Modeling Hair Movement with Mass-Springs

Anna Sokol
ansokol@cs.sunysb.edu
Computer Science Department
SUNY Stony Brook

Abstract: *This paper is presenting a framework for modeling hair movement using mass-springs. This framework will support dynamic simulation, movement, and graphical rendering of hair and simulate different types of hair from very curvy to straight, from very short to very long, from very thin to very thick, and from very dense to bolding. It will also simulate different affects forces have on hair from wind, gravity, and damping. Each individual strand of hair will be modeled as a twisted NURBS cylindrical surface with n control points.*

1 Introduction

Realistic hair simulation has been a huge problem due to its incredible complexity. For example, a human head can have over 100,000 individual strands. The simulation of long and/or curly hair creates even more complexity. Animating hair in real-time is a challenging problem due to the high number of primitives required to model hair accurately and realistically.[1]

While movies like Monsters, Inc. and Final Fantasy have been able to generate very realistic looking hair and fur, they weren't done anywhere near real time. Modeling, styling, simulating, and animating hair remains a slow, tedious, and often a painful process for animators.[1]

In this paper, I propose to model hair movement with mass-springs and to model hair with twisted NURBS cylindrical surfaces. The hair is rendered in almost real time. The more hair the slower the system works. The hair can look short or long, curly or straight, thin or thick, and messy or combed.

Each individual strand of hair will be modeled as a twisted NURBS cylindrical surface with n control points. One end of each hair strand will be fixed on the surface of the plane, torus, or sphere. Mass spring forces will manipulate the remaining parts of each hair strand. The amount of movement each hair strand has will be determined by the stiffness and rest length of each spring between control points and the mass of each control point, as well as the external forces.

2 Related Work

Some fundamental techniques were presented to model the motion of individual hair strands in [Anjyo et al. 1992; Kurihara et al. 1993; Daldegan et al. 1993], with each strand of hair represented as a series of connected line segments and the shape of the hair determined by specifying the desired angles between segments. Forces are applied to the control points of the line segments to simulate the hair motion.

To reduce the overall computation time, strands of hair that are near each other or move in a similar fashion, are bundled together as a group or as a wisp [Kurihara et al. 1993]. Using a similar philosophy, individual strands of hair are grouped together as "wisps" for animating long hair, each modeled using a spring-mass skeleton and a deformable envelope [Plante et al. 2001]. A similar approach is used for interactive hairstyling [Chen et al. 1999; Xu and Yang 2001]. Adaptive guide hairs were used in [Chang et al. 2002] to add more detail to overly interpolated regions. Using guide strands involves animating a few strands and the dynamics of the remaining strands are interpolated from these guides.[1]

An integrated system for modeling, animating and rendering hair is described in [Dald93]. It uses an interactive module called HairStyler [Thal93] to model the hair segments that represents the hairstyle.[2]

3 Technique

Mass springs were used to do hair animation. Twisted NURBS cylindrical surfaces were used to model individual hair strands. The hair was modeled on a plain, a sphere, and a torus. The rest angle range, the length of each partition, the number of hair strands, thickness of the hair strands, as well as, gravity, wind and damping were dynamically user selected. Also the user can decide if the hair should start of as being messy or combed.

3.1 Pipeline

In the pipeline of this system, first, the underlying shape is selected; second, the style (messy or combed), the thickness, length range, and the rest angle range each partition of the hair strand are selected; third, the springs and the particles for the mass-spring system are created; fourth, the external forces (gravity, wind, damping) are selected; fifth, the control points of the hair strands are extracted after each iteration of the mass-spring system; finally, the twisted NURBS cylindrical surfaces are build and rendered.

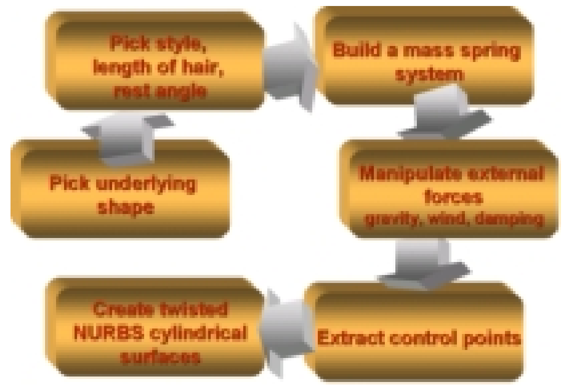

Figure 1: Pipeline

3.2 Underlying Shapes

As seen in Figure 1 the first step in the pipeline is the creation of the underlying shapes. In order to model hair on a surface one first needs a surface to model the hair on. The three underlying shapes that are used in this project are a torus, a sphere, and a plain. The torus and the sphere are modeled using NURBS surfaces. The plain is modeled using only polygonal openGL representation.

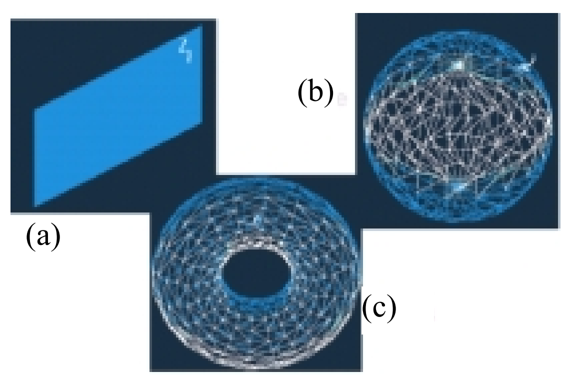

Figure 2: underlying shapes (a) plain, (b) sphere, and (c) torus

3.2 Style

The user can manipulate different aspect of the hair. The amount of hair seen can be manipulated. If the length range of the hair is manipulated the resulting hair strands would be shorter or longer. If the rest angle range is manipulated the resulting hair strands would be straighter or curvier. The manipulation of the thickness will result in thinner or thicker hair strands. Also messy or combed hair can be selected which will result in having the starting hair strands be randomized individually or all together respectively.

4 Algorithm

In this section the mass-spring system, the manipulation of external forces, the extraction of control points, and the actual creation of the twisted NURBS cylindrical surfaces will be discussed.

4.1 Mass-spring system

Now the mass spring system can actually be build. First the springs are created and then the particles. Each iteration of the mass-spring system is done with a modified Euler algorithm.

Only linear springs are used here. They are built based on the amount of hair seen; the rest length of the springs is based on the length range previously selected.

After the springs are built the actual particles are created. The particles are divided into *n* particles per hair strand, where *n*=8. The first particle of each hair strand is *fixed* on the surface. The rest of the particles in the hair strand are grown outward toward the direction of the normal of the underlying surface. The curliness and length of each of these particles are based on the length range and rest angle previously selected. The mass of each particle is set to 1. The force and velocity vectors of every particle are set to zero. The following is the actual algorithm of how the particles are built.

```
COPYV(particles[i*parts].p , surf[k]);
MAKEV(particles[i*parts].v, 0.0f, 0.0f, 0.0f)
MAKEV(particles[i*parts].f, 0.0f, 0.0f, 0.0f)
for (j=1..parts)
  MAKEV(particles[i*parts+j].v, 0.0f, 0.0f, 0.0f)
  MAKEV(particles[i*parts+j].f, 0.0f, 0.0f, 0.0f)
  if (nx==0) particles[i*parts+j].p.x=
        randomp(restang)+particles[i*parts].p.x;
  else particles[i*parts+j].p.x=
        -nx*randomp(leng)+particles[i*parts+j-1].p.x;
  if (ny==0) particles[i*parts+j].p.y =
        randomp(restang)+particles[i*parts].p.y;
  else particles[i*parts+j].p.y =
        -ny*randomp(leng)+particles[i*parts+j-1].p.y;
  if (nz==0) particles[i*parts+j].p.z =
        randomp(restang)+particles[i*parts].p.z;
  else particles[i*parts+j].p.z =
        -nz*randomp(leng)+particles[i*parts+j-1].p.z;
```

4.2 External forces

The user selects the external force, such as gravity, wind, and damping. The wind and the gravity change directions every ten iterations of the mass-spring system.

The following algorithm shows how the forces are actually added to the internal forces of each particle that isn't fixed on a surface. At first the force of each particle is set to 0 then if the particle isn't fixed on the surface the wind and gravity are added to it.

```
For (i=0 ... particles)
        MAKEV(particles[i].f, 0.0f, 0.0f, 0.0f)
        if (particles[i].fixed) continue;
        ADDV(particles[i].f, wind);
        particles[i].f.y +=gravity *particles[i].m;
```

4.3 Control points

Each individual strand of hair will be modeled as a twisted NURBS cylindrical surface with *n* control points, degree 2 and 20×20 subdivision. Each of these cylinders contains a certain thickness and a length. A circle determines the thickness of the cylindrical strand. First, the control points of that circle, which will contain the thickness of the individual strand, are created. Then the *v* knots for these circle control points are created.

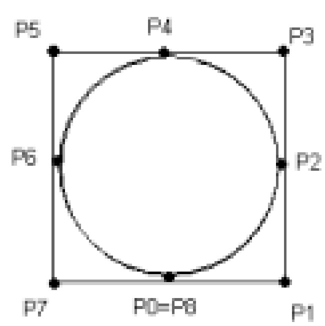

Figure 3: The thickness of each strand with 9 control points.

The twisting of the hair strand is determined by the current iteration of the mass-spring system. The u knots are based on how many particles that hair strand contains. Since in this case a hair strand contains only eight particles there are twelve u knots only.

After all the control points have been extracted for each hair strand, that hair strand is rendered as a twisted NURBS cylindrical surface.

```
For (i=0 … parts)
  For (j=0 … 9)
    MAKEV( cp[i*9+j],
         ADDV( particles[k*parts+i].p, circle))
```

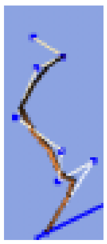

Figure 4: The mass-spring control points on a twisted NURBS cylindrical surface.

4.4 NURBS surfaces

A NURBS surface of degree p in the u direction and degree q in the v direction has the form:

$$S(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}P_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}}$$

The $\{P_{i,j}\}$ from a bi-directional control net, the $\{w_{i,j}\}$ are the weights, and the $\{N_{i,p(u)}\}$ and $\{N_{j,q(u)}\}$ are the non-rational B-spline basis functions defined on the knot vectors:

$$U = \{\underbrace{0,\ldots,0}_{p+1}, u_{p+1},\ldots,u_{r-p-1}, \underbrace{1,\ldots,1}_{p+1}\}$$

$$V = \{\underbrace{0,\ldots,0}_{q+1}, v_{q+1},\ldots,v_{s-q-1}, \underbrace{1,\ldots,1}_{q+1}\}$$

where $r = n+p+1$ and $s = m+q+1$.

5 Results

This implementation uses OpenGL and C++ on a Windows 2000 PC. The interface used is FLTK. It was compiled under Microsoft Visual Studio C++. The less hair strands the faster the rendering time for each iteration. However, even with 400 hair strands each iteration took less than _ a second. The bottleneck happens to be the rendering of the twisted NURBS cylindrical surfaces.

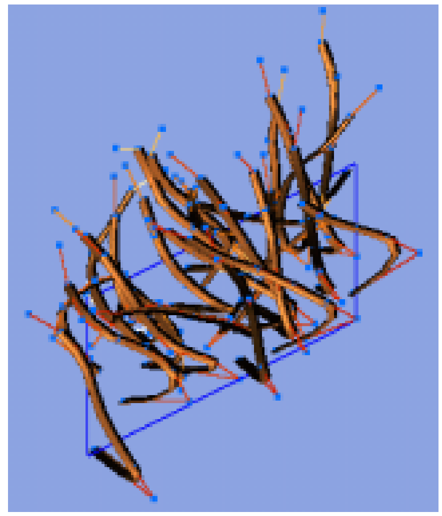

Figure 5: Messy hair on a plain in wireframe with mass-spring control points and twisted NURBS surfaces

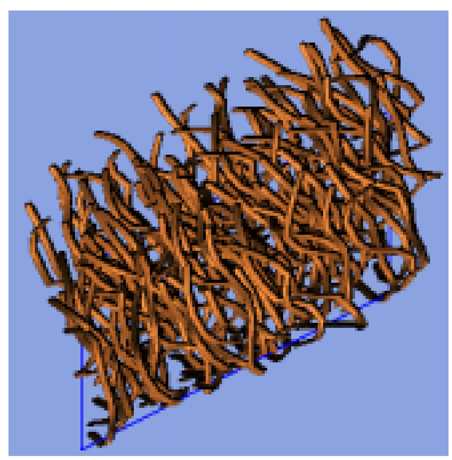

Figure 6: Messy hair on a plain in wireframe with twisted NURBS surfaces.

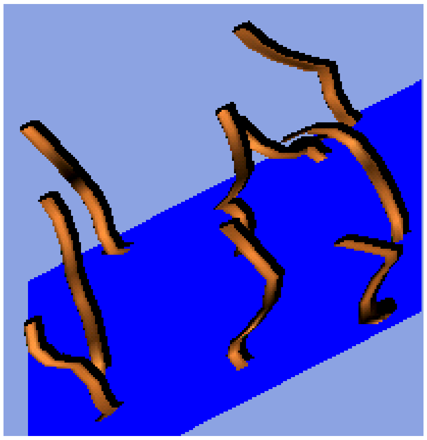

Figure 7: Thick, messy hair on a plain with twisted NURBS surfaces.

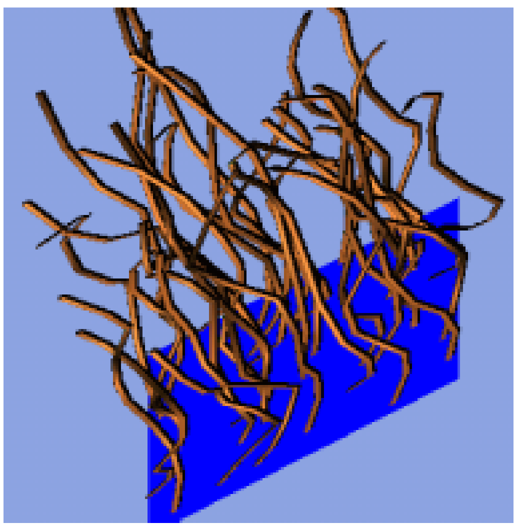

Figure 8:Long, messy hair on a plain with twisted NURBS surfaces.

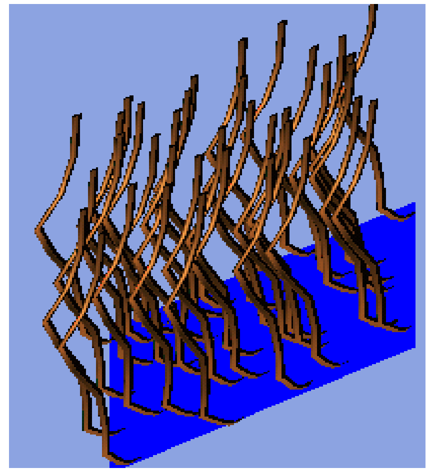

Figure 9:Long, curly, combed hair on a plain with twisted NURBS surfaces.

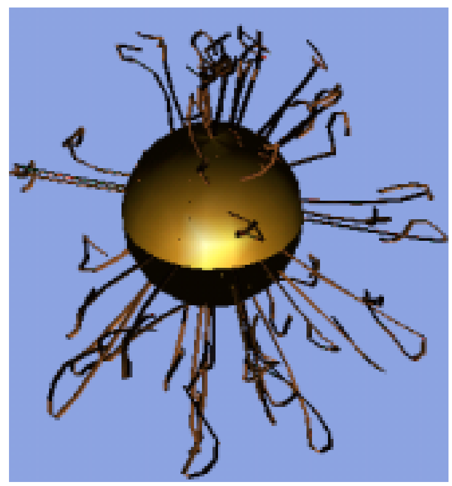

Figure 10:Long, messy hair on a sphere with twisted NURBS surfaces.

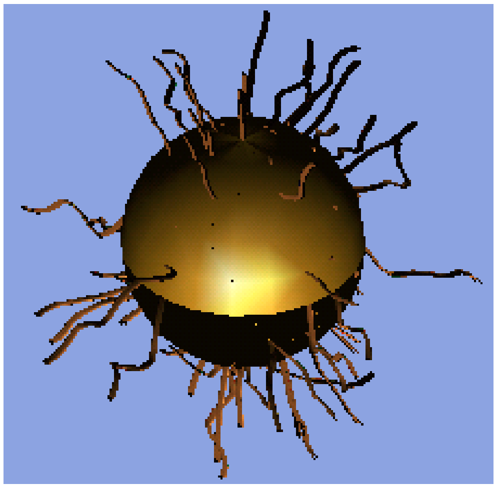

Figure 11: Short, curly hair on a sphere with twisted NURBS surfaces.

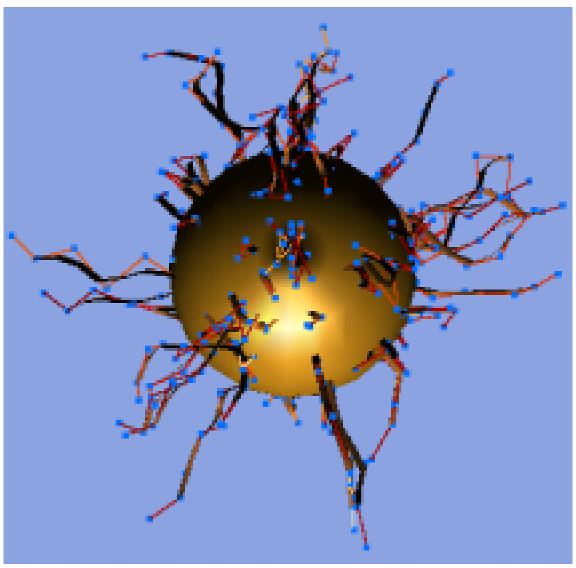

Figure 12: Curly hair on a sphere with mass-spring control points and twisted NURBS surfaces.

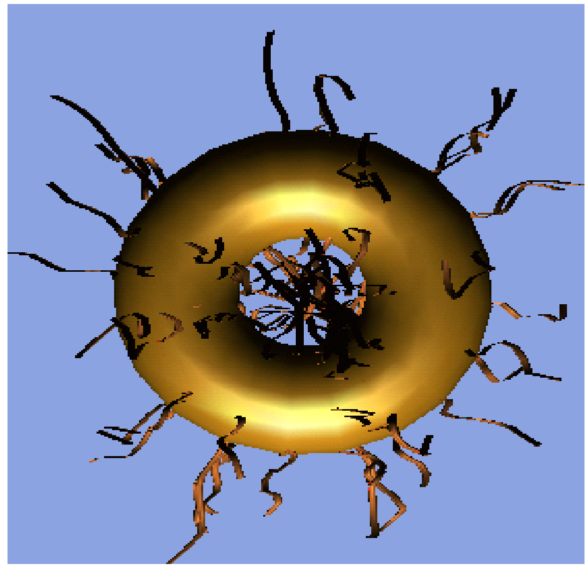

Figure 13: Short, curly hair on a torus with twisted NURBS surfaces.

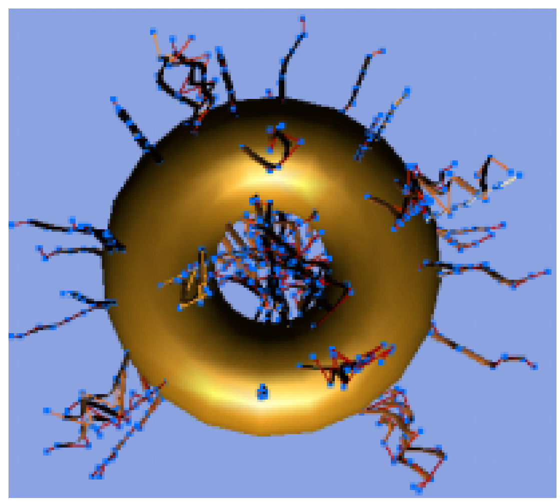

Figure 14: Curly, short hair on a sphere with mass-spring control points and twisted NURBS surfaces.

6 Conclusion

This paper I have presented a framework for modeling hair movement using mass-springs. This framework supports dynamic simulation, movement, and graphical rendering of hair and simulates different types of hair from very curvy to straight, from very short to very long, from very thin to very thick, and from very dense to bolding. It also simulates different affects forces have on hair from wind, gravity, and damping. Each individual strand of hair is modeled as a twisted NURBS cylindrical surface with *n* control points.

In the future I plan to implement collision detection and add angular mass-springs.

Figure 15: The user interface.

7 References

1. Kelly Ward, Ming C. Lin, Joohi Lee, Susan Fisher, and Dean Macri. Modeling Hair Using Level-of-Detail Representations. Computer Animation and Social Agents, 2003.
2. Koh, C. K., and Huang, Z. 2001. A Simple Physics Model to Animate Human Hair Modeled in 2D Strips in Real Time. Proceedings of Eurographics Workshop 2002, 127-138.
3. Yang Guang and Huang Zhiyong, A Method of Human Short Hair Modeling and Real Time Animation, IEEE, 2002
4. Tsuneya Kurihara, Ken-ichi Anjyo, and Daniel Thalmann, Hair Animation with Collision Detection, Models and Techniques in Computer Animation, Springer-Verlag, Tokyo, pp.128-138, 1993
5. Leslie Piegl and Wayne Tiller, Curve and Surface Constructions using rational B-splines, Computer-Aided Design, 19(9), 485-498, 1987
6. K. Anjyo, Y. Usami, and T. Kurihura. A Simple Method For Extracting The Natural Beauty Of Hair, SIGGRAPH (92), pp. 111-120 (1992).
7. W. Böhm. Insert New Knots into B-spline Curves, Journal of Computer Aided Design, 12 (4), pp. 199-201 (1980).
8. 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).
9. E. Cohen, T. Lyche, and R. Risenfeld. Discrete B-Splines and Subdivision Technique in Computer-Aided Geometric Design and Computer Graphics, CGIP, 14 (2), pp. 87-111 (1980).
10. 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 (1993).
11. C. K. Koh and Z. Huang, Real-time Animation of Human Hair Modeled in Strips, Computer Animation and Simulation, Springer-Verlag, pp.101-110 (2000).
12. R. E. Rosenblum, W E. Carlson, and 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 (1991).
13. 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 (1993).
14. J. Lengyel, Real-time fur. Proc. Of Eurographics Workshop on Rendering, 2000
15. J. Lengyel, E. Praun, A. Finkelstein, and H. Hoppe, Real-time fur over arbitrary surfaces, Proc. of ACM Symp. on Interactive 3D Graphics, 2001
16. NVIDIA. Final fantasy technology demo 2001. http://www.nvidia.com 2001
17. NVIDIA. Technical brief. http://developer.nvidia.com/docs/lo/1451/SUPP/accuview.final.pdf