Interactive Hair Simulation Using 2D Hair Strips

By Sean Sullivan

ABSTRACT

This paper focuses on creating an interactive model which simulates hair. By using hair strips, it is possible to make a simulation that runs much faster than a simulation that tries to draw even a small number of hairs. This paper creates a basic implementation of this method.

INTRODUCTION

The modeling of hair has always been a problem in computer graphics. The very nature of hair makes it a difficult to try to model, as a perfectly accurate model would require the modeling of thousands of strands of hair, each which would require a physics simulation and collision detection. In order to make a quicker program, I modeled groups of hair together as strips, reducing the number of animation I had to do by several magnitudes.

Background

An older paper about hair simulation by Daldegan [1993] described a system for modeling a hairs movement in a physics system, including calculating its torque due to internal and external forces. This model created a solid groundwork from which more accurate simulations arouse, however a majority of these sacrificed speed for accuracy [Koh 2001]. Koh then proposed modeling of hair in strips, by which a strip would represent a group of hair without having to make thousands of individual strands [2001]. Each of the strips was then modeled with a NURBS surface using minimal control points in order to maximize efficiency. He also used a bounding sphere to simulate the head in order to make the collision detection system faster. Koster [2004] showed how this general method could be improved through the use of alpha channels on a texture on a strip to simulate the look of individual strands of hair and to make the strips blend together better. He also made use of the buffers and the general volume of the hair to simulate anisotropic reflections.

The Implementation

My model gains much of its structure from Koh's, in that it creates a set of hair strips that each simulate a grouping of hair strands. Each strip simulates two hairs which form the control points to a textured NURBS surface. The hair strands are simulated using an implementation of connectivity similar to that of a standard cloth system [1996], combined with a collision detection system and an angle monitor.

HAIR CREATOR

The hair is generated by the shape of the head, and depends on the triangles of mesh. The segments are generated using the center point of the triangle as well as one of the other points. This occasionally leads to several of the hair strips to have normals that are opposite to the positive direction, however this was kept as it provides for ad-hoc 'highlights' to the hair, as the segments whose normals are backwards will show up as a different shade. To make the hair look better, I used two different methods. One was to increase the number of triangles by subdividing. This adds volume to the hair, however it does not improve the look of the hair in front (as the front of the hair is often shorter than the back).. So, instead I tried smarter

placement of rows of shorter hair. This makes much better results, with less of a cost.

PHYSICS

Each Hair strand is divided into several hair points, each of which keeps track of its position, and its nearest neighbors. The hair strands then apply a force to each of the particles and they fall down according to the force of the environment combined with the force of the spring connections that occur between the nearest neighbors. The hair strands then make corrections to the distance between particles according to Provot [1996]. The system then checks the angle that emerges between the hair particles [1993], and applies a force if the follicle bends too far. Finally, the system goes through a simple collision detection to determine if the control points are too close to the head. If they are found to be too close. The implementation follows Koh [2001] and creates a sphere to check the hair against, however it then goes into a more specific test against the vertices of the mesh in order to make it more accurate. To improve speed, the mesh is divided up into many small segments which each contain some of the vertices, and the hair particle is only compared to one of these sub-areas.

MESH CREATION

After the hair is animated several times, it is then painted. This entails the creation of a NURBS surface between each of the hair pairs, and then a texturing of the surface. Both of these use the default openGL configuration.

Technical Difficulties

ERRORS

My model makes a nice looking and runs interactively, even if somewhat sluggishly when there are a large number of hair segments, however there were many problems that came to light in its creation. One of the larges bugs I never quite solved, and continues to cause problems when there are a large number of hairs, is a stack corruption error. I do not know what causes it, however it forced me to make the hair class more independent than I wanted it to. The problem surfaced when I tried to pass the vector of the divided up vertices of a mesh: the vector suddenly seemed to cease to exist, throwing a corrupted stack as a cause. This error surfaced several more times, making it necessary to use much more memory than I originally intended. I also suspect this bug to be the reason that some of the other meshes seem to crash on use with the program, which makes the on the fly generation of hair particles somewhat hard to demonstrate.

OTHER ISSUES

Besides the stack corruption, I had to rid the hair of jitter, and force the angle restrictor to respond at the correct angles and to give a reasonable response.

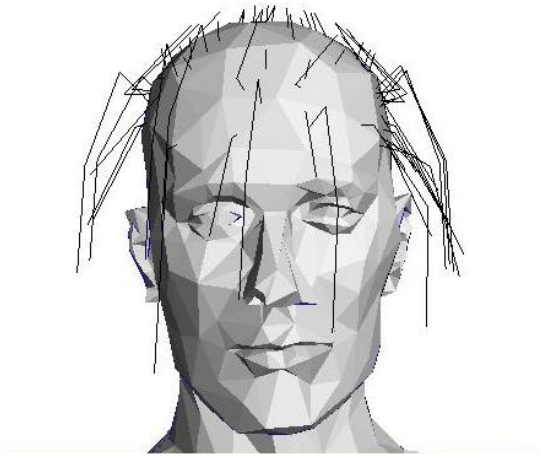

Fig. 1: Before Angle, and collision issues were resolved.

UNSOLVED PROBLEMS

One problem that I tried to solve but didn't perfect was the collision detection around the upper particles. As between the root hair point and the second hair point is somewhat large, by simply obeying the regular physics at each point the interpolated surface sometimes would penetrate the surface. I tried to solve this problem by sensitivity for collision for the second hair particle, but it still somewhat persisted in some situations. The texture application isn't perfect either, as when I try to apply some of my custom textures the program simply fails to recognize them, making it impossible to do the alpha mapping and fake highlighting that I was planning on doing. Also, due to some issue that I have not discovered, when the simulation runs with a

large number of hairs the performance of the program slowly degrades until it becomes unusable.

Results

The run time of the program is largely dependant on the implementation of the number of hair segments, as each one requires about 100 more runs through the collision detection system and the neighbor adjustor. The collision detection also becomes a major bottleneck as even though it is efficient if the hair is far away, as most hair strands end up near the head, each point must go through all of the nearest points. However, even with this the system can attain a frame rate of around 30 with a un-subdivided mesh and 12 with a subdivided mesh(however after a random amount of time, the program crashes), making it at least interactive time. However after running for a short time, some sort of error begins to emerge that slowly degrades the speed of the program and uses up a large amount of ram. The assignment took a long time to complete, including several hours a day for at least the past two weeks. However, even with its faults, it still does a reasonable job of making a head of hair in a relatively fast amount of time. If I had more time, I probably could have improved on the code by making the hair look a little better (getting a better texture) and maybe getting rid of the bug that seems to be draining the performance of the system.

IMAGES:

Fig 2: Rendering of 70 hair strips. (Left to Right) Hair fallen, hair falling to the left, and the hair strips as a wireframe.

References

Koh C.K., and Huang Z. 2001. A Simple Physics Model to Animate Human Hair Modeled in 2D Strips in Real Time. In *AMC Digital Library.* URL: http://portal.acm.org/citation.cfm?id=776362

Koster M, Haber J, SeidelD. 2004. Real-Time Rendering of Human Hair using Programmable Graphics Hardware. In *Computer Graphics International* pg. 248-256. URL: http://www.mpi-sb.mpg.de/resources/FAM/publ/cgi2004.pdf

Daldegan A, Thalmann N, Kurihara T, Thalmann D. 1993. An Integrated System for Modeling, Animating and Rendering Hair. In *Computer Graphics Forum.* pg. 211-221. URL: http://vrlab.epfl.ch/Publications/pdf/Daldegan_and_al_EG_93.pdf

Xavier Provot. Deformation Constraints in a Mass-Spring Model to Describe Ridid Body Cloth Behaivor. 1996. In In Graphics Interface. Pg. 147-154. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.16.4040