Real-Time Hair Simulation for Low Dynamic Movements

CIS 499 SENIOR PROJECT DESIGN DOCUMENT

Emily Weihrich
Advisor: Dr. Norm Badler
University of Pennsylvania

PROJECT ABSTRACT

Simulating hair on a virtual character remains one of the most challenging aspects of computer graphics. Currently, there are numerous different techniques for modeling, animating, and rendering believable hairstyles.

With the emergence of new-generation GPUs, real-time hair simulation techniques are beginning to adopt more detail-promoting methodologies. Computations that once took hours, such as hair-hair collisions, now take minutes; likewise, computations that used to take minutes have been condensed to times favorable for real-time results. This general trend is visible across all three subcategories of hair simulation: modeling, animating, and rendering.

This project seeks to use current advances in real-time hair simulation to create two realistic (in terms of modeling, animating, and rendering) hairstyles, one male and one female, for virtual characters performing simple head movements, such as nods and turns. Since "main" characters are involved, the chosen simulation technique should be a merger of both computationally-friendly and detail-promoting methods. The end product would ideally be used in any context where real-time, low dynamic movement and rendering of hair are required.

Project blog:
http://realtimehair4nods.blogspot.com/

1. INTROUDUCTION

Simulating hair on a virtual character remains one of the most challenging aspects of computer graphics. Currently, there are numerous different techniques for modeling, animating, and rendering believable hairstyles. We have indeed come a long way from the early days of simple hair approximation, in which key-framed polygonal meshes with

texture maps were used (as seen in video games like *Lara Croft: Tomb Raider*, first released in 1996).

Since virtual characters may be implemented in a variety of ways, these simulation techniques often work toward different goals. For example, due to the high frequency of close-ups on virtual characters in animated films, studios in the motion picture industry tend to favor hair simulation techniques that preserve detail over those that yield efficient simulation times. At the other end of the spectrum, studios in the game industry are more likely to favor fast computation over realism, since their virtual characters must be manipulated and rendered in real-time. This divergence within the context of hair simulation has led to a positive correlation between detail and computation time for most techniques.

With the emergence of new-generation GPUs, however, real-time hair simulation techniques are beginning to adopt more detail-promoting methodologies. Computations that once took hours, such as hair-hair collisions, now take minutes; likewise, computations that used to take minutes have been condensed to times favorable for real-time results. This general trend is visible across all three subcategories of hair simulation: modeling, animating, and rendering.

1.1. Significance of Problem or Production/Development Need

This project seeks to use current advances in real-time hair simulation to create two realistic (in terms of modeling, animating, and rendering) hairstyles, one male and one female, for virtual characters performing simple head movements, such as nods and turns. Since “main” characters are involved, the chosen simulation technique should be a merger of both computationally-friendly and detail-promoting methods. In other words, the hairstyles should not draw attention away from the characters to which they are applied. The project will proceed incrementally, starting with a crude "base case", which will be built upon to produce more realistic results. The end product would ideally be used in any context where real-time movement and rendering of hair are required. This project is also intended as a means of gaining personal experience in preparation for entry into the animation industry.

1.2. Technology

This project will be carried out (preferably) on a Linux box (or Windows machine) with a high-end graphics card (NVIDIA, if possible). Programming will be done in C++ (C when appropriate) and will likely utilize the following libraries / toolkits: CUDA, OpenGL, FLTK or QT, OpenMesh. Since a good potion of this project involves evaluating different hair simulation techniques, several papers will be consulted (please refer to the “REFERENCES” section for a current listing). The project’s “base case” will most likely combine and implement the simulation techniques discussed in the following two sources: (1) an excerpt from the book *GPU Gems 2*, found on NVIDIA’s developer site,

(see http://http.developer.nvidia.com/GPUGems2/gpugems2_chapter23.html), and (2) the open-source documentation of a class project created by Princeton graduate student Connelly Barnes (see http://www.connellybarnes.com/documents/hair/).

1.3. Design Goals

Since "main" characters are involved, the chosen simulation technique should be a merger of both computationally-friendly and detail-promoting methods. In other words, the generated hairstyles should not draw attention away from the characters to which they are applied.

Another important design decision pertains to the environment in which simulation techniques will be tested and presented. If an existing viewer cannot be used, a simple UI viewer will be created with (if possible) the following basic features: (1) a point light to light the scene, (2) a movable camera for viewing the virtual character from different angles, (3) some means of moving the virtual character's head.

1.3.1 Target Audience

This project aims to produce results usable by individuals working in areas of development or production where real-time movement and rendering of hair are required (or prove useful). Examples of such areas may include: video games, background characters in movies, virtual characters driven (in real-time) by motion capture data, characters in virtual worlds, and other student projects. Depending on the amount of progress made, this project could produce a unique technique (or merger of techniques) for hair simulation that has never been used before, thereby providing a stepping stone for further research.

1.3.2 User goals and objectives

Given a mesh object (preferably a scalp represented as an .obj file), the user should be able to load the object and add either a female or male hairstyle to it. The user should also be able to load additional collider objects with which the hairstyle is expected to collide (such as a head model, also represented as an .obj). The hairstyle and mesh objects should render in real-time and show up in the UI viewer. Changes to the hairstyle brought on by movements of the mesh objects (hair surface and colliders) or point light should be reflected in the viewer instantaneously. If time allows, additional features, such as user control of hair forces, length, and style, may be available as part of the UI viewer.

Obviously, interacting with the UI viewer would cause the user to focus their attention on the hairstyle. However, if the code is ported such that the hairstyle can be added to a virtual character in a different framework, the goal would be to have the user acknowledge the hair as realistic (in terms of appearance and movement) and focus more on what the virtual character is doing.

1.3.3 Project features and functionality

The main features of this project include:

- Research into the usefulness of different real-time hair simulation techniques. This research will help to determine the steps that will be taken to build up from the "base case" simulation.
- Implement at least two simulation techniques: (1) a crude "base case" and (2) a nicer, more realistic technique. Since hair simulation can be broken down into three subcategories (modeling, animating, rendering), the "base case" would include quick and dirty algorithms for all three subcategories. Likewise, the nicer technique would involve improvements made to the "base case" in two (animating, rendering) if not all three simulation subcategories.
- Creation of a simple UI viewer with callback functions to allow for user control over the camera, light, virtual character, and the hairstyle. The extent of the last two controls is based largely on project time constraints.
- Creation of a C++ class that accomplishes the user goals and objectives listed above. This class should be designed so that it can be easily integrated into other C++ frameworks.

2. PRIOR WORK

[I will fill in this section after I read through my references more thoroughly. I have just been collecting and skimming through papers so far, so I don't have any concrete dates or names; only lists of techniques.]

3. PROJECT DEVELOPMENT APPROACH

3.1. Algorithm Details

The project will proceed incrementally, starting with a crude "base case", which will be built upon to produce more realistic results. It is expected that the project will eventually implement at least two different simulation techniques.

Hair simulation can be divided into three subcategories: modeling, animating, and rendering. Potential real-time hair simulation techniques are listed below under the subcategory they most relate to. Since the project is in its early stages, this list has yet to be narrowed down. Likewise, techniques not on this list (including those that have not yet been used in a real-time context) may also prove to be useful.

(1) Modeling

a. LODs (Level-of-Detail representations)
    i. Strips
    ii. Strands (same as sparse guide hairs?)

iii. Wisps
1. AWT (Adaptive Wisp Tree)

b. Sparse guide hairs (curves) that are interpolated to create density
i. Grown with 2d texture synthesis
ii. Or grown based on scalp geometry

(2) Animating
a. Mass-Spring-Damper
b. Fluid dynamics (volume-related algorithms)
i. Velocity fields
c. Particle dynamics
i. Control vertices of hair curves treated as particles
1. Verlet integration
2. Euler integration
3. Runge-Kutta integration
d. Intersecting spheres for collision detection
e. Metaballs for collision detection

(3) Rendering
a. Alpha maps
b. Opacity shadow maps
i. Voxels
c. Anisotropic reflections
i. Marschner model
ii. Dual scattering approximation

Techniques/algorithms in **bold** are those discussed in NVIDIA's *GPU Gems 2* excerpt and Connelly's project write-up. The "base case" simulation will most likely be constructed using these techniques.

3.2. Target Platforms

3.2.1 Hardware

Linux box (or Windows machine) with NVIDIA graphics card

3.2.2 Software

C++ (maybe C too), CUDA (maybe), OpenGL, FLTK or QT, OpenMesh

4. WORK PLAN

4.1.1. Project Milestone Report (Alpha Version)

- Completed revisions of sections in the proposal that are currently under-developed or need rewriting
- A UI viewer which does the following:
  - Loads a scalp

    - Loads collider objects
    - Allows the user to move the scalp and colliders (nods and turns)
    - Allows the user to pan/rotate around the scalp with the camera
    - Allows the user to change the position of the light source
    - Adds a generic hairstyle to the scalp (via “base case” modeling)
- Have “base case” animation and collision working
- Have “base case” rendering working (if time permits)

4.1.2. Project Final Deliverables

- A developed UI viewer which does the following (in addition to the capabilities listed above):
    - Adds either a female or male hairstyle to the scalp (via “nice case” modeling)
    - Allows the user to toggle between simulation methods (“base” and “nice”)
    - Allows the user to control the forces acting on the hairstyle (if time permits)
    - Allows the user to control the length of the hairs in the hairstyle (if time permits)
    - Allows the user to control the shape of the hairstyle (if time permits)
- Have “nice case” animation and collision working
- Have “nice case” rendering working

4.1.3 Project timeline

09/21/2009 – 09/27/2009	Continue finding and reading research papers related to hair simulation (both real-time and not). Make an Excel spreadsheet that matches up papers (including when they were written) with the algorithms they discuss. Meet with Joe, Norm, or others to examine algorithms discussed in both NVIDIA book excerpt (Nalu demo) and Connelly’s write-up. Decide whether or not they make a good “base case”. Obtain access to a Linux box (or Windows machine) and choose a C++ environment (if Windows machine, preferably Visual Studio).
09/28/2009 – 10/04/2009	Obtain a scalp and virtual character head from Catherine (both as .obj files). Decide if her basic viewer is sufficient or if it will be better for me to build my own. Begin building or modifying the viewer, as necessary (talk with Joe to see if this involves working with the GPU). Continue reading the research papers that I have collected up to this point (and putting them

into the Excel database). Use feedback from Joe, Norm, and others to revise under-developed sections of the proposal.

10/05/2009 – 10/11/2009 Begin implementing the major algorithms associated with the NVIDIA Nalu demo and Connelly's write-up. Focus on modeling and animating first, and then on rendering. Use Connelly's source code as a reference.

10/12/2009 – 10/18/2009 Prepare for the milestone report. 10/16/2009 is the **Alpha Review**.

10/19/2009 – 10/25/2009 Finish up the "base case" simulation if I was unable to do so in time for the alpha review. Narrow down which techniques I want to use for the nicer, more realistic simulation.

10/26/2009 – 11/01/2009 Begin making improvements to the "base case" in one of the three subcategories: modeling, animating, or rendering.

11/02/2009 – 11/08/2009 Finish making improvements to the "base case" in that subcategory.

11/09/2009 – 11/15/2009 Begin making improvements to the "base case" in another one of the three subcategories.

11/16/2009 – 11/22/2009 Finish making improvements to the "base case" in that category.

11/23/2009 – 11/29/2009 Begin making improvements to the "base case" in the subcategory that has not yet been addressed.

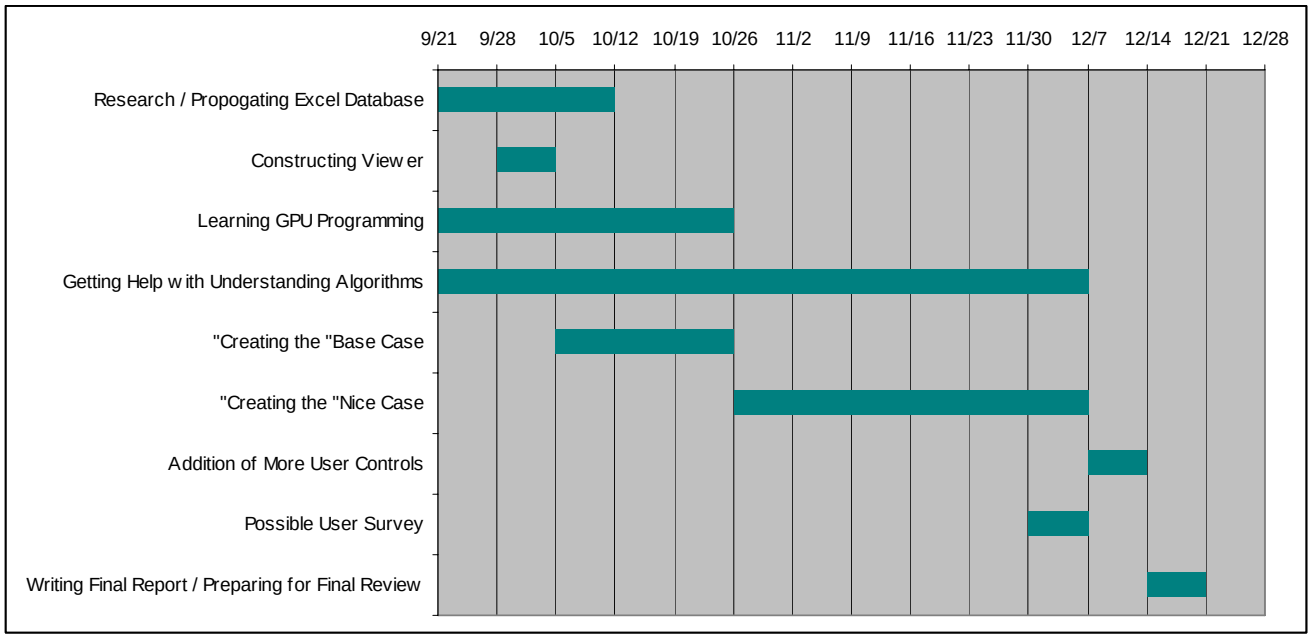
11/30/2009 – 12/06/2009 Finish making improvements to the "base case" in that category.

[If developing the "nice case" does not take as long as projected, an extra week could be devoted to surveying how the hair is perceived by student users. For example, the project code could be ported to a different framework in which virtual character expressions or lip-synching is being tested; users of this other framework could be asked what attracted their attention during the test.]

12/07/2009 – 12/13/2009 Decide what hairstyle parameters will be associated with each hairstyle (male and female). Add user

	controls for the manipulation of hair forces, length, and style.
12/14/2009 – 12/20/2009	Add final revisions to the project report. Prepare for the **Final Review**.

4.1.4 Gantt Chart



5. REFERENCES

[Proper formatting will be added later.]

Hair Animation and Rendering (NVIDIA Nalu Demo)

Dual Scattering Approximation

Connelly - Technical (blog): Real-Time Hair Animation

Animating Complex Hairstyles in Real-Time

Real-Time Animation of Complex Hairstyles

Real Time Hair Simulation and Rendering on the GPU

Modeling Hair Using Level-of-Detail Representations

Adaptive Grouping and Subdivision for Simulating Hair Dynamics

Detail Preserving Continuum Simulation of Straight Hair

Simulating Complex Hair with Robust Collision Handling

A Mass Spring Model for Hair Simulation

Adaptive Wisp Tree

Example-Based Hair Geometry Synthesis

Realistic Hair Simulation: Animation and Rendering

Photo-realistic Hair Modeling, Animation, and Rendering