FEATHERING WILD PIGEON

MSc MASTER'S PROJECT

Shahnaz Kamberi

September 2006

Table of Contents

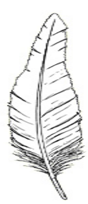
Introduction

"Feathers, like hair a couple of years ago, have become a hot topic in CG animation" (Framestore 2006). Although recently popular, not much modelling techniques have been studied in creating feathers (Franco 2001). The purpose of this project was to create a final 'feathered' rendered character. The Wild Pigeon character modelled and animated previously (Please see Major Animation Project) was used for the feathering process.

Background Information

a) Feathers: Models, Structure, Types

Feathers are made up of rachis and barbs, rachis is the main part of the feather and the barbs originate from the rachis (Franco 2001). There are different types and sizes of feathers varying from 1-2mm (Streit 2002). "The length and angle of barbs can also vary, usually leading to ovade (egg-shaped) feathers, but may also result in obovate (upside-down egg), spatulate (short broad tips), or pointed" (Streit 2002). There are two common types of feathers and they are contour and semi-plume (Franco 2001).



Contour Feather (Anon 2002)

Semi-plume feather (Anon 2002)

Although similar to hair and fur, feathers however are a little more complicated (Streit 2002). Feathers are more complicated then hair in that they are coarser and have many different colours and patterns (Streit 2002). "Feathers' interaction with each other [is] subtler than that of hair […] the interaction of individual feathers could not be fudged, as it might be with hair" (Framestore 2006). Currently in CG animation, there are methods in modelling feathers. A simple way of modelling a contour or semi-plume type feather is to define 3 cubic Bezier curves as a rachis, and then from the rachis generating a number of Bezier curve barbs (Franco 2001). Another method of modelling feathers is by making key feathers using Bezier curves, and interpolating between its parameters and control points which allows for smooth transition in the feather structure (Streit 2002). Specifically, the modeling of feathers is "based on the notion of key representations at multiple levels of detail. An individual feather is designed using a set of parameters describing the length and curvature of the rachis, the length and angle of the barbs, and a set of key barbs for the vane curvature. The curvature of intermediate barbs is specified by interpolating the key barbs. These individual feathers are then used as key feathers in the design of the feather coat, by interpolating this parameterization to create intermediate feathers" (Streit 2002).

After structuring the feather, more detail and a texture map/shader can be added to the geometry (Streit 2002).

b) Texturing/Shading and Rendering

A feather pattern or a scan of an actual feather can be used as a texture map in creating textures for feathers (Streit 2002). The barbs of the feathers are textured so that texture mapping could be "simple and reusable" (Streit 2002). "The texture can be specified in three ways: on the barb, on the vane, or at the base of the rachis during feather growth" (Streit 2002). Best way of texture mapping is by "generating texture coordinates ignoring rachis curvature" (Streit 2002). RenderMan shader can also be applied to the geometry, instead of a texture map. The feather information can then be exported and rendered.

c) Productions Using CG Feathers

Several productions have used CG feathers. *Valiant* is one of the productions that required the use of CG feathers. *Valiant* is a full length animation of a story of a carrier pigeon.

Valiant: Characters (Anon 2005)

The modelling and animating of the characters were done in Maya 4 (Moltenbrey 2005). Once the modelling of the birds was finished the geometry was exported in to Houdini (Moltenbrey 2005).

"Each of the 30 main bird characters went through a rigorous grooming process that formed the basis for the feathering throughout the film. After modelling and rigging the characters in Maya, the team imported them into Houdini, where shaders were assigned. Guide hairs, grown from the surfaces, defined how the feathers looked and behaved-for instance, if they were ruffled or flat. Texture maps, meanwhile, dictated the type of feather, its directionality, colour, and other properties that guided how they looked and moved within a given shot" (Moltenbrey 2005).

Along with the rigorous grooming of each of the characters, problems such as the feathers' reaction to clothing had to be solved by the production company (Moltenbrey 2005). To solve this problem, "Houdini's Attribute Transfer tool" was used "to define areas where the feathers would need to be pushed down. […] By using Houdini, Vanguard (the production company) was able to apply this sophisticated feathering solution to all the bird characters by using a small team of 2-3 artists" (Anon 2005). Once the problem of the feather behaviour was solved, each bird had a total of 50,000 feathers and the amount of feathers was not altered throughout the making of the film (Moltenbrey 2005).

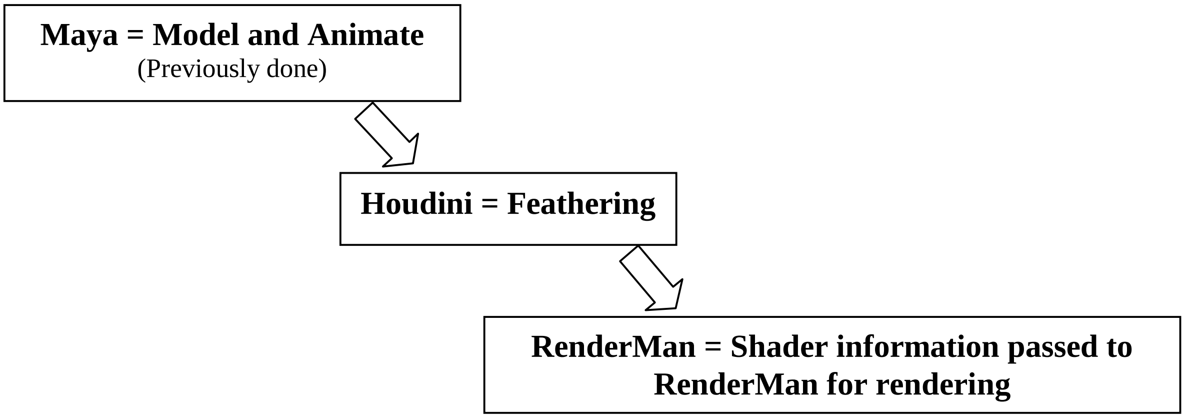
Harry Potter is another production where CG feathers were used. Fawkes is a character used in *Harry Patter*, this character is a "mythological bird that immolates itself and is then born" (Framestore 2006).

Fawkes from Harry Potter (Wikipedia 2006)

The production company studied live birds in order to come up with their pipeline (Framestore 2006).

Both productions discussed used Houdini to do the feathering of their characters, and both were very successful in creating realistic feathers. The pipeline decided on for feathering Wild Pigeon was thus based on the above findings.

Pipeline for Feathering Wild Pigeon



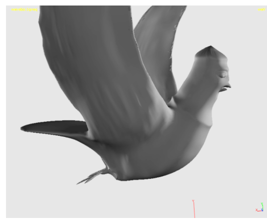
Wild Pigeon Feathering Process

a) Exporting out of Maya into Houdini

Since the Wild Pigeon model was created and animated in Maya, and the feathering process done in Houdini, the first step in feathering Wild Pigeon was to export the animated model from Maya into Houdini. The mel scripts exportSeries.mel and exportSeriesWin.mel, found from http://www.lunaseven.com/images/Maya2LW.zip, were used to accomplish this task.

The exportSeries script consists of a function which contains variables int $i, string $fullname and float $ct. The function operates by using variable $ct to save the current time, and uses a for loop to iterate through and export out an OBJ for each frame as $fullname followed by the frame number. The script only works when the Maya OBJ export plug-in is loaded. The for loop function contains three if functions for naming purposes. The three if functions maintain the naming process by keeping track of the range of frames in which the geometry is being exported out.

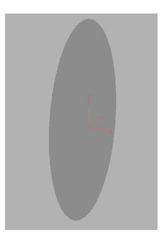
The exportSeriesWin.mel script basically creates an interface window where the user of the script can export out a series of OBJs by specifying the name of the file and the time range. The script works for a frame range up to 999 frames. Once the OBJs are exported out of Maya, they are then imported into Houdini. The normals of the mesh are reversed as the OBJs are imported into Houdini; therefore, the normals were reversed back. The OBJs were exported out as bgeo files (used in Houdini). For this project, the bgeo files of the model were used for the feathering process.



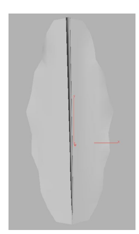
Pigeon Model Used for Feathering

b) Grouping and Grooming

Once the model was imported into Houdini, different areas of the pigeon character were grouped for placement of feathers. The main grouped sections were the non-feathering and feathering. The non-feathering sections consist of eyes, iris, pupils, nose, nostrils, beak, hat, legs and feet. The feathering sections consist of the face, neck, body, wings, and tail. After the feathering sections were grouped, the feathers were placed on them. The surface normal of the geometry and its direction played an important role in the placement of feathers. "The surface normal is a unit vector perpendicular to the local surface" (Cromwell 2004). A polygon's surface normal is calculated by the cross product of two edges of the polygon (Wikipedia 2006). The grouped areas of the pigeon were worked on separately. The 'feathers' used for placement on the different sections of the model were the circle primitive, and geometry created in the shape of a contour feather.



Circle Sop used as Feather

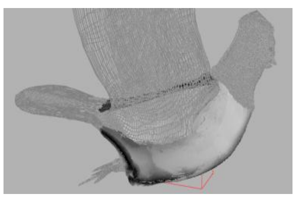


Contour Feather shape used as Feather

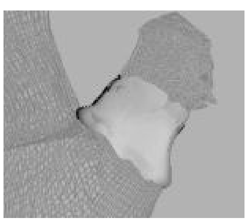
The circle primitive (circle sop in Houdini) was transformed to be different sizes and shapes for the different parts of the body; to conform to the different sections it was placed on. The circle sop was mainly used to cover areas of the pigeon where the feathers are soft and small. The geometry in the shape of the contour feather was created by the curve sop. Half of a contour feather was drawn by curves and mirrored to create a feather shape. A line sop was used to create a rachis down the middle of the feather. The line sop was connected to a polywire sop to make it a polygonal shape. This contour feather created was used on certain areas of the model such as the bottom half of the wings and the bottom half of the tail.

The normals of the selected geometry had to be calculated in order to scatter the feathers on the surface. The placements of the feathers were only the first step. After the feathers were scatted onto the geometry the next stage was to keep the feathers facing the right direction, to keep the feathers facing the correct direction as the geometry moved, and making sure the feathers covered all of the area so that once the geometry underneath is deleted, the feathers will not have spaces in between. These and other problems were solved mainly by changing the parameters of the point and scatter sops in Houdini and maintaining the normals of the primitive areas. The scatter sop in Houdini is an “operation that generates points at random locations over surfaces” (Houdini help). “It distributes points in a roughly uniform pattern and attempts to limit clumping or holes” (Houdini help). For scattering feathers onto the body, face, and neck area, the scatter sop was attached to each of the grouped sections and the feathers copied on top. Once the scattering was completed, the original geometry was deleted underneath so that only the feathers were left. The feathers had to be transformed in a specific manner for each area of the body in order for the model to be groomed properly.

The feathers for the face area were smaller then that of the other areas of the body. The  face area was groomed carefully so as to not have any feathers in the eyes, nose or the beak of the pigeon. Each feather on the face existed in the YZ plane, and the circle (feather) was transformed to be .3, .3 in radius, the centre coordinates of the circle was (-.1, 0, -.1), and it was rotated 0 degrees in the x-axis, 90 degrees in the y-axis, and 90 degrees in the z-axis. These parameters allowed for the feathers to be in the correct direction and cover all of the face geometry without covering the eyes, nose or the beak. There were a total of 12,000 primitive circles scattered and copied onto the face of the pigeon model.



The body of the pigeon was groomed with different sets of feathers. The feathers on top of the body geometry were bigger then that of the feathers on the face. The centre coordinates were also different then that of the face. The size of the feather on the body was 2, .8 in radius, the centre was located at (-.1 -.4 -.4) and was rotated 0 degrees in the x-axis, 90 degrees in the y-axis, and 90 degrees in the z-axis. There were a total of 10,100 scattered feathers copied on to the body geometry of the model to cover the whole area.

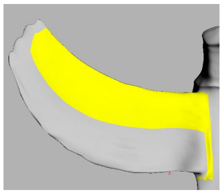


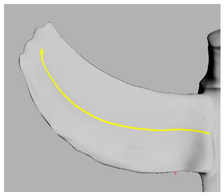
Pigeons have different types of feathers on their neck. The feathers on the neck were copied onto the area separate from the face and the body. These feathers were .8, .7 in radius with the centre located in coordinates (-.1 -.4 -.4), and rotated 0 degrees in the x-axis, 90 degrees in the y-axis, and 90 degrees in the z-axis. The total feathers on the neck area equal 5,700.

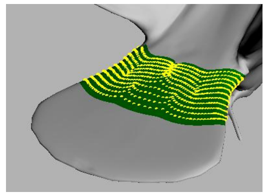
The grooming process involved in the interaction of feathers on the face, neck, and body was keeping track of where one group ends and the other begins. The overlapping of feathers on one area of the model with another area causes problems with shading. To avoid this, the model had to be groomed carefully around the areas where the two groups merge. The face area and the neck area merging caused the much larger neck feathers to overlap the face feathers, and cover either the beak of the pigeon or other areas of the face. To solve this problem, the neck and face border area had to be groomed carefully. The scattered area on the neck bordering the face had to be carefully regrouped so that the face feathers can then cover those areas without being overlapped by the neck feathers. The careful placement of feathers was essential for these problem areas.

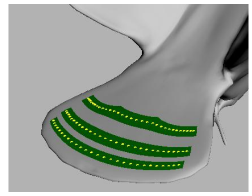
While the geometry being feathered required the scatter sop and the copy sop to copy the feathers onto the area, the careful placement of individual feathers was necessary for those places where two types of feathers meet. Regrouping of sections and placing the feathers of one group to cover the deleted section, helped make the merge between the two groups come together seamlessly. The greatest problem area occurred between the neck and the body of the pigeon model. The feathers of the body collided with the neck feathers around the area where the two groups meet. To solve this problem, several surface area points of the body near the neck had to be deleted and covered with the neck feathers. This careful and specific grooming allowed for the correction of collision between two separate areas.

The wings and the tail of the pigeon was groomed in a similar way but included several different techniques. Unlike the previous areas, where the feathers are basically smooth and small, the wing and the tail consist of different types of feathers. Therefore, the wings and the tail were further broken down into different sections. Each wing and the tail were broken down to two groups. The left and right wing was broken down to left upper wing, left lower wing, right upper wing and right lower wing. The tail was broken down to upper tail and lower tail.


Upper Wing


Lower Wing

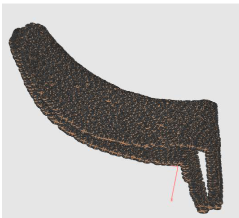

Upper Tail


Lower Tail

The upper areas of the wings and tail of the pigeon consisted of the same feathers as the face, neck and the body. The lower parts, however, consisted of the contour shaped feather modelled in Houdini by the curve sop. The default for the copy sop "copies [based on the] arranged normals to the template points (if they have a normal attribute)," however if other attributes are added, Houdini will "use them to control the orientation of the copies" (Houdini help). Grooming of the wings and the tail required the addition of a separate attribute, because the normal copy default for the geometry did not produce the desired results for placement of feathers on those areas. Without the additional attribute, scattered feathers on the upper wing and upper tail of the pigeon would copy on to the geometry facing in all different directions, creating a very messy look.

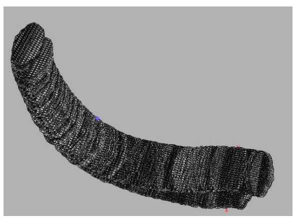
In order to solve this problem, a point sop was attached to the scatter sop. The point sop is used in Houdini to change the surface normal direction. Since the feathers were copied onto the normal of the surface and the original scattering did not produce the desired result, the point sop had to be added to calculate a new normal direction. The 'add normal' parameter of the point sop was used in order to calculate new normals. The equation $TX2-$TX was used to solve the problem of the feathers not facing the right direction on top of the geometry. What TX2-TX does is that it calculates the difference between transform x as it changes. The equation stands for subtract transform x1 from transform x2; to find the difference between the two positions. The recalculation of the normals allow for the feathers to stay copied on top of the surface of the wing and tail, face the right direction, and not change directions or positions as the geometry moves. As the wings flapped and the tail moved up and down, the feathers were able to stay in its place on the geometry.

For the upper wings, the circle sop was used. These feathers were then copied on to the



new calculated normals of the upper wing geometry. The feathers placed on the upper wings of the pigeon have radius 1.5, 1.1, is centred at (0, 0, 0), and is rotated 0 degrees in the x-axis, 90 degrees in the y-axis, and 90 degrees in the z-axis.

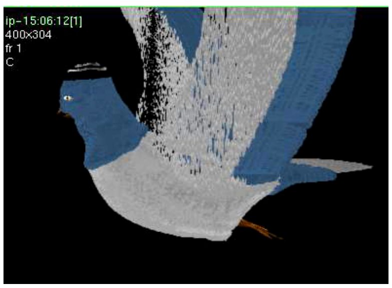
For the contour feathers used for the bottom half of the wings, the feathers were scattered



after being transformed. The feather for the bottom half of the right wing was translated to coordinates (3, 3, -1.2) and rotated 0 degrees in the x-axis, 0 degrees in the y-axis, and -100 degrees in the z-axis. The scale of the contour feather was (5, 6.5, 2). For the bottom half of the left wing of the pigeon the parameters was a little different. The feathers were translated to coordinates (3, -.8, -1) and rotated 0 degrees in the x-axis, 0 degrees in the y-axis, and -90 degrees in the z-axis. The scale of the feather was (5.5, 5, 1). These parameters allowed for the contour shaped feathers to line up in a row along the bottom halves of the wings. A wing like shape was created when the lower half of the wing merged with the feathers placed on the upper half of the wing, thus creating a realistic feathered wing of a bird.

Bottom Half of Wing merged with the Top Half

Several problems were encountered for the upper half of both wings with the direction of the normals and the feathers scattered on top. When the feather is set as being on the YZ plane and scattered on the upper part of the left and right wings, the feathers face the right direction and cover all of the geometry. However, this behaviour is only true when the viewer is looking at the pigeon from the side. Once the view is changed to the front of the pigeon, one is able to see many spacing between the feathers on the upper wings and can see right through the geometry.



Plane Parameter of Feather is set to ZX

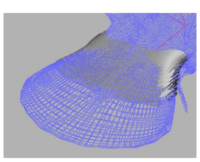
Plane Parameter of Feather is set to YZ

This problem is visually displeasing. In order to fix this, the plane specified for a feather had to be changed based on the direction the viewer is looking at the model. A plane is, “a fundamental two-dimensional object […] or a two dimensional space,” (Wikipedia 2006). To solve the problem, the plane parameter of the feather was changed to ZX when looking at the pigeon from the front and to YZ when looking at it from the side. To make the work area more user-friendly and more interactive, a switch sop was used to make this transition between planes easier. Two circle sops (feathers), one set to the ZX plane and one set to the YZ plane, were connected to a switch sop for both wings. The switch sop in Houdini is used for the switching between two different functions. The switch sop made it easier to switch between the two different parameters by copying the parameters of the left wing switch sop and pasting it on to the parameters of the right wing switch sop. When the pigeon is being viewed or rendered out from the front, the user simply has to change the parameters of a switch sop of just one wing and that will automatically change the parameters of the other wing.

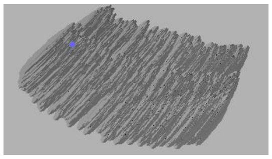
This fixed the problem of holes appearing between feathers on top of the wing geometry. So for the upper part of the wings, based on whether it is being viewed from the side or the front/back, the plane for the feather has to be switched from YZ to ZX. This makes it difficult to have camera movements from the side of the character to the front and rendering it all out together at once. The render has to be done by first rendering one angle by changing the parameters and then rendering again in a different angle. This, unfortunately, limits the number of camera angles and movements that can be done on the pigeon character.

The collision between the upper and lower wings' two different types of feathers were prevented by changing the translate parameters of the lower wing feathers. The translate coordinates had to be very specific so that the feathers on the bottom of the wing did not go on top of or underneath the upper wing feathers. In total, there were 5300 right upper wing feathers and 300 right lower wing contour shaped feathers, and there were 5100 left upper wing feathers and 300 left lower wing contour shaped feathers; which means 11,000 feathers on the wings alone.

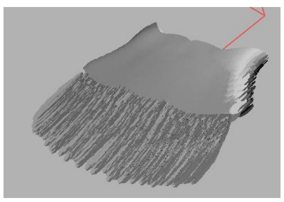
The tail of the pigeon was also divided into two parts. The upper tail of the pigeon was groomed differently then the rest of the body. Instead of using the scatter sop to scatter feathers at random points on top of the upper part of the tail, the point sop was used. The feathers were placed on the upper tail of the pigeon by placing a feather on each vertex point of the model.



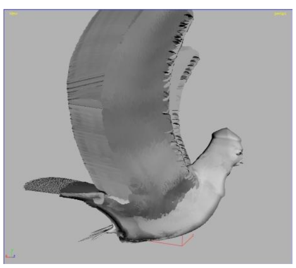
The upper part of the pigeon tail has a certain pattern and thus the feathers were placed in that pattern, giving the tail a unique design. The feather for the upper tail is set to be in the YZ plane, has a 1, .5 radius, its centre coordinates are (0, -.8, 1), and it was rotated 90 degrees in the x-axis, 90 degrees in the y-axis, and 90 degrees in the z-axis. The total amount of feathers on the upper tail equals 400.



The contour shaped feather was used to scatter on the bottom part of the tail. The point sop was attached to the scatter sop of the bottom tail group in order to recalculate the normals for the correct placement of the feathers. The twist sop was also used for the feathers at the bottom part of the tail. The twist sop was used to give the tail feather a slight bend so to give it a more feather-like behaviour. The twist sop's parameters were set so that its primary axis is z and secondary axis is y, the pivot was set to (0, 0, 0), and the strength of the bend was set to 7 with the rollof (attenuation of deformation) equalling 1. The contour shaped feather for the lower tail was translated to coordinates (1, -.4, -.4), rotated 0 degrees in the x-axis, 0 degrees in the y-axis, and 110 degrees in the z-axis. The scale of the feather was set equal to (2, 3, 2). The amount of feathers on the bottom half of the tail equals 300.



Upper and Lower Part of Tail Merged



In the end, the total number of feathers on Wild Pigeon added up to roughly 40,000. The grooming of the feathers was all based on specific areas. Once the feather geometry were placed on the model geometry, the geometry underneath were deleted leaving only the feathers. The grooming process took time for testing in making sure there were no holes in between the feathers, no collisions between feathers from different areas, and no movement of a feather to another location as the geometry moved. Once the testing of the feathers was finished, the next step was to apply shaders.

c) Shading

The first step in shading the feathers for the pigeon in Houdini was to collect all of the separate feather groups and shading them according to area. The plan was to shade the face and the bottom halves of the wings and tail a similar colour, to shade the body and the upper part of the wings a similar colour, and shading the neck of the pigeon a separate brighter colour. The feathers of the pigeon were grouped according to the colours being assigned to them.

The RenderMan (RMan) shaders provided in Houdini were used to shade the hat, eyes, nose, beak, and feet of the pigeon. RMan stone shader provided by Houdini was used to shade the feet, nose and beak of the pigeon. The RMan cloth displacement was used for those areas as well. For the pupil, eye, iris, and the black and white areas of the hat, the RMan painted shader was used. To give the hat a more cloth like resemblance, the RMan cloth displacement was used with parameters of frequency equal to 100 and depth equal to .015.

For the feathers of the pigeon, a custom shader was used. The custom RenderMan shader titled feather_fur.sl written by Mark Newport (2005) was imported into Houdini to be used as the shader for the feathers of Wild Pigeon. The custom shader was used in Houdini by first enabling Houdini to view RenderMan shaders. This was done by sourcing the Houdini setup bash and enabling RenderMan shader viewer. The custom shader was loaded into Houdini by typing in rhGo8 feather_fur.sl at the prompt and then clicking on refresh operator type libraries in Houdini.

After being loaded in, the custom shader was ready to be used in the SHOP level. The feather_fur shader works by passing in the base colour and surface normal of the feather to capture specular properties. The feather colour is more prominent towards the tip of the feather. The specular illumination model written by James T. Kajiya and Timothy L. Kay (1989) was used for the specular highlights calculated by the shader. The parameters of the custom shader were changed for each area of the pigeon where the feathers were in different colour. Blue and white was the base colours used for the pigeon feathers and thus were the colours assigned to the feather_fur shader. The parameters that are included in the feather_fur shader are end specular, specular size fade, illumination width, var_fade start, var_fade end, root colour, tip colour, and specular colour. All these parameter values differ for various feathered areas of the model.

Lighting

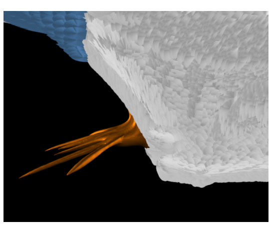
The lighting of the scene for rendering out the feathered Wild Pigeon character is important since the specular highlights of the feathers depend on it. If a light is too close or too bright on the feathers, the rendered feather will look too shiny or too sparkly, unlike a real feather. The basic three point lighting system was used to light the scene before rendering out the character. The three point lighting system consists of a key light, a fill light, and a back light (Wikipedia 2006). The key light is usually pointed at the object from the front and controls the overall lighting outcome (Wikipedia 2006). The fill light points at the object from the side and usually is a less brighter light then the key light, and the back light points at the back of the object and is the dimmest light of the three (Wikipedia 2006). The lighting for Wild Pigeon consists of a light pointing from the back, the side, and from the top of the character. The interaction between the light in the scene and the feathers were important in producing the correct colour and specular highlights and creating a realistic feather-like behaviour.

Rendering RIB files vs. Rendering out as Tiffs

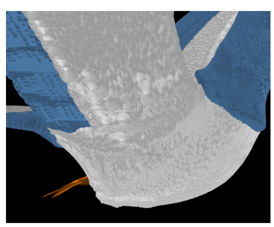
Two options were available in rendering out Wild Pigeon, and the options were rendering out each frame as a RIB file and then rendering it to tiff images in RenderFarm, or rendering out straight to tiff images directly out of Houdini. To determine which process was the most convenient, both were tested. It was found that to create one RIB file for one frame took 45 seconds, while rendering out straight to tiff images from Houdini took 1 minute 40 seconds per frame. Although rendering out as RIB files took less time, rendering the RIB files in RenderFarm afterwards took a little longer. The result was that both options produced the images in about similar time frames. Wild Pigeon was rendered out straight to tiff images from Houdini.

Compositing

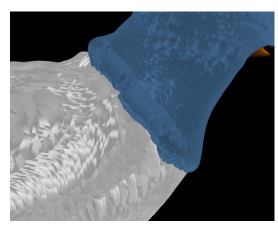
To best portray the details of the feathering on Wild Pigeon, six different camera angles were set up, and these different shots were put together to make a roughly 30 second animation showing off the feather detail. The six camera angles included a close up of the feet of the pigeon, the wing, the neck and wing joint area, a complete back view, a side view, and a front view. These six shots were composited together, in that order, to produce a short animation showing the complete feathering and shading of the character.



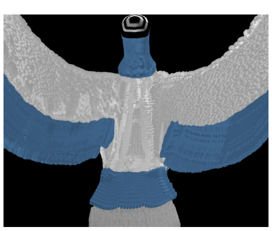
Shot 1



Shot 2

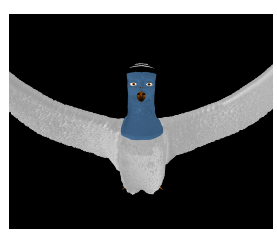


Shot 3



Shot 4

Shot 5



Shot 6

Future Additions

There are numerous additions that can be done in the feathering of Wild Pigeon to make it an even better process. A solution can be found to the problem of the upper wing feathers of the pigeon. The constant change between the YZ plane and the ZX plane can be solved, so that the user does not have to use the switch sop when viewing the pigeon from a certain angle. The entire network at the sop level of Houdini can be changed to a digital asset so that it can be used for more then just one character. The feather type can be changed from geometry copied onto geometry, to using Bezier curves or lines to model the rachis and barbs of a feather instead. The process can be updated and used as a tool for feathering all types of birds.

Conclusion

The objective of this project was to place feathers on a pigeon character modelled and animated previously in Maya. After research on different production companies and how they created CG feathers, it was decided that the feathering of the character Wild Pigeon will be done in Houdini. By using a simple mel script, the animated model was exported out of Maya into Houdini. The feathering process of the character then consisted of three steps: 1) grouping 2) grooming and 3) shading. The grouping process consisted of two large groups: areas with no feathers and areas with feathers. The grooming process included the further breaking down of different body parts of the character, and placing feathers on each area by transforming the feathers specifically for that area of the body. The grooming process required the calculating of surface normals, the editing and transforming of the feather types, the re-calculations of surface normals, collision and overlaps testing, the placement of feathers based on random scatter points, and placement by following the original model vertex points. The shading process included the loading of a custom feather shader along with using Houdini provided RMan shaders. Lighting was also necessary to interact with the shader to produce a desired feathered effect. The completed feathered pigeon was then rendered out directly to tiffs rather then rendered out as RIB files. The final 30 second animation consists of different shots revealing six close-up angles of the pigeon’s different feathered sections. The feathering of Wild Pigeon in Houdini was a success and produced the desired stylized look aimed for.

References

ANONYMOUS, 2005. *Maya2LW.zip* [online]. Available from: http://www.lunaseven.com/images/Maya2LW.zip [Accessed 8 August 2006]

ANONYMOUS, 2005. *Vanguard achieves new levels of realism in valiant using side effects software's Houdini for advanced feathering pipeline* [online]. Available from: http://www.cgchannel.com/news/viewfeature.jsp?newsid=4260 [Accessed 27 July 2006].

CAMPBELL, J., 2004. *Maya To Houdini Camera Matching* [online]. Available from: http://studentpages.scad.edu/%7Elcampb22/tutorials/cam_matching/maya_moving_geo.html [Accessed 8 August 2006]

CROMWELL, B., 2004. *3-D Data Collection, Analysis, and Applications -- Calculating Surface Normals and Curvatures* [online]. Available from: http://www.cromwell-intl.com/3d/normals.html [Accessed 8 August 2006].

FRAMESTORE, 2006. *Framestore CFC just wild about Harry* [online]. Available from: http://www.framestore-cfc.com/feature/harry_potter/index.html [Accessed 27 July 2006].

FRANCO, C., Marcelo Walter, 2001. *Modelling the Structure of Feathers* [online]. Available from: csdl.computer.org/comp/proceedings/sibgrapi/2001/1330/00/13300381.pdf [Accessed 27 July 2006].

HOUDINI HELP. *Side Effects Software.* 2006.

Image: ANONYMOUS, 2005. *Valiant Characters* [online]. Available from: http://movies.apple.com/trailers/disney/valiant/assets/characters.jpg [Accessed 27 July 2006].

Images: ANONYMOUS, 2002. *Contour and Semiplume Feathers* [online]. Available from: http://www.arctic.uoguelph.ca/cpl/organisms/birds/biology/morphphys/page8.htm [Accessed 27 July 2006].

Image: WIKIPEDIA, 2006. *Fawkes* [online]. Available from: http://upload.wikimedia.org/wikipedia/en/7/70/Fawkes_screenshot_from_Chamber_of_Secrets.jpg [Accessed 27 July 2006].

KAJIYA, J., and Timothy L. Kay, 1989. Rendering Fur with Three Dimensional Textures. *Computer Graphics*, 23 (3), 271-280.

MOLTENBREY, K., 2005. *A Wing and a Prayer* [online]. Available from: http://cgw.pennnet.com/articles/article_display.cfm?article_id=235683 [Accessed 27 July 2006].

NEWPORT, M., 2005. *Start to Finish Feathers Solution* [online]. Thesis (MSc). Bournemouth University. Available from: ncca.bournemouth.ac.uk/jmacey/Msc05/mnewport_MastersThesis.pdf [Accessed 27 July 2006].

STREIT, L., W. Heidrich, 2002. *A Biologically-Parameterized Feather Model* [online]. Thesis. The University of British Columbia. Available from: www.cs.ubc.ca/~heidrich/Papers/EG.02.pdf [Accessed 27 July 2006].

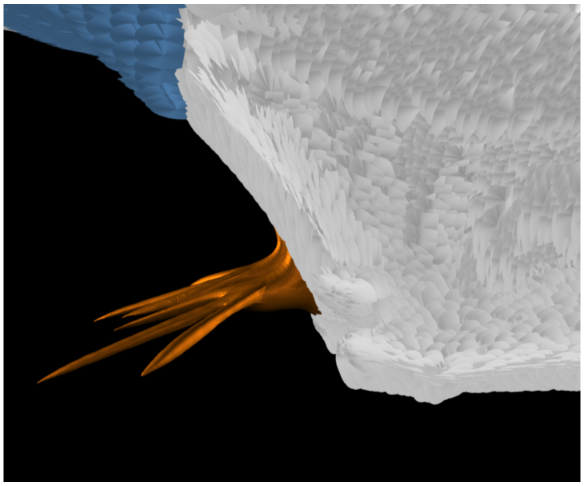
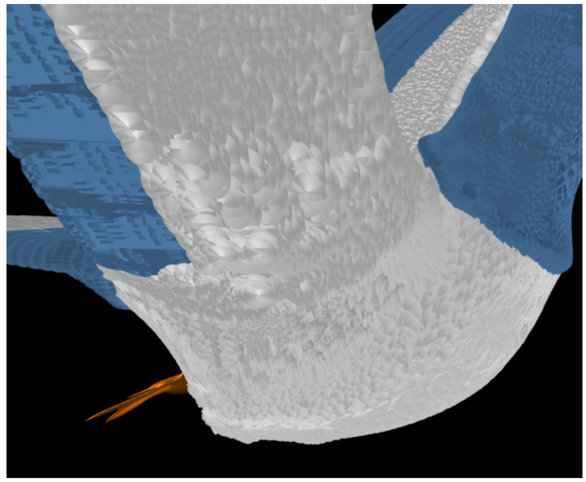
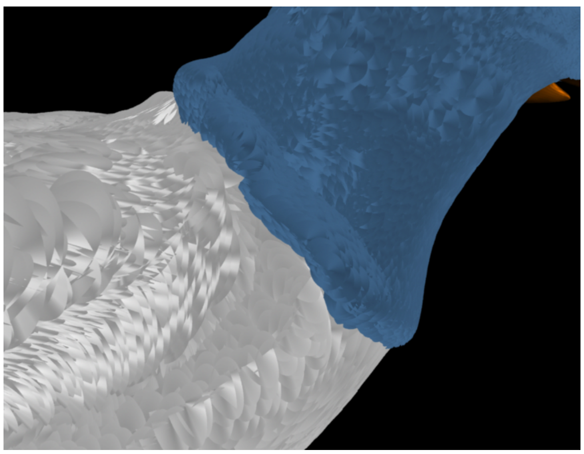
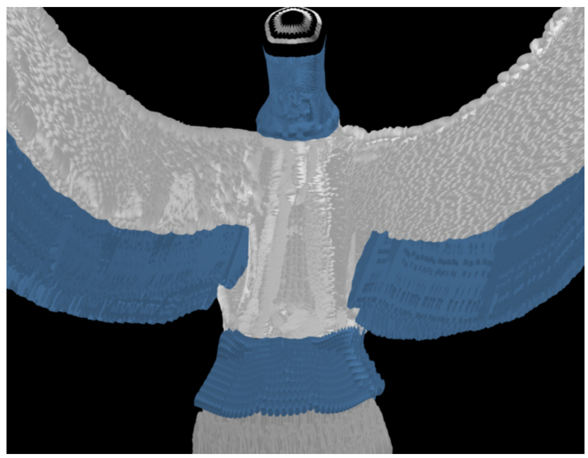
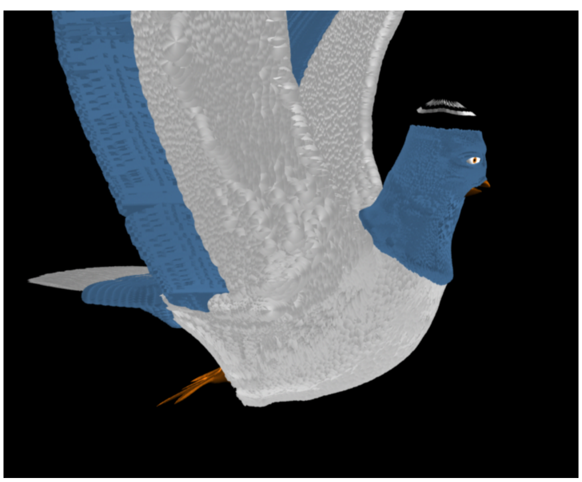
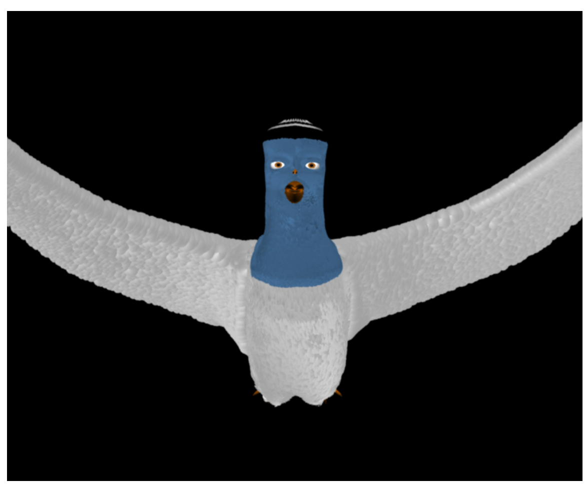
STREIT, L., W. Heidrich, 2002. *Generating Feather Coats Using Bezier Curves* [online]. Thesis. The University of British Columbia. Available from: pages.cpsc.ucalgary.ca/~streitl/papers/SIG2002Sketch.pdf [Accessed 27 July 2006]

WIKIPEDIA, 2006. *Plane (mathematics)* [online] Available from: http://en.wikipedia.org/wiki/Plane_%28mathematics%29 [Accessed 8 August 2006]

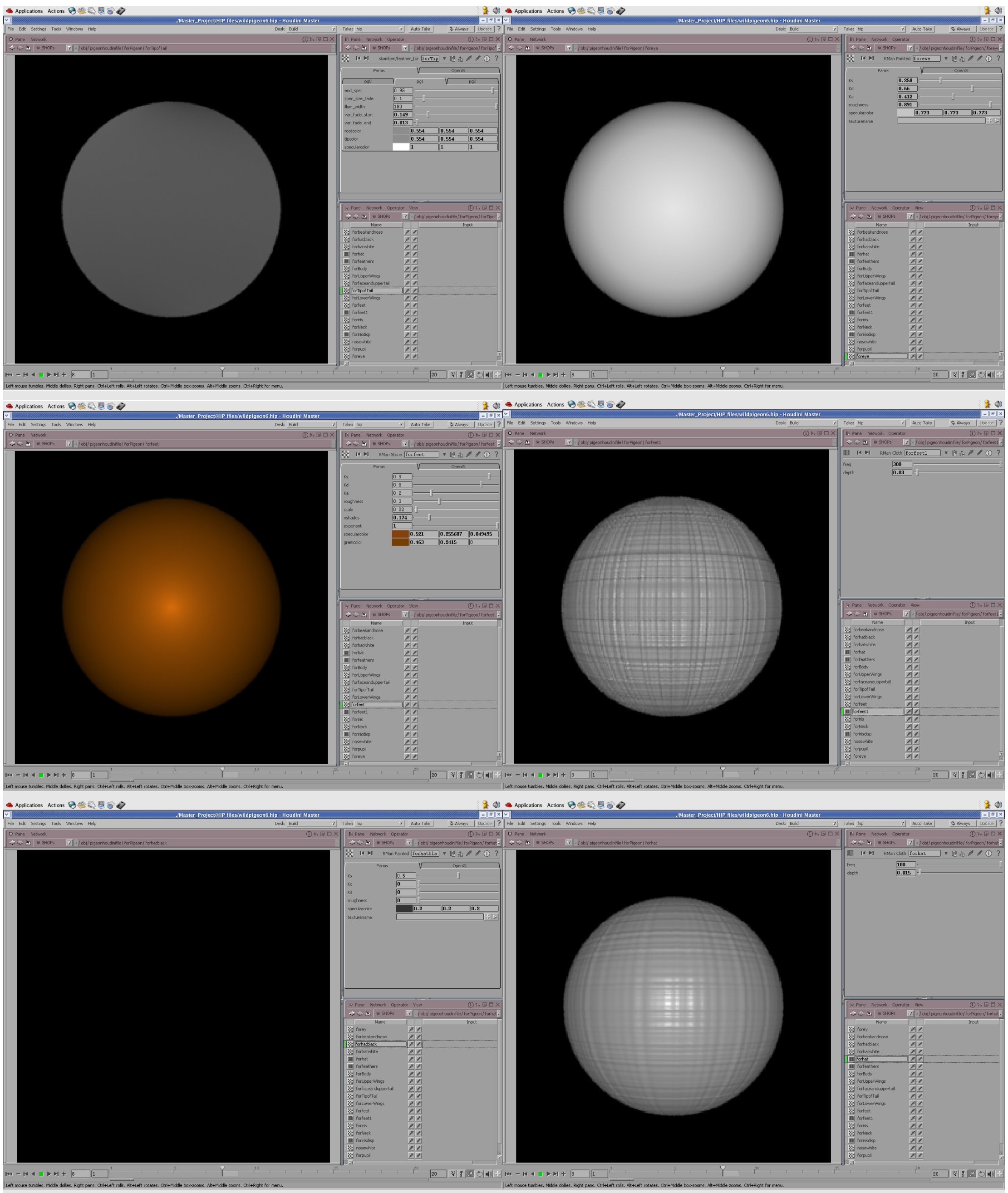
WIKIPEDIA, 2006. *Surface Normals* [online]. Available from: http://en.wikipedia.org/wiki/Surface_normals [Accessed 8 August 2006]

WIKIPEDIA, 2006. *Three-point lighting* [online]. Available from: http://en.wikipedia.org/wiki/4-point_lighting_setup [Accessed 8 August 2006]

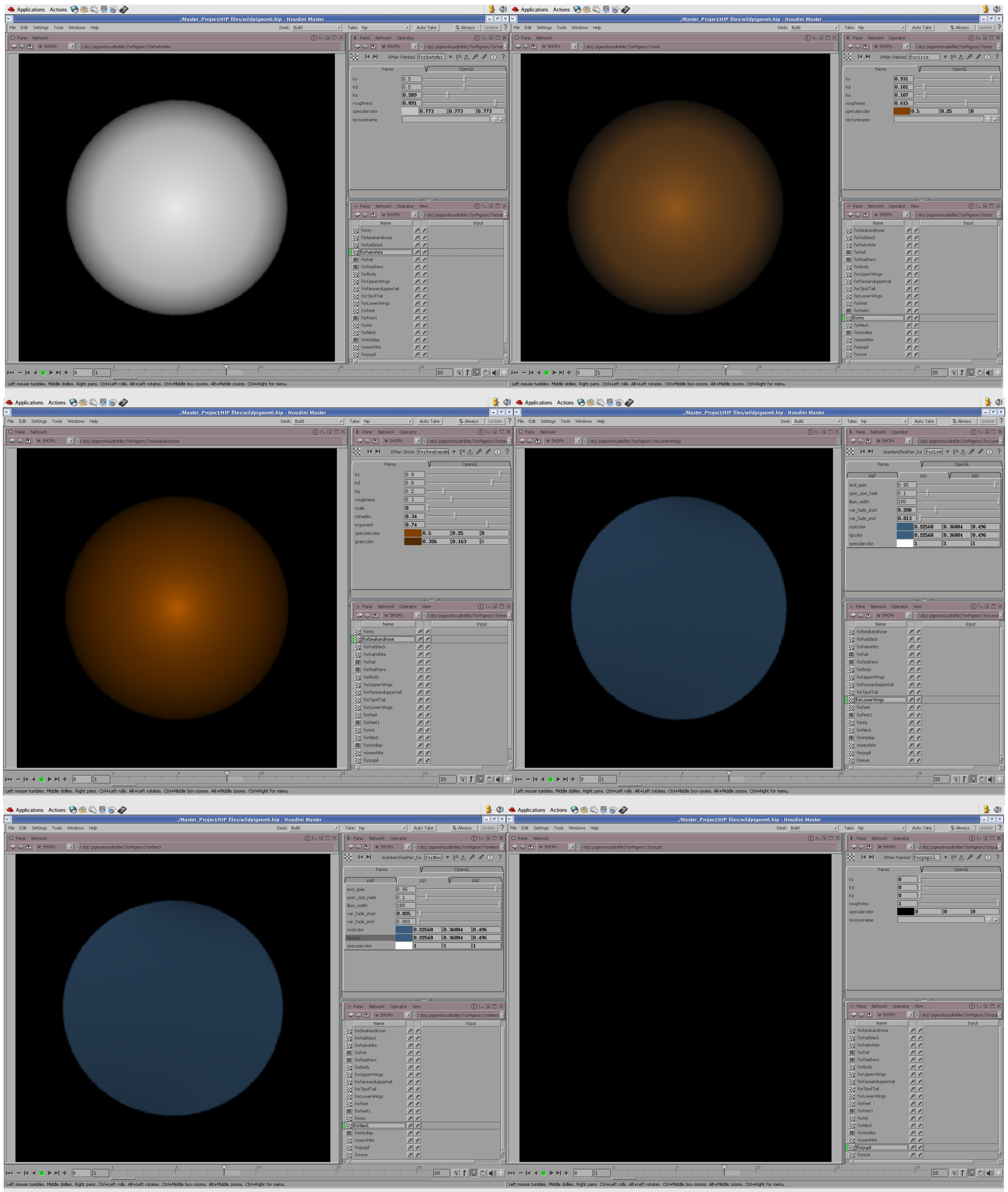
Appendix I – Final Stills

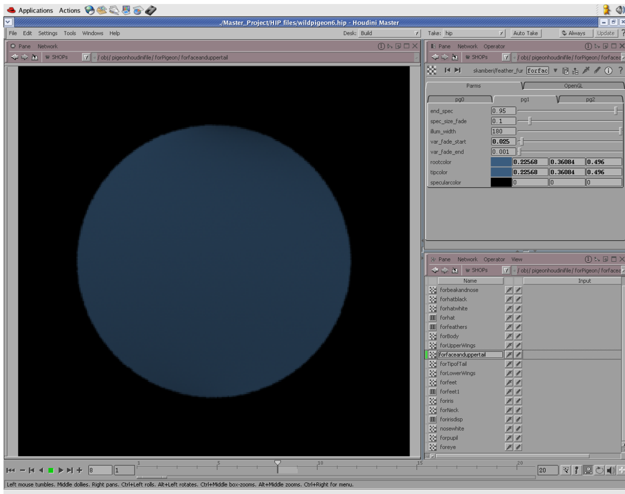
Appendix II – Shaders/Dislacements

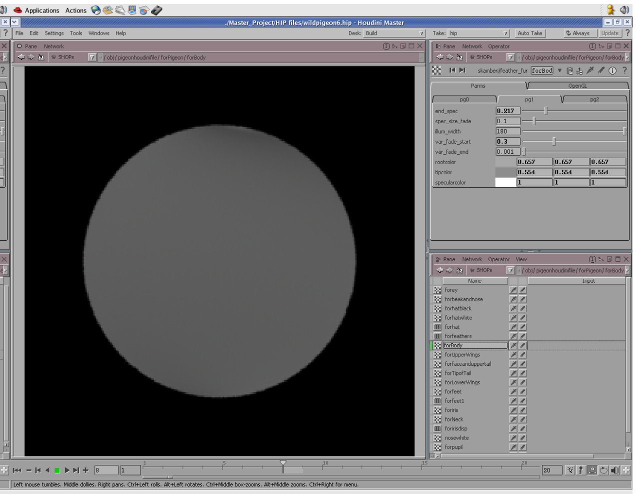


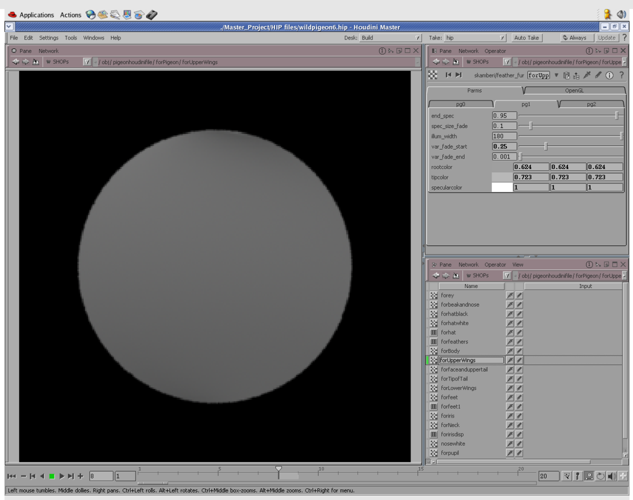
Appendix II – Shaders/Displacements



Appendix II - Shaders/Displacements







Appendix III - Houdini Network