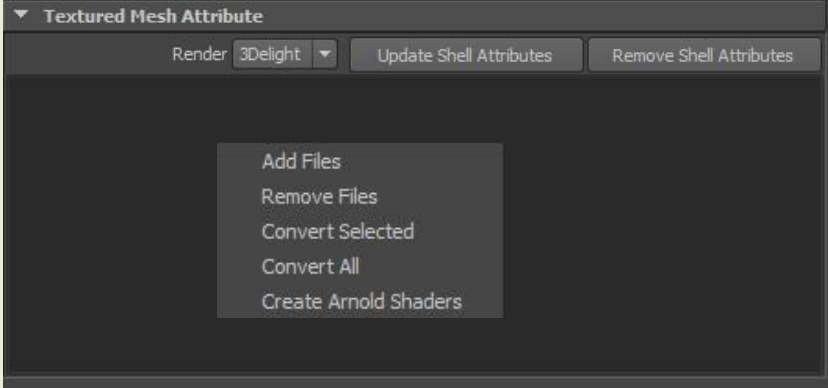
3.2.4

UI	
Intro	Select particle and jcScatter node to Replace, Add, Remove the position of the scatter points of jcScatter node. The algorithm depends on the jcScatter node's scatter mode, Uv and InPosition. Area mode will not use this operation. So please make sure the scatter mode first.
Replace	Replace the scatter points with selected particle.
Add	Add selected particle to jcScatter.
Remove	Find the closest points to selected particle, and remove them from the jcScatter node.
Clear	Clear all the selected jcScatter points.
Tolerance	The max distance to find the points for Add and Remove operation. In UV scatter mode, it is uv unit. In InPosition scatter mode, it is worldspace unit. So

3.2.5

UI	
Intro	This tab is used to paint the rotate and scale of preFeather coat mesh. you can select the mesh and paint the Roll, Yaw, Pitch, Uniform Scale, Length, Thick, Width attribute.
Add Feather Paint Attr	Add feaheer painting attributes. This operation has been done at Create JCPreFeather.
Reset Paint Data	Reset rotate to 0, scale to 1,bend to 0.
Flood	Flood the operation and value to all feather mesh.
Paint	Change the value of Profile, Radius, Value, Operation for your painting.
Rotate,Scale	Click the button to paint the specified attribute.
Paint Data IO	<p>"Export Painting Data", export all the rotate,scale painting data to 6 *.iff image.</p> <p>"Import", import all the 6 *.iff image to the 6 attributes. This button only needs a directory containing 6 iff files. Files will be imported and assigned to the attributes according to their names postfix.</p>

3.2.6

UI	
Intro	<p>This tab is used to add and update array attributes to feather coat mesh. The attributes will be used for later rendering for 3delight and arnold. And in the scroll list, right click to bring the popup menus. You can convert images files to mipmap textures for 3delight or arnold.</p>
Render	<p>Select the renderer you want to use.</p>
Update Shell Attributes	<p>Select jcFeatherComponentArray node to add and update attribute to the input connected mesh shape. For arnold, "mtoa_uniform_jcshells", " mtoa_uniform_jcshellst", " mtoa_uniform_jcshellid " will be added. Arnold will export those attributes data for later rendering.</p>
Remove Shell Attributes	<p>Select jcFeatherComponentArray node to remove attributes from the input connected mesh shape.</p>
Add Files	<p>Add disk images to this scroll list.</p>
Remove Files	<p>Remove the selected files from this scroll list.</p>
Convert Selected	<p>Convert selected images to *.tdl for 3delight, *.tx for Arnold.</p>
Convert All	<p>Convert all the images in the scroll list to mipmap textures.</p>
Create Arnold Shaders	<p>Select textures in the scroll list to create a default Arnold shader network for feather coat mesh.</p>

3.3 *Jc Grow Feather*

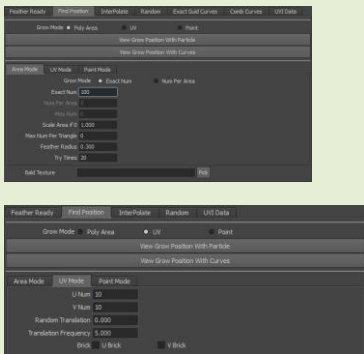


Jc Grow Feather scripts is used to help you make a feather coat. First it scatters particles on geometry and then replaces the particles with geometries. It works similar to *JcScatter*, *JcPreFeather* node. This procedure is obsolete.

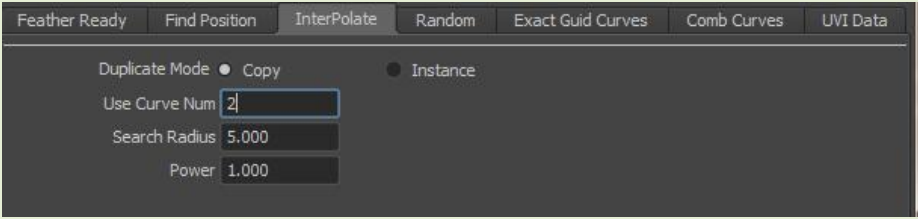
3.3.1

UI	
Intro	This tab is used to prepare some geometries for later growing.
Subdivision	Subdivision of feather template.
Make Feather Template	Click it to create a poly plane.
Feather	Select a feather template, click <i>pick</i> to use it to instance.
Grow On	Select a geometry, click <i>pick</i> to use it as the body.
Guid Curves	Select curves, click <i>pick</i> use them as guid curves.

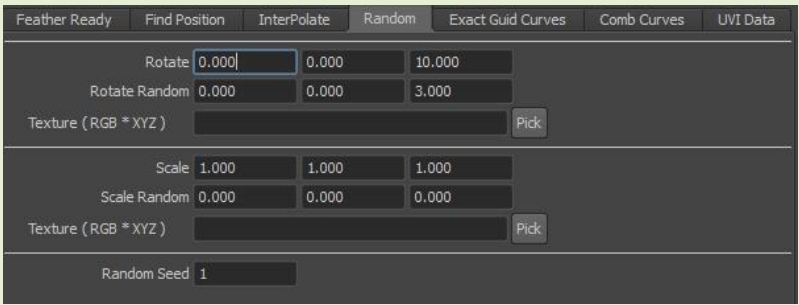
3.3.2

UI		
Intro		<p>This tab is used to compute position for each feather object.</p> <p>In Poly Area mode, particles will be scattered on poly faces based on the face area.</p> <p>In UV mode, particles will be scattered based on the uv of the geometry.</p> <p>In Point mode, you can use particles as the growing position.</p>
Grow Mode		How to scatter particles on geometries.
View Grow Position With Particles		Grow particles on the body geometry.
View Grow Position With Curves		Grow curves on the body geometry.
Bald Texture		Sample a 2d texture and use the v(hsv) value as the grow density.
Area Mode	<i>Exact Num</i>	How many geometries will be scattered for the whole mesh face area.
	<i>Num Per Area</i>	Geometry num for each unit face area.
	<i>Max Num</i>	The max num of geometries.
	<i>Scale Area if 0</i>	Scale the area if the face area if geometry num on this face is 0.
	<i>Feather Radius</i>	The size of each geometry. It's used to compute self collision.
	<i>Try Times</i>	Iteration for computing the self collision.
UV Mode	<i>U Num</i>	The geometry num in the u direction.
	<i>V Num</i>	The geometry num in the v direction.
	<i>Random Translation</i>	Random the particle position
	<i>Translation Frequency</i>	Random frequency.
	<i>Brick</i>	Brick effect in the u and v direction.
Point Mode	<i>Use Selected Points To Grow</i>	Select particle points and click this button. The particle positions will be use to scatter geometries.
	<i>Select Growing Points</i>	Click to select the points those are used to compute position.

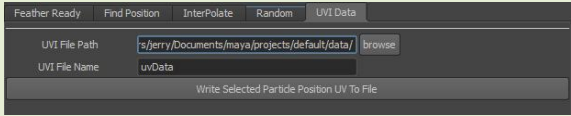
3.3.3

UI	
Intro	This tab is used to compute the direction for each feather object.
Duplicate Mode	Copy or instance feather object.
Use Curve Num	The closest num curves will be used to interpolate the direction.
Search Radius	Only the curve that is in this range will be used to compute direction.
Power	The fall off parameter of each curve weight.

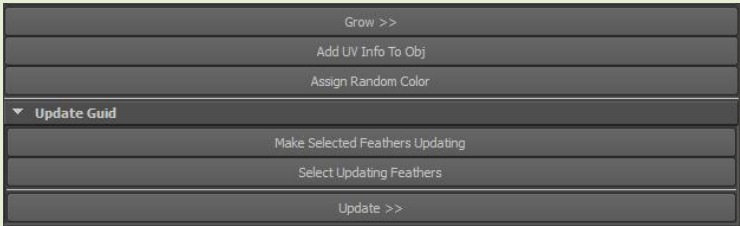
3.3.4

UI	
Intro	This tab is used to rotate and scale the feather object.
Rotate	Rotate feather object along xyz pivot.
Rotate Random	Add random rotation along xyz pivot.
Texture(RGB*XYZ)	Select a 2d texture and click pick to use the texture rgb values to multiply the final rotation.
Scale	Scale feather object along xyz pivot
Scale Random	Add random scale along xyz pivot.
Texture(RGB*XYZ)	Select a 2d texture and click pick to use the texture rgb values to multiply the final scale.
Random Seed	Random seed.

3.3.5

UI	
Intro	This tab is used to write out relative position data file *.uvi. It outputs each particle's closest uv value on the <i>Grow On</i> (see Feather Ready tab) geometry. The uvi file can be used in <i>jcFeatherSystem</i> node to generate feathers.
UVI File Path	File directory.
UVI File Name	File name.
Write Selected Particle Position UV To File	Do write out.

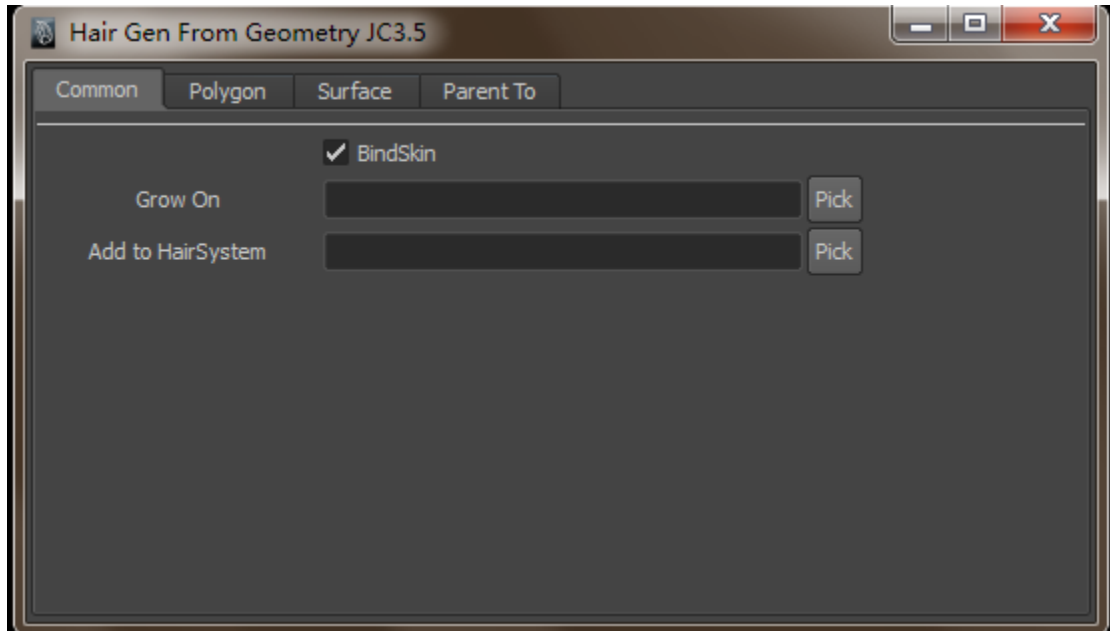
3.3.6

UI	
Intro	All the buttons bellowing do the computing.
Grow>>	Do grow computing.
Add UV Info To Obj	Add 2 attributes to selected geometries and store the closest uv value to them.
Assign Random Color	Assign random color to selected geometries.
Make Selected Feathers Updating	Select the feather geometries generated by this procedure to make them active for updating.
Select Updating Feathers	Select active updating feather geometries.
Update>>	Do update.

3.4 Hair Gen From Geometry



This is a script to make geometry dynamic with hairSystem in Maya. It has nothing to do with feather making.

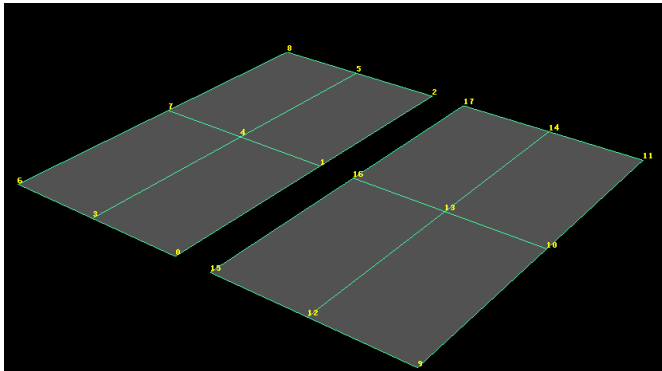




3.5 Create multiple *jcFeatherComponent*

This script is used to quickly select mesh vertices with some rules. It is designed to quickly make multiple *jcFeatherComponent* to make feather.

The procedure is a little bit similar to *jcFeatherComponentArray* node. Actually they use the same rule to make feather component. The difference is this procedure create multiple single *jcFeatherComponent* node. The *jcFeatherComponentArray* node create multiple feather data with one node.



For example, the mesh in the above image is a combined mesh with two meshes. If we want use vertices 0 1 2, 3 4 5, 6 7 8, 9 10 11, 12 13 14 and 15 16 17 to make feathers. We just input *Shell Num*:-1, *Shell Vertices Num*:9, *VertexGroup*:0 1 2, *Step In Shell*:3, *Group Per Shell*: 3. Select the mesh and click *Create Component* button to get six *jcFeatherComponent* node named *jcFeatherComponent#*. This rule means we will use every 3 vertices until to the end of the mesh.

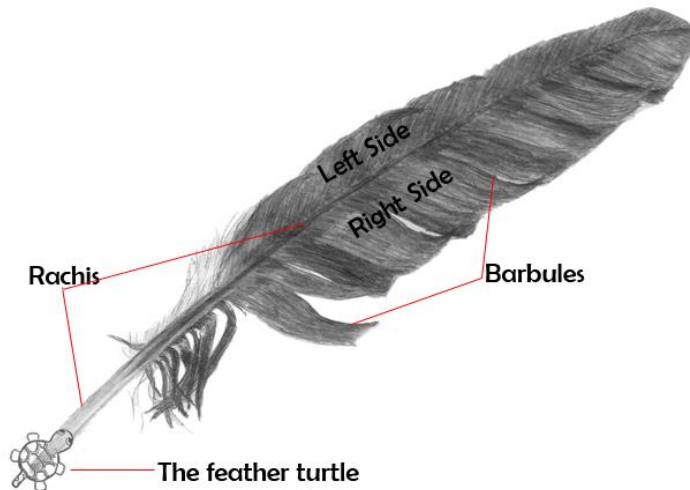
UI	
Shell Num	How many shells in the mesh. This example 2. -1 means using all the shells in the mesh.
Shell Vertices Num	How many vertices in each shell of the mesh. This example 9.
Step In Shell	The vertices step in each group. This example 3.
Group Per Shell	How many vertices group in each shell. This example 3.
Vertex Group	The start vertices group. This example 0 1 2.
Component Name	The prefix name of the set or the <i>jcFeatherComponent</i> .
Clear Vertex Group	Clear the text field Vertex Group.

Select Vertices	Select the vertices computed by the rules.
Set Vertex Group	Set the selected vertices id to the Vertex Group.
Create Vertices Set	Create set for each vertices group.
Create Component	Create jcFeatherComponent node for each vertices group.

Chapter 4

Technique Details In JCFeather

3.1 The Feather World In JCFeather:



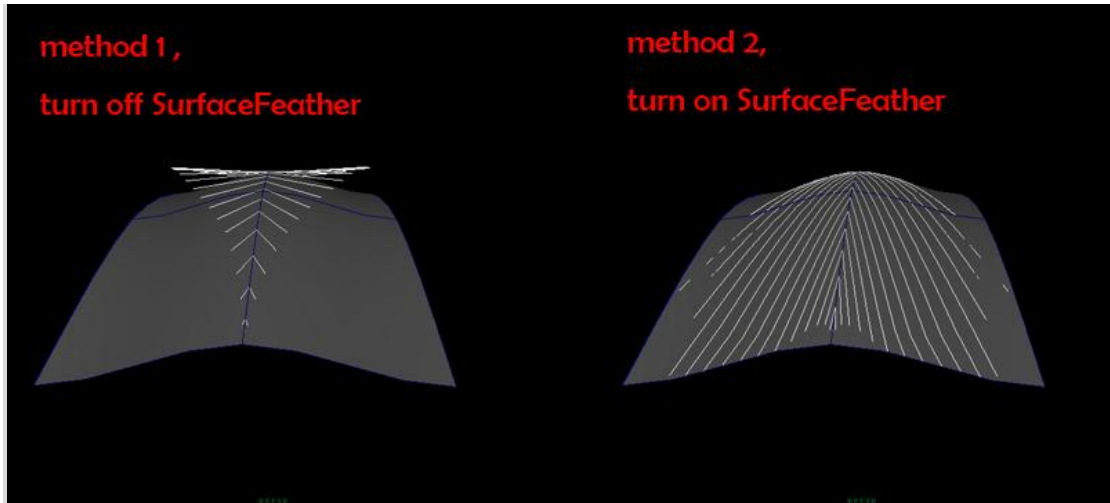
In JCFeather, I call the main branch rachis, and call the little branch barbule.

I shape individual feathers with a system similar to L-System. I put a feather turtle at the root of the rachis, the feather turtle heads for the rachis tip. The feather turtle walks step by step, and turns left or right at the cross of the rachis and barbules to shape feathers.

With nurbs surface as the basic geometry, I have two methods to shape a feather. First one, I use the u isoparm or v isoparm to locate the rachis. Then at the cross of the rachis and barbules, the feather turtle turns around the normal vector at the cross point on the nurbs surface. And the feather turtle goes to the end of the barbule. This means the feather turtle may go far away from the nurbs surface.

Another one, I also use the u isoparm or v isoparm to locate the rachis. But the feather turtle will always stick on the nurbs surface. This means all the barbules will always lie on the nurbs surface.

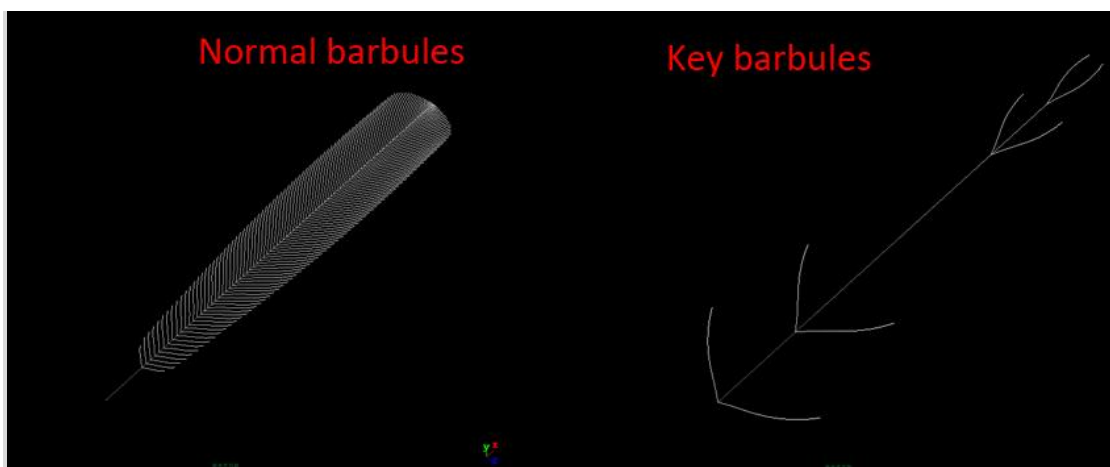
The image bellowing shows the difference of this two methods. I turn off the surfaceFeather parameter of jcFeather to use method 1, turn it on to use method 2.



With mesh vertices, I use *jcFeatherComponent* and *jcFeatherComponentArray* node to make a feather shape. These two nodes use the vertices defined in their attributes to make the rachis curve of the feather.

3.2 Behind The Barbules Shape:

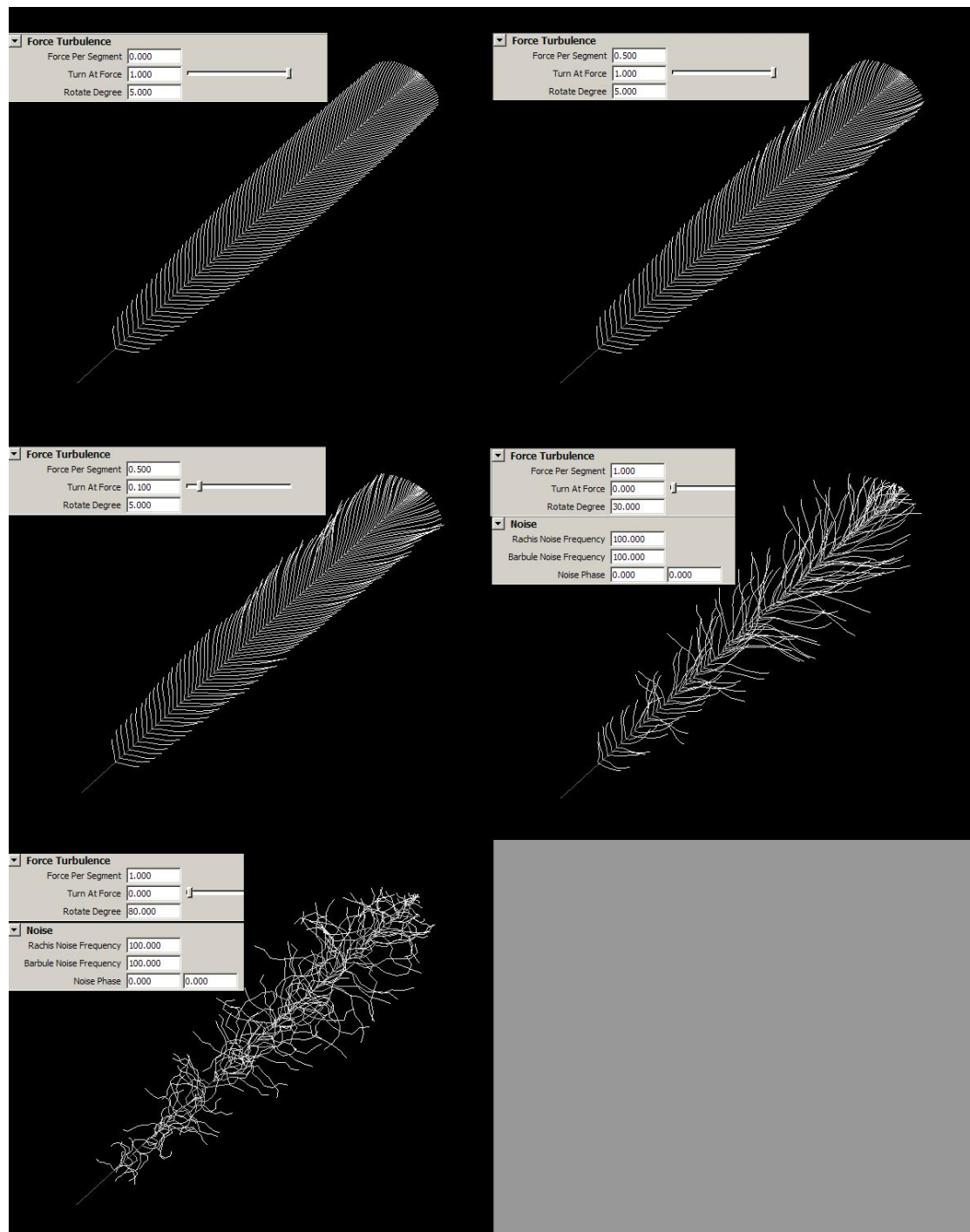
To shape barbules, I use 4 key barbules as the guid barbules. You can change the *displayAs* parameter of the *jcFeather* to *keybarbule* to see the key barbules. See the below image.



The key barbules only affect the bend angle of the normal barbules(rendered barbules). You can change the key barbule's position and the bend angle of it. See *Key Barbules* tab in *jcFeather* attribute editor.

Note: Normally the 4 key barbules locate in the order 1 2 3 4 from root to tip. You can change the order to others. But that would make unexpected barbule shapes.

To make the barbules fluffy or random, I use barbule turbulence. Expand the *Barbule Turbulence* tab in the attribute editor , you will get some parameters to tweak.



Different style feather with force turbulence

Here is the basic method to turbulence the barbules.

At the root of each barbule, the feather turtle get a force(let's call it force) valued 0. And then at each segment of the barbule, the feather turtle will add a random value

to its force value. When its force is bigger than the number you give, the feather turtle turns left or right based on the noise function and clear the force to 0 for the next adding. You can change the frequency and phase of that noise function to get different style turbulence.

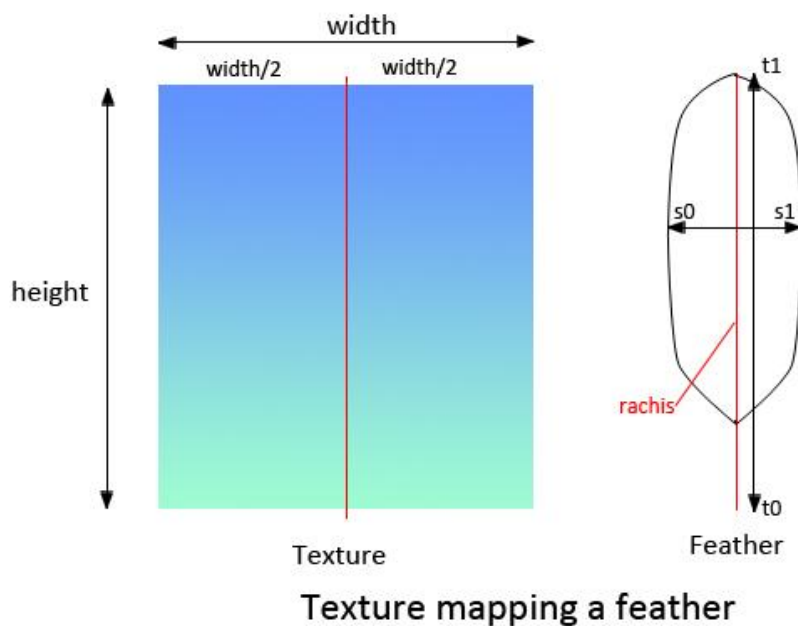
When come into the attribute editor, the feather turtle add a random number whose max value is *ForcePerSegment* at each segment. When the accumulated force is bigger than *TurnAtForce* , the feather turtle turns a degree whose max value is *RotateDegree*. See the image *Different style feather with force turbulence*.

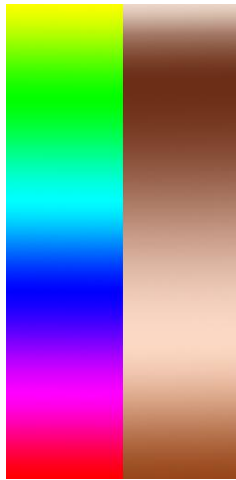
3.3 Rendering JCFeather:

Currently this plugin only supports 3Delight. JCFeather uses maya command to export feather data to renderman rib files. Then read those files into your scenes with a *delightRibArchive*(named *jcFeaReadArchive*) node for rendering. You can read those rib files in your own RenderMan-compliant render ,for example *prman*.

Based on the hair and fur shader, I wrote a new shader for the feather. Feathers are different from fur, but they are both Ricurves in renderman. So the shader adds some special features to render the feather. You can also specify a picture for the feather.

You pick a picture in the attributer editor, and it will be converted to *.tex(*.tdl in 3delight). The texture will be projected on the feather. The feather image in chapter 1 uses this texture method. See the image *Texture mapping a feather* and *Rendered feather with the texture*.





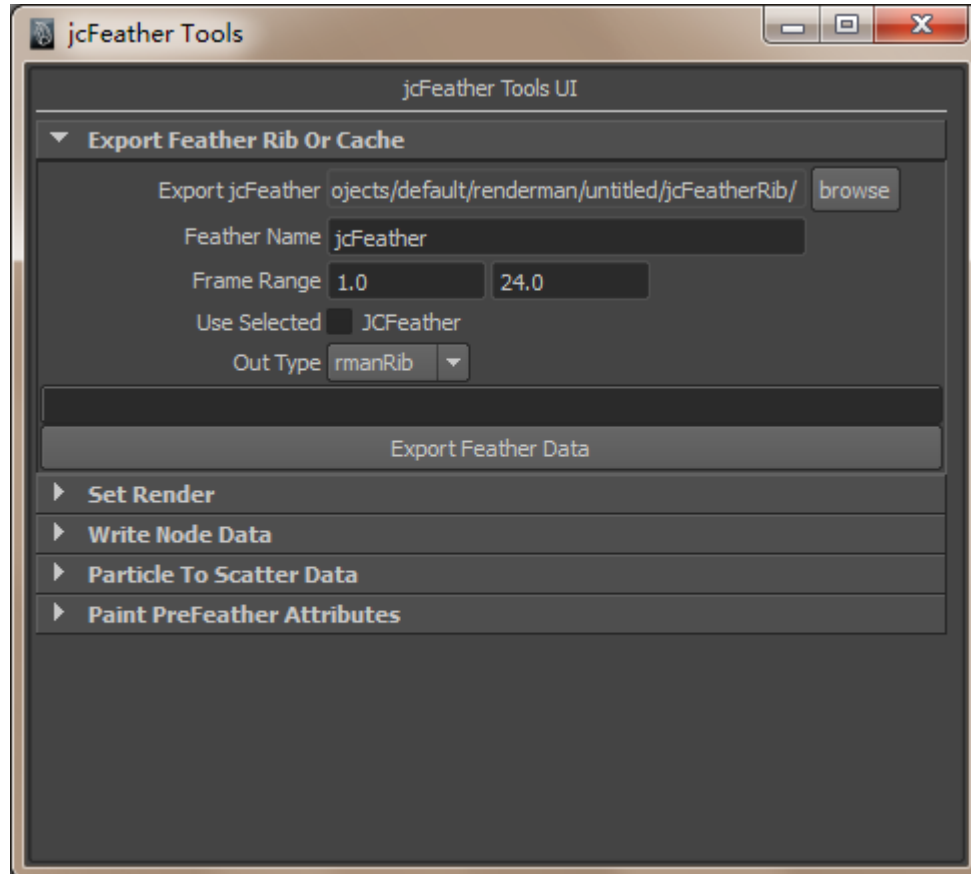
Texture



Rendered feather with the texture

In *JCFeather Global Settings*, there are also some parameters for rendering. For example the motion blur factor.

You can click *jcFeather Render Tools* button on the *jcFeather* shelf to export feather. This editor will export all the active feathers.



3.4 *jcFeather* in *prman* and *3delight*:

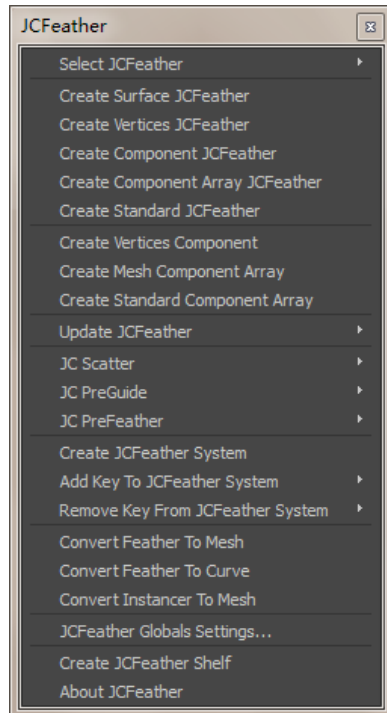
You can render *jcFeather* with *prman* and *3delight*. Before rendering *jcFeather* you need to init some settings for the render. Open the *jcFeather Tools*, the four buttons are used to do the init things.

Init for renderman, just click the button *Init For Renderman* to do settings, and click the *Uninit For Renderman* to remove settings (Not tested).

Init for *3delight*, you need create a full renderpass for *3delight*, and select the pass, then click the *Init For 3Delight* button or *Uninit For 3Delight*.

Chapter 5

Menus Reference



Menu	Intro
Select jcFeather	Select existing jcFeather in this scene.
Create Surface jcFeather	Select nurbs surface uv ranged in 0 to 1, then click <i>Create Surface Feather</i> to make a new jcFeather.
Create Vertices jcFeather	Select mesh vertices, then click <i>Create Vertices Feather</i> to make a new jcFeather. The vertices must be owned by the same mesh shape.
Create Standard JCFeather	Select a standard jcFeather template mesh(import from jcFeather Library), then click to create a standard jcFeather.
Create Vertices Component	Select mesh vertices those must be in order, then click this menu to create a <i>jcFeatherComponent</i> . The <i>jcFeatherComponent</i> node will use your selected vertices id to output info for later making jcFeather.
Create Component JCFeather	Select <i>jcFeatherComponent</i> node and click this menu to use its output info to make a jcFeather.
Create Mesh Component Array	Select mesh and click this menu to create a <i>jcFeatherComponentArray</i> node. Use

	<i>jcFeatherComponentArray</i> to make multiple <i>jcFeather</i> with one mesh.
Create Standard Component Array	Select a standard <i>jcFeather</i> template mesh or combined mesh to create a standard <i>jcComponentArray</i> node.
Create Component Array Feather	Select a <i>jcFeatherComponentArray</i> node or <i>jcPreFeather</i> node to create a new <i>jcFeather</i> .
Update JCFeather/Add Surface to jcFeather	Select nurbs surfaces to add them to an existing <i>jcFeather</i> .
Update JCFeather/Add Vertices to jcFeather	Select mesh vertices to add them to an existing <i>jcFeather</i> .
Update JCFeather/Add Component to jcFeather	Select <i>jcFeatherComponent</i> node to add it to an existing <i>jcFeather</i> .
Update JCFeather/Add ComponentArray to jcFeather	Select <i>jcFeatherComponentArray</i> node or <i>jcPreFeather</i> node to add it to an existing <i>jcFeather</i> .
Update JCFeather/Add Feather Mesh To ComponentArray	Select meshes to add a new mesh to <i>jcComponentArray</i> to make a new feather instance.
Update JCFeather/Update Grow Mesh To ComponentArray	Select a mesh to update the <i>inGrowMesh</i> of <i>jcComponentArray</i> . It is used to get the closest uv on the mesh for feather info.
Update JCFeather/Update Feather Texture	If you turn off <i>jcFeather</i> 's "interactiveColor" to get fast feedback in the viewport, you may need to update its texture after its main color texture changes.
Remove From jcFeather	Select nurbs surfaces or <i>jcFeatherComponent</i> or <i>jcFeatherComponentArray</i> to remove them from the <i>jcFeather</i> .
JC Scatter / Create JCScatter	Select a mesh to create a <i>jcScatter</i> node.
JC Scatter / Update Grow Mesh	Select a mesh or mesh faces to update <i>jcScatter</i> 's grow mesh.
JC PreGuid / Create JCPreGuid	Select a poly mesh or mesh vertices to create a <i>jcPreGuid</i> node.
JC PreGuid / Update JCPreGuid Grow Mesh	Select a poly mesh or mesh vertices to update an existing <i>jcPreGuid</i> node's input grown mesh.
JC PreFeather / Create JCPreFeather	Select a poly mesh to create a <i>jcPreFeather</i> node.

JC PreFeather / Update Grow Mesh	Select a poly mesh to update an existing jcPreFeather node's input grow mesh.
JC PreFeather / Update Instance Mesh	Select a poly mesh to update an existing jcPreFeather node's input instance mesh.
JC PreFeather / Update Instance JComponent	Select a jcComponent node to update jcPreFeather's input instance jcComponent.
JC PreFeather / Update JCScatter	Select a jcScatter node to update jcPreFeather's scatter points.
JC PreFeather / Update JCPreGuide	Select a jcPreGuid node to update an existing jcPreFeather node's input jcPreGuid.
JC PreFeather / Create JCPreController	Create a jcPreController node.
JC PreFeather / Create JCPreController	Select a jcPreController node to add the controller to an existing jcPreFeather node.
Create jcFeather System	Select mesh or nurbs surface , click it to create a jcFeather system node on it. Then you can contribute a jcFeather as key feather to the system to create a feather coat on the surface or mesh. Use <i>Add Key Feather To jcFeatherSystem</i> menu to add key feather.
Add Key Feather To jcFeather System	Select a jcFeather node, click it to add the key feather to you're the feather system node.
Remove Key From jcFeather System	Select a jcFeather node , click it to remove the feather node from the feather system node.
JCFeather Global Settings...	Each scene has a global settings node to define parameters for jcFeather in rendering. Click it to select the node and tweak its parameters.
Convert Feather To Mesh	Convert selected jcFeather to mesh.
Convert Feather To Curve	Convert selected jcFeather to nurbs curves.
Convert Instancer To Mesh	Convert particle instance to mesh. The instanced objects must be polygon.
Create JCFeather Shelf	Create a shelf called JCFeather which has some tools to make feather making easier.
About jcFeather	Show some information about jcFeather.

Chapter 6

Nodes Reference

jcFeather

jcFeather is the most important node in this plugin. You can connect several nurbs surfaces or mesh vertices to this node to make feathers. Also you can connect jcFeather to jcFeather System to make feather coat.

Attribute Name	Type	Default Value	Detail
active	Bool	True	Whether this feather node is active or not, it will affect jcFeatherSystem node.
render	Bool	True	Whether this feather node will be rendered or not. If turned off, this feather will be displayed in yellow color in viewport. This attribute will not affect jcFeatherSystem node.
uniformWidth	Bool	False	If turned on ,the output RiCurves will have uniform width .Each curve will not vary along curve u direction. The rib file has less data and will be rendered more efficiently.
surfaceFeather	Bool	False	If true feather babules will exactly lie on the nurbs surface if this node is based on nurbs surface. Or only rachis will lie exactly on the nurbs surface.
displayQuality	int	20	The percent of barbules those will be displayed in viewport.
curveType	Enum	1	The feather curves will be rendered in linear or cubic.
displayAs	Enum	0	The feather will be displayed as rendered babules or key barbules.
direction	Enum	0	The feather will be made along u or v direction if it computes on nurbs surface.
rachisRenderAs	Enum	0	The feather will be rendered as curves or subdivision mesh or not rendered.

rachisSegment	Int	10	The segment of rachis curve.
rachisPosition	Float	0.5	This value defines the u or v position of the rachis on the nurbs surface.
rachisStart	Float	0	This value defines the position of the rachis root on the nurbs surface.
rachisEnd	Float	1	This value defines the position of the rachis tip on the nurbs surface.
rachisSides	Int	4	If you render rachis as subdivision mesh or convert this feather to mesh, this value defines the sides of the rachis cylinder.
rachisThick	Float	0.1	This value defines the curve width or the rachis cylinder's radius if you convert it to mesh, or render it as subdivision mesh.
rachisThickScale	Ramp		This value defines the width scale along the rachis from root to tip.
rachisRootColor	Color		This value defines the basic root color of the rachis.
rachisTipColor	Color		This value defines the basic tip color of the rachis.
barbuleNum	Int	100	This value defines the number of barbules on each side of the feather.
barbuleSegment	Int	8	This value defines the segment of the barbule curve.
barbuleDensity	Ramp		This defines the density of the barbule along the feather rachis from the root to the tip.
shapeSymmetry	Bool	True	If true, the barbule length of each side is the same.
barbuleLength	Float	1	This defines the length of the barbule.
barbuleLengthRandScale	Float	0	How large the barbule will be randed.
barbuleLengthTex	Float	1	Connect a 2D texture to this parameter to make each feather barbules length scale based on the texture.
leftBarbuleLengthScale	Ramp		The length scale of the left side of the feather.
rightBarbuleLengthScale	Ramp		The length scale of the right side of the feather.
gapForce	Float2	0	From the rachis's root to its tip, a random force value for each barbules

			will accumulate. This attribute means the the force will be [0,gapForce].
gapMaxForce	Float2	1	When the accumulated force is beyond gamMaxForce, a gap appears.
gapSize	Float2	3	Scale the gap size.
leftGapDensity	Ramp		The density of the left gaps along the rachis.
rightGapDensity	Ramp		The density of the right gaps along the rachis.
forcePerSegment	Float	0	There is a random force value On each vertex of the barbule. This defines the max number of the force. I add the force one bye one to the total force value. When the total force value is bigger than the parameter <i>turnAtForce</i> , the barbule rotate <i>rotateDegree</i> .
turnAtForce	Float	1	See the parameter <i>forcePerSegment</i> .
rotateDegree	Float	5	See the parameter <i>forcePerSegment</i> .
forcePerSegmentScale	Ramp		Scale the <i>forcePerSegment</i> value along the barbule from root to tip.
forceRotateScale	Ramp		Scale the <i>rotateDegree</i> along the barbule from root to tip.
upDownNoise	Bool	True	When you need feather rotate up or down, this adds noise random to the up and down degree.
upDownLength	Float	0	The amplititude of up and down.
barbuleUpDownRachis	Ramp		The scale of up and down along rachis from root to tip.
barbuleUpDownBarbule	Ramp		The scale of up and down along barbules from root to tip.
rachisNoiseFrequency	Float	10	The noise frequency of up down and force random along rachis.
barbuleNoiseFrequency	Float	20	The noise frequency of up down and force random along barbule.
noisePhase	Float2	0	The noise phase of <i>rachisNoiseFrequency</i> and <i>barbuleNoiseFrequency</i> .
barbuleThick	Float	0.2	The width of the barbule curve.
barbuleRandThick	Float	0	How much random the width of the barbule curve will be.
barbuleThickScale	Ramp		The scale of the width of the barbule along the barbule.

barbuleThickAdjust	Ramp		The scale of the width of the barubule along the rachis.
keyBarbulePos1 2 3 4	Float		The position of the key barbule 1 2 3 4. It ranges 0 to 1 from rachis's root to its tip.
keyBarbuleStartAngle1 2 3 4	Float		The start angle the feather turtle will rotate at the root of the key barbule.
keyBarbuleStepAngle	Float	10	The angle of the feather turtle will rotate at each segment of the barbule.
kbRotate1 2 3 4	Ramp		The angle scale of the feather turtle will rotate at each segment of the barbule along the barbule.
rootColor	Color		The basic root color of the barbule.
tipColor	Color		The basic tip color of the barbule.
projectTexUV	Bool		If true, a uv map will be projected onto the feather from right above the feather. So you can use texture on the feather.
uvProjectScale	Float	1	Scale the uv map along the width of the feather.1 means no scale.
colorTexture	String		Use this to pick a painted texture.
autoConvertTex	Bool	True	If true, the colorTexture will be automatically converted to *.tdl for 3delight. Or you can rewrite the AEjcFeatherTemplate.mel to convert it to other textures such as PRman's *.tex.
interactiveColor	Bool	True	If you have lots of feather in one jcFeather node, the "mainColor" texture computing is time consuming. So you can turn it off and update the color by clicking menu "Update JCFeather/Update Feather Texture" when needed.
mainColor	Color	1	Connect a texture to this parameter to make each feather different based on the texture.
textureProcedure	string		jcFeather will call this attribute as a mel procedure. The procedure should be something like <i>global proc string getTex(float \$u,float \$v,int \$id)</i> <i>\$u</i> and <i>\$v</i> are the feather uv value on

			<p>the object for <code>jcFeatherSystem</code>. $\\$i$ is the feather id in the node. For <code>jcFeather</code> node you must give the <code>u</code> and <code>v</code> value on <code>nurbssurface</code> or <code>jcFeatherComponent</code> those are connected to <code>jcFeather</code>.</p> <p>The returned string represents a texture path. For example: <code>D:/color.tex(for prman)</code></p>
rootOpacity	Float	1	The base opacity of the barbule.
tipOpacity	Float	1	The tip opacity of the barbule
fadeStart	Float	1	The opacity will fade from <code>rootOpacity</code> to <code>tipOpacity</code> at this position.
barbuleDiffuse	Float	0.5	The diffuse of the barbule.
barbuleSpecularColor	Color		The specular color of the barbule.
barbuleSpecular	Float	1	How much specular the barbule will be.
barbuleGloss	Float	0.09	The gloss of the barbule specular.
selfShadow	Float	1	The selfShadow of the barbule.
hueVar	int	0	Color Hue variation of each barbule.
satVar	Float	0	Color saturation variation of each barbule.
valVar	Float	0	Color value variation of each barbule.
varFrequency	float	325.1 2	Variation frequency of the color.
outputFeatherMesh	Bool	False	If true, this feather will output a mesh if a mesh node is connected to <code>outRachisMesh</code> or <code>outBarbuleMesh</code> .
outRachisMesh	mesh	NULL	Output rachis as mesh.
outBarbuleMesh	mesh	NULL	Output barbules as mesh.
meshUVScale	Float2	1	Scale the uv data of the output mesh.
randSeed	Int	4352	The seed used in this whole feather node.
featherInsertRib	String		Something to insert before the feather Ricurves in the rib files.
rachisMeshInsertRib	String		Something to insert before the rachisMesh if you render the rachis as subdivision mesh.
displayProxyBoundingBox	Bool	False	Whether display proxy boundingbox of the feather. If on, a green proxy boundingbox will be displayed for each feather in the viewport.
proxyBoundingBoxScale	Float		Scale proxy boundingbox along

			barbule, rachis and feather normal.
inFeatherTurtleData	feather TurtleD ata		This attribute should always be connected to jcFeatherComponent node's outFeatherTurtleData attribute.
inFeatherTurtleArrayData	feather TurtleA rrayDat a		This attribute should always be connected to jcFeatherComponentArray node's outFeatherTurtleDataArray.

jcFeather system

jcFeather System is based on a creature body which might be a nurbs surface or a mesh and a jcFeather node connected to it.

Attribute Name	Type	Default Value	Detail
active	Bool	True	If turned on, the feather system will be computed and rendered.
displayBarbuleQuality	Int	10	How many barbules will be displayed in the viewport. This won't affect the rendered result.
displayQuality	Int	20	How many feathers will be displayed in the viewport. This won't affect the rendered result.
globalScale	Float	1	Global scale of the feather in this system.
randScale	Float	0	How much random each feather's size will be.
guidsPerFeather	Int	2	How many key feathers each feather in this system needs.
Radius	Float	10	How far the feather will look for the key feather to affect itself.
Power	Float	1.0	This node interpolates the feather with Gaussian radial basis functions $f(c,r) = \text{pow}(e, -c^2r^2)$. c is power, r is the distance between the interpolated feather and the guid feather. So the bigger c is, the light weight each guid feather contributes. The smaller r it is, the heavier weight each guid feather contributes.
renderQuality	Int	100	How many feathers will be rendered out. 0 means nothing, 100 means all of them.
useUVIFile	Bool	False	If turned on, jcFeatherSystem will use a *.uvi file to locate the feather growing position on the body mesh or surface instead of uv method.
uviFile	String		Uvi file path. Uvi file is a binary file that you can get it using the JCGrowFeatherGuids mel procedure

			on the JCFeather shelf. The file contains points' position on mesh or nurbssurface. Get more detail in <i>jcFeather Tools</i> help doc.
uSegment	Int	10	The grid size along u on the surface.
vSegment	Int	10	The grid size along v on the surface.
uvOffset	Float2	0	Offset uv position of feathers every two lines of u and v.
uvTranslate	Float2	0	Translate the uv position of each feather.
jitterDistance	Float	0.1	How much the feather's position will be jittered.
jitterFrequency	Float	15	The frequency of the jitter.
colorMap			The map controls the color of the feather over the system surface.
baldnessMap			The map controls the baldness of the feather over the system surface.
scaleMap			The map controls the size of the feather over the system surface.
radiusMap			The map scale the research radius of the feather over the system surface.
randomSeed	Int	5	The random seed the system needs to compute.
displayProxyBoundingBox	Bool	False	If turned on, proxy boundingbox will be displayed in the view port in green color.
uvSetName	String		The uvSetName the detail maps need to compute.
outputFeatherMesh	Bool	False	The system will output a mesh if this value is turned on.

jcFeatherGlobals

jcFeatherGlobals includes the global settings for jcFeather such as the shader for jcFeather, motion blur stuff.

Attribute Name	Type	Default Value	Detail
featherRibText	String	jcFeather	Feather surface shader name should be put here.
rachisMeshRibText	String		Usually you put the rachis mesh surface shader here if you render the rachis as subdivision mesh.
diceHair	Bool	False	If use the attribute <i>Attribute "dice" "int hair" [1]</i> in rib files.
exportBarbuleUV	Bool	True	Export the barbuleUV for texture the feather.
exportSurfaceUV	Bool	False	If the feather is one of the feather system, turn this on to export the closest uv point to the feather root.
exportNormal	Bool	False	Export Ricurves normal based on the guide nurbsSurface and poly.
relativePath	Bool	True	Export rib files using relative path or full path.
deleteTempFile	Bool	True	If turned on, the temp Rib files will be deleted after rendering.
dso	Bool	False	Use dso plugin to render feather or not.
dsoName	String	jcFeatherDelight	The dso plugin name. You can write your own dso plugin for renderman.
delayReadArchive	Bool	False	If turned on, the rib files will be called with <i>Procedural "DelayedReadArchive"</i> . The boundingbox is the proxyBoundingBox of the jcFeather. This attribute also affects the rendering in dso mode. If on, dso will delay each jcFeather's computing with its proxy bounding box, if off, dso will compute once for all feathers.
relativePath	Bool	False	If turned on, the rib files will be

			constructed in relative Path.
gzipCompression	Bool	False	If turned on, the output rib files will be gzip compressed.
tempFileDir	String		The directory the rib files will be outputted to. %temp% means project \data\jcFeather\
enableMotionBlur	Bool	False	If turned on, the feather will have motion blur.
motionBlurSamples	Int	2	Samples times for motionblur.
motionBlurStep	Float	1.0	How many frames will be included to compute motion blur. The bigger it is ,the bigger the motion blur will be.
motionBlurFactor	Float	1	Scale the motion blur effects. The bigger this value is, the smaller the motion blur will be.
useMultiThread	Bool	True	Use multi thread or not.
maxThreads	Int	0	How many threads will be used. If 0,the thread num will be equal to the num of processors. Much higher value is not recommended.

jcFeatherComponent

jcFeatherComponent node collects mesh vertices info and output data for later making jcFeather.

Attribute Name	Type	Default Value	Detail
displayLocator	Bool	True	Display the locator or not.
exMeshFeather	Bool	false	exMeshFeather means "exact mesh feather". It needs "left vertex group" and "right vertex group" to shape a feather.
leftVertexGroup	String		The left vertex group ids to shape the left edge of the feather.
rightVertexGroup	string		The right vertex group ids to shape the right edge of the feather.
vertexGroup	String		This attribute describe the vertices used to define a feather rachis. It's a string which contains vertices id. For example, 0 1 2 3 means the vertices id. 0:3 also means 0 1 2 3.
locatorSize	Float	1	This attribute defines the locator size in 3D viewport.
featherUV	Float2	0	This attribute will be used in jcFeather node to make each feather different in color hint.
growVertexID	Int	0	This attribute is not used yet.
outVertexGroup	string		This attribute display the real vertices id num.
inMesh	Mesh		The vertexGroup attribute will find its vertices in this mesh.
outFeatherTurtleData	featherTurtleData		This attribute should always be connected to jcFeather node's inFeatherTurtle. It outputs the data that is

			used to make a feather.
--	--	--	-------------------------

jcFeatherComponentArray

jcFeatherComponentArray is an array type of jcFeatherComponent. It uses a kind of rule to collect multiple components to make multiple feathers. The rule in this node is something like this:

vertexGroup 0 1 2 3

walkStep 4

walkTimes 3

The real vertices that will be used to make feather is:

0 1 2 3,

$(0+4*1)$ $(1+4*1)$ $(2+4*1)$ $(3+4*1)$,

$(0+4*2)$ $(1+4*2)$ $(2+4*2)$ $(3+4*2)$,

So this node will make three feathers data 0 1 2 3, 4 5 6 7, 8 9 10 11.

If walkTimes is -1, this node will output data until the index reach to the max num of vertices of the input mesh.

Attribute Name	Type	Default Value	Detail
displayLocator	Bool	True	Display the locator or not.
exMeshFeather	Bool	false	exMeshFeather means "exact mesh feather". It needs "left vertex group" and "right vertex group" to shape a feather.
leftVertexGroup	String		The left vertex group ids to shape the left edge of the feather.
rightVertexGroup	string		The right vertex group ids to shape the right edge of the feather.
vertexGroup	String		This attribute describe the initial vertices used to define a feather rachis. It's a string which contains vertices id. For example, 0 1 2 3 means the vertices id. 0:3 also means 0 1 2 3.
locatorSize	Float	1	This attribute defines the locator size in 3D viewport.
walkStep	Int	1	This attribute defines the step for the next feather data.
walkTimes	Int	-1	This attribute defines how many feather data will be output. -1 means it will compute until reaching to the last vertex of the <i>inMesh</i> attribute.
inMesh	Mesh		The <i>vertexGroup</i> attribute will find its vertices in this mesh.
inGrowMesh	Mesh		<i>jcFeatherComponentArray</i> will find the

			closest uv and the closest vertex on this mesh for each feather data.
uvType	Enum	1	How to get uv data for each feather. The uv value will be used in <i>jcFeather</i> node.
useGrowVertexID	Bool	false	If on, the node will compute to find the closest vertex id on the <i>inGrowMesh</i> attribute.
uvSet	string		The node will find the closest uv on the <i>inGrowMesh</i> on this uvMap.
outFeatherTurtleDataArray	featherTurtleArrayData		This attribute should always be connected to <i>jcFeather</i> node's <i>inFeatherTurtleArrayData</i> .
outFeatherNumArray	intArray		Feather num for each input mesh.
outFeatherUArray	doubleArray		Feather u data for each single feather.
outFeatherVArray	doubleArray		Feather v data for each single feather.
outRootIdArray	intArray		Mesh root vertex id of each feather.
outRootPositionArray	vectorArray		The root vertex position of each feather.

jcPreGuide

jcPreGuide node is used to define the direction of vertices on mesh. This node needs several curves and a poly mesh , then it will find the closest direction of curves for mesh vertices. The output data is the direction data , it should be connected to *jcPreFeather* node. And you can also use mel command *jcWriteGrowData* to write the data to a *.vd file and then browse the file in *jcPreFeather* node.

Attribute Name	Type	Default Value	Detail
displayLocator	Bool	True	Display the preguide locator or not.
colorHue	Int	130	Locator color in hsv.
curvesPerVertex	Int	2	How many closest curves will be used to interpolate the direction of each point.
searchCurveRadiu s	Float	10.0	The range of searching for the closest curves.
curvePower	Float	0.6	The power parameter of the Gaussian radial basis functions $f(c,r) = \text{pow}(e, -c^2r^2)$. Here <i>c</i> is <i>curvePower</i> .
maxAngle	Float	10	If the angle between the closest interpolated direction of curves and the vertex edge is bigger than this value, the vertex will not have direction data.
sampleTolerance	Float	0.001	Tolerance to sample closest direction.
inGrowMesh	mesh		A static mesh should be connected to this attribute to compute the vertex direction.
inGuideCurves	nurbsCurve Array		Curves those will be used to interpolate the closest direction for each mesh vertex.
inComponentList	component List		The vertex componentList that will have direction data. If none, all the vertices of the inGrowMesh will be used.
outVertexDir			The direction data of all the mesh vertices.

jcScatter

jcScatter is used to scatter grow points on poly mesh. The grow points data should be connected to *jcPreFeather* node to make a feather mesh.

Attribute Name	Type	Default Value	Detail
displayLocator	Bool	True	Display the locator or not.
colorHue	Int	180	Locator color in hsv.
scatterMode	Enum	UV	How to scatter points on mesh. UV type means scattering grid points based on uv. Area type means scattering point based on poly area. InPosition mode means the inPosition attribute will be used, the closest point to the points in "inPosition" on grow mesh will be used to scatter.
numPerArea	double	10	In Area scatter mode, how many points should be scattered one unit area. One point per poly face at least.
maxDistance	Double	0.01	If "inPosition" scatter mode is used, if the distance between the closest point and the point in inPosition attribute is bigger than this value, the point will not be used.
uSegment	Int	10	The grid size along u on the surface.
vSegment	Int	10	The grid size along v on the surface.
uvOffset	Float2	0	Offset uv position of feathers every two lines of u and v.
uvTranslate	Float2	0	Translate the uv position of each feather.
jitterDistance	Float	0.1	How much the points' position will be jitterd.
jitterFrequency	Float	15	The frequency of the jitter.
uvSetName	string		The uvset name that will be used in this node.
baldnessMap			The map controls the baldness of the point on the mesh.
randomSeed	Int	5	The random seed the node needs to compute.
userUvData	Double2		In UV scatter mode, add a customer specified uv value to this node to get a

			customer points. When the value is negative , the point won't be instanced.
userUArray	doubleArray		Add customer u array values to get customer points. The size of this array should be equal with <i>userVArray</i> .
userVArray	doubleArray		Add customer v array values to get customer point. The size of this array should be equal with <i>userUArray</i> .
inGrowMesh	Mesh		A static mesh should be connected with this attribute to compute scatter points on it.
inComponent List	component List		The face componentlist that will have scattered points. If none, the whole mesh will be scattered.
inPosition	vectorArray		If "inPosition" scatter mode is used, the points in this attribute will be used to scatte points. You can set this attribute with some vector points or connect a vectorArray attribute to this attribute, eg, particle's "worldPosition" attribute.
outGrowData			The grow position data.

jcPreFeather

jcPreFeather node is a feather mesh instancer. It collects the out data of jcPreGuide and jcScatter to get the feather's direction and position, and scatter the instanceMesh on the inGrowMesh. The inGrowMesh must have the same topology with the jcPreGuide's inGrowMesh and jcScatter's inGrowMesh.

Attribute Name	Type	Default Value	Detail
Active	Bool	True	Active or not.
globalScale	Float	1.0	Scale the input mesh instance.
randScale	Float	0.0	Randomly scale the input mesh instance.
rotateOffset	Float3	0	Rotate each feather mesh along XYZ pivot.
useRotateNoise	Bool	False	Add noise to rotate or not.
rotateNoiseFrequency	Float	1.0	The frequency of the noise.
rotateNoiseAmplitude	Float3	0	The noise amplitude added to the rotate.
rotateNoisePhase	Float3	0	The noise phase.
rotateOrder	Enum	5	The rotate order of output feather coat.
interactiveScale	Bool	True	Whether computing the attribute "scaleMap" or not. Texture sampling is time consuming. So if you get the right size of feathers, please turn this attribute off to get faster feedback in the viewport if you have lots of feather polygon in this node.
scaleMap	String		Scale the feather mesh using a 2d texture.
randSeed	Int	1	The rand seed.
sampleTolerance	Float	0.001	Tolerance to sample closest direction. Usually it should be the same with the <i>sampleTolerance</i> attribute of the input <i>jcPreGuid</i> node.
uvSetName	String		The uvset name of the input mesh.

activeBend	Bool	False	Activate bend or not.
bendPosition	Float	0	Bend position from the root to the tip. It ranges 0 to 1.
bendCurvature	Float	1	The curvature of the bend.
bendRandCurvature	Float	0	Rand the curvature of the feather coat.
guideFile	String		The *.vd file generated from jcPreGuide node.
scatterFile	string		The *.st file generated from jcScatter node.
inGrowMesh	Mesh		The mesh that will grow feather mesh. It must have the same topology with jcScatter's and jcPreGuide's inGrowMesh.
instanceMesh	Mesh		The feather mesh that will be instanced on the inGrowMesh.
instanceFeatherTurtle			A jcFeatherComponent node should connected to this attribute.
controlInfo			This attribute should be connected with jcPreController node. It receive controller's data such as rotate, scale, bend and weight map, and interpolate the value for each feather polygon. The child attribute controlBend, controlRotate, controlScale are used to get the data from controller.
outFeatherMesh	Mesh		Output the feather coat mesh. Use this mesh to make a jcFeather coat. This attribute needs instanceMesh attribute connected.
outFeatherTurtleDataArray			This attribute works like the jcFeatherComponentArray node. Connect it to jcFeather node To get a feather coat. This attribute needs instanceFeatherTurtle attribute connected.

inScatterData			The scatter position data, it should be connected with jcScatter's outGrowData.
inVertexDir			The vertex direction data , it should be connected with jcPreGuide's outVertexDir.
paintRoll	DoubleArray		Paint attribute for feather roll.
paintYaw	DoubleArray		Paint attribute for feather yaw.
paintPitch	DoubleArray		Paint attribute for feather pitch.
paintLength	DoubleArray		Paint attribute for feather length.
paintWidth	DoubleArray		Paint attribute for feather width.
paintThick	DoubleArray		Paint attribute for feather thick.
paintBendCurvature	DoubleArray		Paint attribute for feather bend curvature.

jcPreController

jcPreController node is used to rotate and scale the each feather mesh which is computed by *jcPreFeather*. Usually, it should be connected to *jcPreFeather*. And when you create a jcPreController, it is recommended to put it in a transform node. Just by moving and rotating its parent transform node to change its location. Then rotate(scale) the jcPreController node to rotate(scale) the feather mesh.

Attribute Name	Type	Default Value	Detail
controllerSize	Float	1.0	The size of this controller.
controllerColorHue	String		This attribute is used to display the color of the locator .
controllerRotate	Float3	0	Rotate the locator.
bendCurvature	Float	0	Bend value for each controller.
controllerWeightMap	Float	0	This attribute should be connected to a 2d texture. <i>jcPreFeather</i> node will sample the luminance of the texture and use it as a weight map to control the feather mesh smoothly. To use this attribute, the attribute <i>controlMethod</i> in the jcPreFeather node should be <i>Color Map</i> .

Chapter 7

Commands Reference

jcWriteFeather

Flag	Type	Default value	Detail
-a/-ascii	Bool	true	If true, exported files will be ascii, or they will be binary. Now this command only supports ascii.
-ft/-fileType	Int	0	0 means renderman rib files. 1 means feather cache. 2 means feather cache and a rib file that is used to read cache for renderman render.
-p/-path	String	None	The path of exported files.
-s/-selected	Bool	False	Export the selected jcFeather, jcFeatherSystem node or all the renderable feather nodes.
-n/-nodes	String	None	Feather nodes that will be exported. If this flag is used, the -s flag will be overridden. -n "jcFeatherShape1 jcFeatherShape2".

This command is used to export jcFeather to files those can be rendered by renders. It writes rib files or mi files. Then you can read them in your scenes. The command get the attributes in jcFeatherGlobals node and use the parameters to do the export.

Example:

```
jcWriteFeather -a 1 -ft 0 -p "C:/feather.rib"
```

This command will write all active feathers in current scene to C:/ in ascii fommat renderman rib files.

```
jcWriteFeather -a 1 -ft 0 -s true -p "C:/feather.rib"
```

This command will export selected feathers.

```
jcWriteFeather -a 1 -ft 0 -n "feather1 feather2 feather3" -p "C:/feather.rib"
```

This command will export 3 jcFeather nodes named "feather1 feather2 feather3".

jcWriteGrowData

This command is used to write the jcPreGuid out data to a file *.vd file and jcScatter out data to *.st file. The *.vd file and *.st file can be read with jcPreFeather node to determine the direction and position of the feather mesh. This command has been put on the JCFeather shelf in *JCFeather Tools*.

Example:

```
jcWriteGrowData "D:/vdData.vd" jcPreGuid1
jcWriteGrowData "D:/stData.st" jcScatter1
```

jcFeatherAbout

Get jcFeather version info.

The result string will be like this, "JCFeather Version 2.2.5 For Maya2012x64 ".

jcFeatherToCurves

Flag	Type	Default value	Detail
-n/-nodes	String	none	Nodes name that will be converted to curves. If this flag is not used, the selected jcFeather nodes will be used to do conversion.
-aa/-addAttr	Bool	False	If this flag is true, a "feaBarbU" attribute will be added to each barbule curve. The attribute ranges from 0 to 1 for left barbules and 1 to 2 for right barbules. It means the parameter from the rachis root to the tip.

Convert jcFeather and jcFeatherSystem to nurbs curves.

Example:

```
jcFeatherToCurves;
```

This command will convert selected jcFeather nodes and jcFeatherSystem nodes to curves.

```
jcFeatherToCurves -n "feather1 feather2";
```

This command will convert feather1 and feather2 node to curves;

```
jcFeatherToCurves -n "feather1 feather2" -aa true;
```

This command will convert feather1 and feather2 node to curves, and add some feather attributes to each barbule curve. Currently only one "feaBarbU" attribute,

some new attributes will be added in the future.

jclnstanerToMesh

Flag	Type	Default value	Detail
-n/-nodes	String	none	Instancer nodes name that will be converted to mesh. If this flag is not used, the selected instancer nodes will be used to do conversion.

This command convert a particle instance to mesh. The instancer must be mesh.

Example:

```
jclnstanerToMesh -n "instancer1 instancer2";
```

instance1 and instancer2 will be converted to mesh.

jcQueryData

Flag	Type	Default value	Detail
-bb/-boundingbox	none	none	Use this flag to query the bounding box of cache file.
-p/-path	String	None	Specify the xml file path.

This command is used to quer the feather cache xml file data. Currently you can use it to query the total bounding box of cache.

Example:

```
float $boundingBox[]=`jcQueryData -bb -p C:/a.xml`;
```

Chapter 8

JCFeather SDK

After jcFeather 2.8.0, two types of jcFeather develop kits are available.

8.1 C++ sdk outside Maya

You can use jcWriteFeather command in maya to export jcFeather to small cache files, and generate feather curves data in memory for your own render with the sdk. But you need to install the specific version of Autodesk Maya to work correctly.

JCFeather cache files include a xml file which contains the feather attributes and some *.fc files which contain the feather contour data. In sdk, you read in the xml file, and get feather curves data. You can find a simple example cpp file in the %JCFeather%/samples/ directory. Find more information in the sdk c++ header files.

include files : jcFeatherSDK.h jcFeatherSDKDataType.h
library files : jcFeatherLib.lib

8.2 C++ sdk inside Maya

Inside Maya, you also have a sdk to generate feather curves data from MObject (jcFeather node). A simple example cpp file can be found in the %JCFeather%/samples/. Find more information in the sdk c++ header files.

include files: jcFeatherMayaSDK.h
library files: jcFeatherMaya.lib

Chapter 9

Current Limitations And Future Work