The Motion of Hair

Gatlin Johnson
Tim Malone

May 7, 2010

Abstract

There are a number of different models in use for the efficient simulation of a mass of hair, from masses of single strands to volumetric masses which only become "hair" during illumination. Our goal was to simulate convincing hair motion, not convincing illumination, so we chose a simple mass-spring model and empirically discovered appropriate forces and spring constants which gave our model a nice bounce. During our research we also came across an alternative to Euler's method for the computation of differential equations. Our model is limited to modeling long straight or wavy hair under certain conditions, but we believe our basic approach can be applied to realistic movement of hair, cloth, and other objects.

0.1 Our Hair Model

0.1.1 Hairs as B-Splines

Our basic strand of hair is a B-spline curve whose control points are connected to one another by simple rigid springs. Initially we did not use any sort of damping effect, resulting in wildly chaotic hair which monotonically increased in energy very quickly. We also noted that the strands would lengthen a considerable amount before going back to their rest state. Accordingly, we empirically developed suitable damping and spring constants which gave a nice bounce to the hair when moved without creating any sort of noticeable lengthening. As with other models we have looked at over the semester, this is not a physically accurate model (most notably, it lacks conservation of energy and length); however, the results we came up with are convincing enough representations of long hair blowing in wind.

We experimented with non-uniform masses at each control point. Our thinking was that, since real hair is rooted in a follicle which essentially makes the lower part of a strand sturdier, that higher masses would resist the surrounding forces more and have the ends flutter around more like real hair. In the end the effects were not noticeably different from a uniform mass model, and the idea was scrapped.

0.1.2 Forces

In our animator, one can define any number of global forces, represented as 3-dimensional vectors. Our algorithm is based on Euler's method: for each control point other than the first, at every time step, do the following:

1. Calculate the spring forces pulling at the given control point from either side;
2. Sum the spring forces with the global forces into a total force;
3. Use this total force to calculate a new velocity for the control point;
4. Use this new velocity to calculate the new position of the control point.

The first control point in each hair is a special case: it does not move or allow any forces to act on it, so that there is a rigid anchoring off which the rest of the strand may flagellate.

We initially chose to use a straight application of Euler's method to update the position and velocity using previous values. Invariably, given enough time, our system would begin to go out of control, with the hairs jumping around erratically and spinning in an odd motion. In a set of lecture slides[1] from Rensselaer Polytechnic, the author described behavior very similar to what we were observing when using Euler's method with mass-spring systems.

As a correction, we took an idea from a project by students in Spain[2] to augment Euler's method. Instead of updating the position from previous values and then the velocity, we update the velocity and then use this new velocity to

[1] $http : //www.cs.rpi.edu/\ cutler/classes/advancedgraphics/S09/lectures/06_{m}ass_{s}pring_{s}ystems.pdf$

[2] $http : //www.gmrv.es/\ cgarre/APO_{M}assSpring_{J}un08.pdf$

update the position of the control point. This is a method termed the “leapfrog” method, and it corrected the erratic behavior caused by Euler’s method. Our animator has an option of turning leapfrogging on or off to illustrate the benefits of this new procedure.

0.1.3 Multiple hairs

We decided that, given that we could create the realistic spring-like motion of one hair, the next logical step would be to generate a set of hair from the motion of a few keyhairs at render-time. This way, we only compute the effects of forces on a few “keystrands.” This idea was borrowed from a paper by Pixar[3] on volumetric hair rendering. We constrain our keystrands to having the same number of control points. For the *ith* control point of each hair, we generate a B-spline curve using them as control points, and then evaluate the curve a certain number of times (depending on the density of hair wanted), and create new strands with the interpolated values as control points. This results in a smooth transition from one control point to another which allows any number of control points per keystrand and any number of keystrands. Additionally, the hairs do not intersect one another.

0.2 Limitations of our system

0.2.1 Types of hair

Our model was created with the image of long hair blowing in the wind in mind, as this is a very noticeable and visually pleasing effect when done properly. Each strand is modeled as a B-spline curve, which does not work very well for curly or frizzy hair. Though strictly speaking we did not set out to model every kind of hair but to model a certain kind of motion, this neuters our system’s potential as a useful rendering tool.

A paper on “super-helices,”[4] which are based on Cosserat curves, from a team in France demonstrated an efficient method of rendering different kinds of hair of any length, which would supplant our technique of rendering a single line. This would be a very useful improvement in future versions of our animator, however we did not incorporate these techniques owing to time and lack of complete mathematical understanding.

0.2.2 Intersection

We were not able to come up with an efficient collision detection system for each hair, and instead constrained our strands to interpolate smoothly from keystrand to keystrand in such a way that intersections do not occur. Also, seeing as how collision detection largely depends on how the actual strand is modeled, this would go hand in hand with our render model for each strand.

[3] $http://graphics.pixar.com/library/Hair/paper.pdf$
[4] $http://www-evasion.imag.fr/Publications/2006/BACQLL06/$

0.2.3 Fragile conditions

We experimented at great length on coming up with empirical spring and damping constants. However, changes to these constants can cause extremely erratic behavior at times for reasons which we are not entirely certain (most likely due to our integration approximation method). Use of a more stable integration approxmation technique, such as Runge-Kutta, might mitigate these issues.

0.2.4 Constant global forces

We did not create a way for forces to change over time in either magnitude or direction. Though our aim was not to create a general purpose animation system, it might be a useful feature to add in order to highlight how the hair reacts in different conditions more efficiently.

0.3 Applications of our system

0.3.1 Cloth

It occurred to us when completing the interpolation system that our model would be very useful in rendering the motion of cloth by incorporating additional spring forces and rendering our points as a mesh of control points rather than length-wise strands.

0.3.2 Strings and similar constructs

All values in our system are user-specified, allowing a great deal of freedom in deciding how springy the control points are, and which forces act on them. Thus, our system could be used to model the motion of, for example, a field of grass or flowers (owing to the ability to specify non-uniform masses for control points).

0.4 Artifacts

We have prepared a sequence of different "heads" of hair which are subject to a gamut of different forces, spring constants, and integration techniques (Euler's method and leapfrog). These all demonstrate different motions of hair, and will be presented as part of our in-class demonstration. These are encapsulated in simple text files which specify each keystrand, the number of hairs to interpolate between them, the global forces, and all other user-modifiable variables.

0.5 Bibliography

The following are papers we referenced in this report and others which we looked at over the course of our project.

- $http://www.cs.rpi.edu/\ cutler/classes/advancedgraphics/S09/lectures/06_{m}ass_{s}pring_{s}ystems.pdf$
- $http://www.gmrv.es/\ cgarre/APO_{M}assSpring_{J}un08.pdf$

- *http : //graphics.pixar.com/library/Hair/paper.pdf*
- *http : //www − evasion.imag.fr/Publications/*2006*/BACQLL*06*/*
- *http : //physbam.stanford.edu/ aselle/papers/*9*/hair.pdf*