Modeling and Animation of Human Hair in Strips

Chuan Koon Koh and Zhiyong Huang
{kohchuan | huangzy}@comp.nus.edu.sg
Department of Computer Science, School of Computing
National University of Singapore, Lower Kent Ridge Road, Singapore 119260

ABSTRACT

We propose a novel framework of human hair modeling and animation based on the modeling of the hair strips. Each hair strip, modeled by one patch of parametric surfaces, represents a group of hair strands. Keyframing is implemented. The parametric representation of hair strips can handle a deformation of any complexity and still appear smooth. Finally, we use polygon tessellation, texture mapping with the Alpha map for the rendering.

1 INTRODUCTION

In recent years, we have seen significant advances being made in 3D computer graphics technology, software APIs such as Microsoft's Direct3D and Sun's Java3D, as well as in the consumer 3D acceleration hardware. Thus there is a need to review the existing techniques of modeling and animation of hair.

Various methods have been employed to model and animate human hair, including methods using hair strands individually [1, 4], trigonal prisms with 2D hair distribution maps [3, 9] and more recently, volumetric models [6, 7]. These approaches concentrate mainly on modeling hair accurately, and often require specialized rendering algorithms. As such, hardware acceleration is unlikely to be available for the above approaches, making them more suitable for off-line graphics system.

We propose a different but simple framework of hair modeling and animation that is suitable for real-time applications. The main idea is to model and animate hair in strips. Each hair strip, modeled by one patch of parametric surfaces in particular NURBS, represents a group of hair strands. A variety of shapes may be defined for each strip. For the rendering, we use polygon tessellation, texture mapping with the Alpha map.

We took all advantages of parametric surface such as its accurate and compact representation, support of global and local shape editing, and inherent continuity.

Another advantage of this framework is that the level-of-detail is scalable by applying multiresolution techniques. When the viewpoint is far, a courser tessellation is used. When getting closer, it can be tessellated more finely. Similarly, we can use different resolution of images for texture mapping. The established technique of MIP Mapping is also applied [10].

The use of texture mapping, as well as a large number of polygons tessellated from parametric surfaces, makes good use of current development of software and hardware acceleration technology [5, 8].

Animation is achieved by displacement of the control points of hair strips. The parametric representation of hair strips can handle a deformation of any complexity and still appear smooth. Existing techniques can be applied to hair animation such as the physically based modeling of clothes [2, 11]. The current experiments are done using keyframing.

2 RELATED WORK

In human animation, hair modeling and animation are very challenging for the large number of hairs and the complex interaction of light, shadow casting among them, and more difficult, collision detection and response among hair [4].

Hair modeling is fundamentally important for the animation. A well-known framework is *HairStyler*, an interactive framework to model hair segments [4, 12]. First, the three-dimensional curved cylinder is composed of straight cylindrical segments connected by points. Second, the in-between points can be interactively edited. Third, each segment is assigned to each triangle in the scalp mesh. The orientation of hair segments can be adjusted by rotating around the normal of its scalp triangle. Other modeling methods include trigonal prisms with 2D hair distribution maps [3, 9], volumetric models [6, 7], single triangle laid out on the surface [14], a pyramid of triangles for one strand [15], connected segments of triangular prisms, and volume densities controlled with pseudo-random functions [16].

Almost of all hair animation work is based on different Physics models. Two well-known approaches include a method using one-dimensional projective differential equations and pseudo-force fields [1, 4] and a method using mass spring model [13].

Our work is different in that we model and animate hair in strips. Each hair strip is modeled by one patch of parametric surfaces in particular, NURBS [18]. The animation is achieved by displacement of the control points. We describe in detail from the next section.

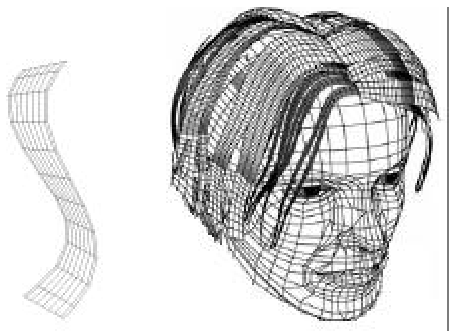
3 OUR WORK

There are basically four problems to produce realistic animated synthetic actors with hair: hair modeling and creation, hair motion, collision detection and response, and hair rendering [4]. We describe our framework for a solution of these basic problems.

3.1 Hair Modeling

We model hair in strips. The motivation is to reduce the large number of geometric objects when each hair strand is individually represented. Each hair strip is a group of hair strands in the shape of thin flat patch (Figure 1 (a)). Thus, all the hair strands are represented in layers of strips overlaying each other on top of the scalp (Figure 1 (b)).

There are four major geometric schemes for geometric modeling: polygon meshes, parametric surfaces, implicit surfaces, and subdivision surfaces. We considered each of them in order to represent the hair strips geographically.



(a) One hair strip (b) All the hair strips overlaying on the scalp

Figure 1. Hair modeling in strips

Polygon mesh scheme is most popular in 3D graphics for its simplicity. Moreover, the direct rendering support from the graphics hardware. However, its discrete nature makes difficult to model and animate the object globally.

Tensor product non-uniform rational B-splines (NURBS) have become the standard representation for complex smoothly varying surfaces [18]. However, a major drawback of NURBS is the requirement that control nets consist of a regular rectangular grid of control points.

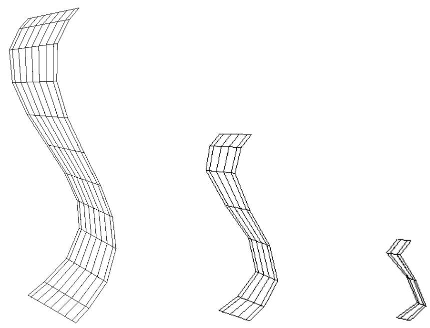
Implicit surfaces are typically defined by the isosurfaces derived from implicit functions in the form of the sum, minimum, or maximum of the simpler functions such as spherical function. It is excellently used to model and deform human body [20]. However, to define shapes with sharp edges is very difficult. Rendering is another computational expensive problem.

Subdivision surfaces are defined as the limit of an infinite refinement process [19]. Although subdivision surfaces have been known for nearly fifteen years, their use has been hindered by the lack of a closed form -- they are defined only as the limit of an infinite procedure. The recent work on subdivision surfaces is very active and results are used in the modeling and animation of human body [21].

We model the hair strips using NURBS. We took all advantages of parametric surface such as its accurate and compact representation, support of global and local shape editing, and inherent continuity. We noticed the major drawback of NURBS that control nets consist of a regular rectangular grid of control points. However, as the hair strips take the rectangular shape, we can avoid the drawback.

There is a direct mapping between the parametric space and the texture space using NURBS. We will discuss more in the subsection of rendering.

We tessellate the NURBS representation into polygon mesh for the final rendering. The Oslo's algorithm is implemented using the multiple knot insertion for the tessellation [17, 25]. If more than a few knots are being inserted at once, the Oslo algorithm is more efficient than the Böhm algorithm [24]. The different levels of resolution of tessellation can be automatically selected according to the distance to the viewpoint. When the viewpoint is far, a courser tessellation is used, and vice versa (Figure 2).



(a) 88 polygons (b) 35 polygons (c) 12 polygons

Figure 2. Tessellation of NURBS representation of a hair strip with different resolution

For hair modeling, interactive software is implanted with a GUI for the user to define the hair strips. Each hair strip can be duplicated constantly or differently with changing some parameters such as the location, orientation, and weights for knots. The resulted strips represent one group of hair on the scalp. An important function is to attach groups of strips to the triangles of the scalp mesh. The final modeling result can be saved for further use.

The modeler is object-oriented and implemented in Java 1.2 and Java3D [22]. The root of the scene-graph is a *Virtual Universe* node, which holds all the objects. A *Locale* is attached to the root to get the viewpoint. A *BranchGroup(BG)* node is a container for other nodes, such as *TransformGroups(TG), Behaviours, Lights, Shapes* etc. A *Behavior* node provides interaction between the user and the scene graph. A *TransformGroup* node performs a transformation on all its children. A *Shapes* node represents the geometry and appearance of an object, which is one leaf node. The *Hair BranchGroup* holds all of the hair strips created. The *Spline BranchGroup* holds the shapes currently being modified. The *Head BranchGroup* is used to hold a bald scalp where the hair strips attach (Figure 3).

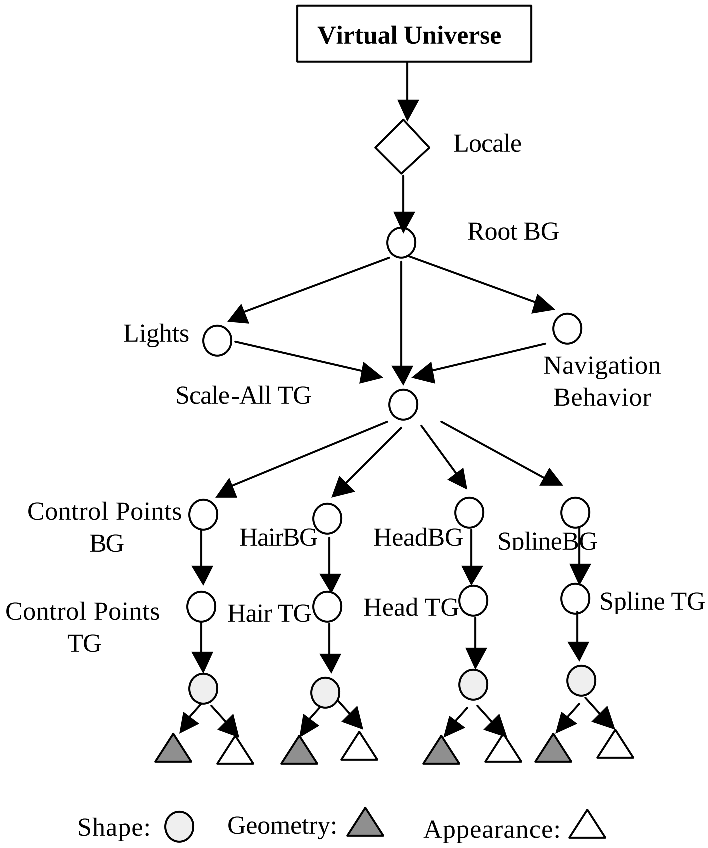


Figure 3. Scene graph of the hair modeler

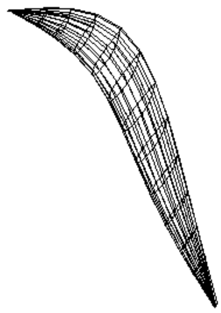
3.2 Hair Animation

The hair animation is achieved by displacement of the control points of NURBS patch representing the hair strips. The most significant advantage the parametric representation enjoys over the polygon meshes lies in the fact that it can handle a deformation of any complexity and still appear smooth. This is because in altering the positions of the control points we are merely changing the coefficients of the basis functions. The deformed parametric surface is therefore in no sense less well defined than its un-deformed counterpart.

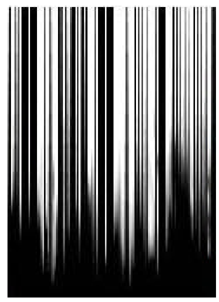
We have implemented the keyframing animation for the current system. A GUI is provided for the user to interactively define the key frames. The user can displace the control points of strips for the whole group or individually for one strip. A few snapshots in color are shown in the section of results.

Modeling hair in strips is closer to the scheme of cloth modeling than the scheme of hair modeling in individual strands. So the animation methods of cloth can be applied [2, 11]. We need to extend the animation framework from the polygon meshes to the parametric surfaces.

3.3 Rendering



(a) One strip represented by polygon mesh after tessellation



(b) The Alpha map of the hair

(c) The strip after texture mapping with the Alpha map

Figure 5. Hair rendering of one strip

Texture mapping is very important for the visual effect. It becomes more popular with the advent of more powerful and cheaper graphics hardware. Another significant advantage of the parametric representation is that there is a direct mapping between the parametric *u-v* space and image *s-t* space.

We also can take the rendering advantage of the polygon meshes by tessellation of the parametric representation. As described in the section of hair modeling, we tessellate each hair strip with different resolution. For each resulted vertex (x, y, z) of the polygon mesh, its parametric coordinate (u, v) is already known. Thus, the coordinate (s, t) of the texture space can be derived easily.

We use the image with the Alpha channel (Alpha map) for texture mapping. In an Alpha map, the color intensity of each pixel is defined by (r, g, b, a), where $r, g,$ and b are components for red, green, and blue components as the normal texture map, but different from the normal texture map, there is the fourth component a, the Alpha-value. The Alpha-value is used to represent the degree of transparency in computer graphics [23]. It represent total opaque if $a=1$, total transparent if $a=0$, and translucent if $0<a<1$. The image is called Alpha map if not all pixels with the value 1 for the Alpha-value.

We use different resolution of images and Alpha map for texture mapping in order to achieve real time performance. The established technique of MIP Mapping is applied [10].

One rendering result of one hair strip is shown in Figure 5 using the Alpha map. More results are shown in the section of the results.

4 IMPLEMENTATION

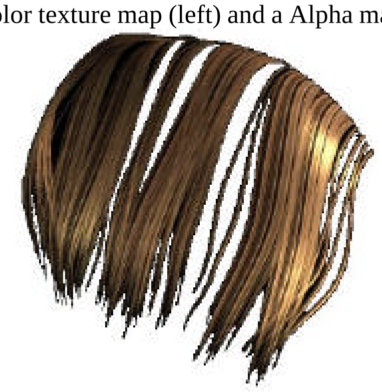
Our framework is object-oriented. The Java3D 1.1.3 API is used for the implementation. Each hair strip is implemented as one object, which contains information of its geometry and appearance. In addition, the hair strip object contains methods to re-tessellate and deform its geometry to generate new key-frames. It also has listeners that wake up on events to trigger animation and selection of different level-of-detail for tessellation. The object-oriented model improves interactivity.

5 RESULTS

In this section, we show more results of our hair modeling and animation framework. There are the rendering result (Figure 6), Hair strips attached with scalp (Figure 7), and finally, the snapshots of keyframing (Figure 8).

(a) A group of Hair Strips in Gouraud Shading

(b) A color texture map (left) and a Alpha map (right)



(c) The rendering result by texture mapping using the color texture map and Alpha map

Figure 6. The rendering result of a group of hair strips

Figure 7. Hair strips attached to the scalp model

(a) Key frame 1

(b) Interpolation resulted frame 1

(c) Interpolation resulted frame 2

(d) Key frame 2

Figure 8. Snapshots of hair animation

6 CONCLUSION

We have presented a novel framework of human hair modeling and animation based on the modeling of the hair strips. Each hair strip, modeled by one patch of parametric surfaces, represents a group of hair strands. Keyframing is implemented. Finally, we use polygon tessellation, texture mapping with the Alpha map for the rendering. The results are encouraging.

Our framework is applicable to the modeling and the animation of tree leaves and grass.

We are working on the animation by extending the cloth animation framework [2, 11].

7 REFERENCES

[1] K. Anjyo, Y. Usami, T. Kurihura. A Simple Method For Extracting The Natural Beauty Of Hair, SIGGRAPH(92), pp. 111-120.

[2] D. Baraff and A. Witkin. Large Steps in Cloth Simulation, SIGGRAPH(98), pp. 43-54.

[3] L. H. Chen, S. Saeyor, H. Dohi, and M. Ishizuka. A System of 3D Hair Style Synthesis Based on the Wisp Model, The Visual Computer, 15 (4), pp. 159-170 (1999).

[4] A. Daldegan, T. Kurihara, N. Magnenat Thalmann, and D. Thalmann. An Integrated System for Modeling, Animating and Rendering Hair, Proc. Eurographics '93, Computer Graphics Forum, Vol.12, No3, pp.211-221.

[5] H. Hoppe. View-dependent Refinement of Progressive Meshes, SIGGRAPH(97), pp. 189-198.

[6] W. Kong and M. Nakajima. Visible Volume Buffer for Efficient Hair Expression and Shadow Generation, Computer Animation '99, (May 1999, Geneva, Switzerland). IEEE Computer Society.

[7] F. Neyret. Modeling, Animating, and Rendering Complex Scenes Using Volumetric Textures, IEEE Transactions on Visualization and Computer Graphics, 4(1), pp. 55-70 (January-March 1998).

[8] C. C. Tanner, C. J. Migdal, and M. T. Jones. The Clipmap: A Virtual Mipmap, SIGGRAPH(98), pp. 151-158.

[9] Y. Watanabe and Y. Suenaga. A Trigonal Prism-Based Method For Hair Image Generation, CGA(12), No. 1, 1992, pp. 47-53.

[10] L. Williams. Pyramidal Parametrics, SIGGRAPH(83), pp. 1-11.

[11] P. Volino, N. Magnenat Thalmann, and N. Magnenat Thalmann. Versatile and Efficient Techniques for Simulating Cloth and other Deformable Objects. SIGGRAPH (95), pp. 137-144.

[12] N. Magnenat Thalmann and A. Daldegan. Creating Virtual Fur and Hair Styles for Synthetic Actors. In Communicating with Virtual Worlds, Springer-Verlag, Tokyo, pp. 358-370.

[13] R. E. Rosenblum, W E. Carlson, I. E. Tripp. Simulating the Structure and Dynamics of Human Hair: Modeling, Rendering and Animation. The Journal of Visualization and Computer Animation, 2 (4), pp. 141-148.

[14] J. T. Kajiya and T. L. Kay. Rendering Fur with Three Dimensional Textures, SIGGRAPH (89), pp. 271-280.

[15] G. Miller. From Wire-Frame to Furry Animals, Graphics Interface (88), pp. 138-146.

[16] K. H. Perlin. Hypertexture, SIGGRAPH (89), pp. 253-262.

[17] E. Cohen, T. Lyche, and R. Risenfeld. Discrete B-Splines and Subdivision Technique in Computer-Aided Geometric Design and Computer Graphics, CGIP, 14 (2), 1980, pp. 87-111.

[18] A. R. Forrest. The Twisted Cubic Curve: A Computer-Aided Geometric Design Approach. Journal of Computer Aided Design, 12 (10), pp. 350-355, 1980.

[19] E. Catmull and J. Clark. Recursively Generated B-spline Surfaces on Arbitrary Topological Meshes. Journal of Computer Aided Design, 10 (6), pp. 350-355, 1978.

[20] J. Shen and D. Thalmann. Interactive Shape Design Using Metaballs and Splines, Implicit Surfaces (95), pp. 187-196.

[21] T. DeRose, M. Kass, and T Truong. Subdivision Surfaces in Character Animation, SIGGRAPH (98), pp. 85-94.

[22] Sun Microsystems. Java3D API documentation and Tutorial, http://java.sun.com, 2000

[23] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, Computer Graphics, Principles and Practice, Addison-Wesley Publishing Company, Inc. 1996, pp. 835-840.

[24] W. Böhm. Insert New Knots into B-spline Curves, Journal of Computer Aided Design, 12 (4), pp. 199-201, 1980.

[25] A. Meyer. A Linear Time Oslo Algorithm, TOG (10), 1991, pp. 312-318.