Physically-based Simulation of Hair Strips in Real-Time

Hasan Doğu Taşkıran and **Uğur Güdükbay**
Department of Computer Engineering
Bilkent University
Bilkent 06800 Ankara, Turkey
Phone: {(+90) 533 513 1426, (+90) 312 290 1386}
E-mail: {taskiran, gudukbay}@cs.bilkent.edu.tr

ABSTRACT

In this paper, we present our implementation of physically-based simulation of hair strips. We used a mass-spring model followed by a hybrid approach where particle systems and the method of clustering of hair strands are employed. All the forces related to springs are implemented: gravity, repulsions from collisions (head and ground), absorption (ground only), frictions (ground and air), internal spring frictions. Real-time performance is achieved for physically-based simulation of hair strips and promising results in terms of the realistic hair behavior and hair rendering are obtained.

Keywords: hair simulation, physically-based, real-time, strip-based

1 INTRODUCTION

Hair is very important to create natural looking virtual characters. However, producing realistic looking hair is one of the most challenging problems in computer graphics. There are approximately 100,000 to 150,000 individual hair strands to simulate on the scalp of a human head. Besides, different hairs have different colors and varying degrees of waviness and thickness. These complexities and the number of hair strands on the head create difficulties for generating a formulation for modeling and simulating hair [HT00, LK01]. There are three main aspects of simulating hair: *hair modeling*, *hair animation (dynamics)*, and *hair rendering*. There are various approaches to hair modeling, dynamics and rendering such as the explicit models, particle systems [BCN03], volumetric textures, cluster models and hair as a fluid etc. A detailed survey and discussion on the state of the art in these aspects can be found in Hadap et al. [HT00].

Each individual hair strand interacts with its environment. From a dynamics point of view, most important interactions occur with the head, the body itself, and the other strands. Strands are also responsive to the external forces such as the movement of the head, collisions with obstacles, the wind and the forces created by the static electricity, etc. All these interactions have to be simulated for each strand to achieve realistic hair behavior and accurate and smooth animation. Since the number of strands is very large we need somehow to simplify physical formulas for the motion of the hair and fast and high-quality collision detection mechanisms to be able to do the simulation in real-time [KH00, KH01, CJY02].

WSCG SHORT Papers Proceedings, ISBN 80-903100-9-5
WSCG'2005, January 31-February 4, 2005
Plzen, Czech Republic.

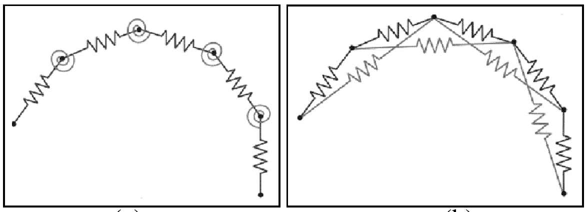
2 OUR WORK

In this paper we are mainly concerned with the dynamics and rendering aspects of human hair simulation. We used a mass-spring model for implementing hair dynamics and followed a hybrid approach where particle systems and the cluster model are used.

Hair Modeling and Rendering

Human hair may be modeled in strips (clusters) but this somehow results in less realistic images. However, increasing the number of the strips may solve this problem. The more strips you create, the better you approximate the hair. Of course, the better you approximate the hair, the more calculations are needed for the physically-based simulation of the hair [WLL$^+$03, YXWY00, KH01].

The rendering of hair involves dealing with hair color and texture, shadows, specular highlights, varying degrees of transparency, and anti-aliasing. Mainly, we only used strips to visualize the hair realistically in our implementation. An alpha blended texture, which is used to make the strips that has randomized tips, is mapped onto the patches that were created on the fly from top to bottom of each hair strip. The texture and the method we used is the same as in [KH01]. Since we mainly concentrated on the dynamics and rendering of the hair, no hair style was applied. However, some curliness was given to the strands by procedural meth-



(a) (b)

Figure 1: The angular spring model for hair strands (a) and the linear spring representation of the corresponding model (b).

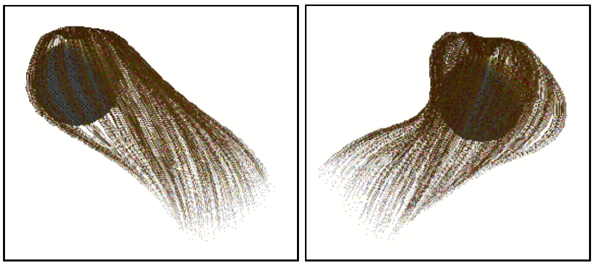


Figure 2: Collisions with the head and the bending stiffness of the strips.

ods. The connection points of hairs are determined by using the tools 3ds max 4.2 and Milkshape3D.

Hair Dynamics

A simple mass-spring system for an individual hair's dynamics includes a damping factor that is used to calculate the bending stiffness. To be able to simulate the bending stiffness and elasticity of each hair strand, we may see the hair as an oriented particle system as shown in Figure 1 where each spring constant affects both bending and stretching of the strand [iAUK92, RCT91, HT01]. The effects of the bending stiffness and collision with the head is shown in Figure 2.

The spirals shown at each particle position represents the angular springs. This is used to obtain the bending stiffness needed for each hair strand. However, linear springs are used to represent those angular springs for uniformity and performance. Linear representations of angular springs supply bending and torsion effects. We used springs with relatively lower spring constants and lower internal friction constants in their implementation. Thus, some other springs are connected with particles such that each particle also has a connection with the particle next to its neighbor.

The implemented forces, which have an impact on the motion of the hair strands, are spring forces, gravity, repulsions from collisions (head and ground), absorption (ground only), friction (ground and air), and internal spring frictions.

Although the cantilever dynamics and collisions of each hair strand with the body are within the reach of current computing power, modeling the complex hair-to-hair interactions is still a challenge [iAUK92, RCT91, KH01, KH00]. Therefore, hair-to-hair interactions are not considered in this implementation to achieve real-time performance.

3 IMPLEMENTATION

The hair simulation algorithm is given in pseudo-code in Figure 3 and the algorithm for solving the equations of motion of the hair particles is given in Figure 4. At each frame update, we iterate the springs used in the simulation a number of times according to the seconds-per-frame rate and a predefined maximum seconds that may pass between two frames of the animation. This is needed in order to prevent a pass over of the non-precise time calculation.

A simulation is first initialized, solved and then updated. In our implementation, each hair strand is a separate simulation of particles and springs that hold those particles together. Particles are initialized to their stable states at each frame, i.e. the forces applied on the masses are set to zero. Each new force adds a new constraint to the movement of the mass. A spring is just an additional force on the two masses that this spring is virtually connecting together.

At each frame of the simulation, each hair strip, which is also a simulation itself, is simulated independently. All the external forces and the forces that stem from the springs are added together, which give the hair its final movement direction. When the end of the frame update comes, the changes in the velocities of the particles are calculated, which then affects the change in the position of the mass. Forces are applied according to the time between two frames and the motions of the virtual particles are simulated afterwards.

Masses have properties such as their position, and the force that is applied on this mass at an instance. Springs have properties such as their spring constants, inner friction constants and their stable lengths. There are other constants such as the air friction constant, ground friction constant, ground repulsion constant, ground absorption constant and most importantly the gravitational acceleration constant. All these constants are given when initializing the simulation. A simple sphere is used for the head and a simple plane is used for the ground. No collisions applied with the body of the model.

After the forces on each particle are calculated, these forces are applied on each particle. Basic physics formulas apply at this point and the velocity of the particle and its new position are calculated. We used Euler Method for solving the equations of motion. This method is not always accurate, but it is simple to implement.

4 RESULTS

We implemented the described methods using the C++ language and OpenGL[1]. Performance results of our im-

[1] Animations generated with our implementation can be seen at `http://www.cs.bilkent.edu.tr/~gudukbay/hairmodeling.html`.

```
procedure SolveHairSimulation {

  for i from 0 to numHairStrands do {
    // Initialize the masses for this frame
    for j from 0 to hair(i)->numParticles do
        hair(i)->particle(j)->force := 0;

    // Solve spring forces and apply forces
    // on the masses that this spring connects
    for j from 0 to hair(i)->numSprings do {
      hair(i)->spring(j)->force := 0;
      // Solve spring equation by taking the
      // spring's internal friction into account
      solve(hair(i)->spring(j));

      // Apply the spring force to the masses
      // in opposite directions
      applyForce(hair(i)->spring(j)->particle1,
                 hair(i)->spring(j)->force);
      applyForce(hair(i)->spring(j)->particle2,
                 -1 * hair(i)->spring(j)->force);
    }

    // Apply other forces
    for j from 0 to hair(i)->numParticles do {
      // Apply gravitation
      applyForce(hair(i)->particle(j),
                gravitation_force);
      // Apply air friction
      applyForce(hair(i)->particle(j),
                air_friction_force);
      // Collide with the head
      if hair(i)->particle(j) is colliding
         with the head then do
         applyForce(hair(i)->particle(j),
                    head_repulsion_force);

      // Collide with the ground
      if hair(i)->particle(j) is colliding
         with the ground then do
      {
        // Absorb some energy from the hair
        // by applying an absorption force
        applyForce(hair(i)->particle(j),
                   ground_absorption_force);
        // Apply the ground friction
        applyForce(hair(i)->particle(j),
                   ground_friction_force);
        // Apply the ground repulsion
        applyForce(hair(i)->particle(j),
                   ground_repulsion_force);
      }
    }
  }
}
```

Figure 3: The algorithm for solving the hair simulation

```
procedure SimulateHair(dt) {

  // Solve the simulation
  SolveHairSimulation();

  for i from 0 to numHairStrands do {
    for j from 0 to hair(i)->numParticles do {
      // Calculate new velocity of each particle
      hair(i)->particle(j)->velocity +=
        dt * (hair(i)->particle(j)->force /
              hair(i)->particle(j)->mass);
      // Calculate new position of each particle
      hair(i)->particle(j)->position
        += dt * hair(i)->particle(j)->velocity;
    }
  }
}
```

Figure 4: The algorithm for solving the equations of motion for hair particles

NHS	*PES*	*NSP*	*SES*	*NSS*	*FPS*
100	10	1000	17	1700	62.50
100	15	1500	27	2700	62.50
100	20	2000	37	3700	62.50
100	25	2500	47	4700	31.50
200	5	1000	7	1400	62.50
200	10	2000	17	3400	62.50
200	12	2400	21	4200	32.50
200	14	2800	25	5000	10.60
300	6	1800	9	2700	62.50
300	7	2100	11	3300	45.20
300	8	2400	13	3900	32.00
300	10	3000	17	5100	6.41
400	3	1200	3	1200	62.50
400	5	2000	7	2800	45.20
400	6	2400	9	3600	21.25
400	7	2800	11	4400	8.00

Table 1: Performance results of the simulation. **NHS**: Number of Hair Strips, **PES**: Particles at each Strip, **NSP**: Number of Simulated Particles, **SES**: Springs at each strip, **NSS**: Number of Simulated Springs, **FPS**: Frames-per-Second.

plementation, running on a computer that has a Pentium IV (HT) 2.80GHz CPU, NVIDIA GeForce FX 5600 chipset VGA Adapter and 512 MB RAM, is given in Table 1.

All the other parameters that affect the frame rate were held fixed as the number of particles are increased in the simulation. As the number of particles changes, the number of springs connecting the particles together changes too. Each additional particle adds two more springs to the system. Total number of springs are the number of hair strips multiplied by the number of springs for each strip.

We may derive from the table that after some point, an increase in the number of particles - and so the springs - creates an exponential degradation in performance. Of course there are other factors affecting these results such as the usage of alpha blending, increase in the number of patches as the number of particles increase, the width of the patches used, etc. However, we could still obtain the realtime performance with 300 hair stripes each having 8 particles, or 400 hair stripes each having 5 particles. Most of the force calculation steps are dedicated to the springs. The spring force calculation includes a square-root calculation, 3 multiplications and a division operation. It is possible to optimize the simulation for speed. We could create a look-up table for the square-root values to eliminate the costly square-root function calls.

5 CONCLUSION AND FUTURE WORK

We presented our implementation of physically-based simulation of hair strips. It is still very far to imagine hair blowing with full rendering and collisions in real-time, however we obtained promising results. We will work on other applications of our implementation.

As a future work, levels-of-detail for hair may be added as described in [KH01] and [WLL$^+$03, WL03]. This can be done in two ways. Either we may decrease the number of hair strands, or we may decrease the number of particles used to represent each hair strand while the viewer is far from the hair. Both of these methods reduce the calculations needed in terms of dynamics and graphics.

Besides, other collisions, such as the ones with the body and obstacles, etc., and other external forces, such as the wind, static electricity, etc., could be handled. Wet hair simulation could be done by increasing the weight of each particle in the hair. Another improvement could be adding hair shape modeling into our framework. A comb, which is used to brush the hair would not be just a fine addition, but also it would be a good practice of dynamics.

6 ACKNOWLEDGEMENTS

This work is partially supported by Turkish State Planning Organization (DPT) under Grant No. 2004K120720. 3ds max is a registered trademark of Autodesk; Milkshape3D is a registered trademark of ChumbalumSoft; Pentium is a registered trademark of Intel; GeForce is a registered trademark of NVIDIA; OpenGL is a registered trademark of Silicon Graphics.

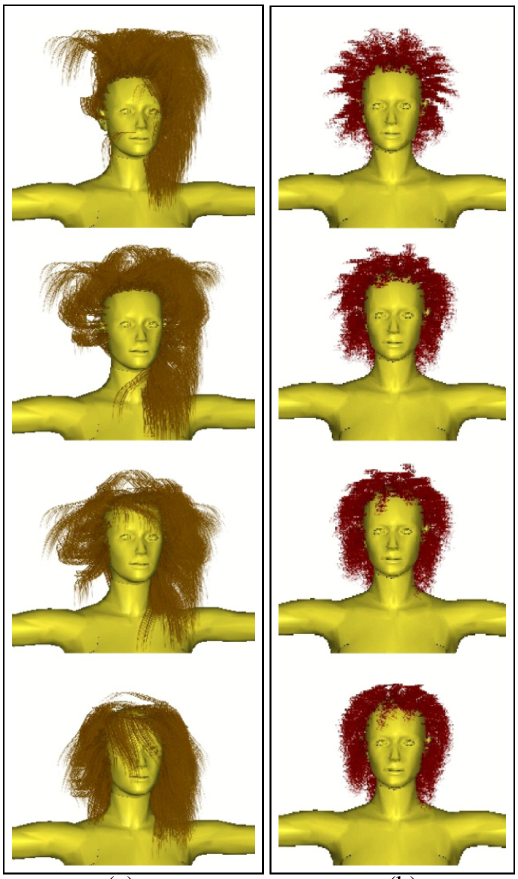

(a) (b)

Figure 5: Still frames generated with our implementation, showing long straight hair (a) and curly hair (b) while falling down after jumping.

References

[BCN03] Yosuke Bando, Bing-Yu Chen, and Tomoyuki Nishita. Animating hair with loosely connected particles. *Computer Graphics Forum*, 22(3):411–418, 2003.

[CJY02] Johnny T. Chang, Jingyi Jin, and Yizhou Yu. A practical model for hair mutual interactions. In *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 73–80, 2002.

[HT00] Sunil Hadap and Nadia-Magnenat Thalmann. State of the art in hair simulation. In *Proc. of International Workshop on Human Modeling and Animation*, pages 3–9, June 2000.

[HT01] Sunil Hadap and Nadia-Magnenat Thalmann. Modeling dynamic hair as a continuum. *Computer Graphics Forum (Proc. of Eurographics 2001)*, pages 329–338, 2001.

[iAUK92] Ken ichi Anjyo, Yoshiaki Usami, and Tsuneya Kurihara. A simple method for extracting the natural beauty of hair. In *Proc. of SIGGRAPH'92*, pages 111–120, 1992.

[KH00] Chuan Koon Koh and Zhiyong Huang. Real-time animation of human hair modeled in strips. In *Proc. of the Eurographic Workshop on Computer Animation and Simulation*, pages 101–110, 2000.

[KH01] Chuan Koon Koh and Zhiyong Huang. A simple physics model to animate human hair modeled in 2d strips in real time. In *Proc. of the Eurographic Workshop on Computer Animation and Simulation*, pages 127–138, 2001.

[LK01] Doo-Won Lee and Hyeong-Seok Ko. Natural hairstyle modeling and animation. *Graphical Models*, 63(2):67–85, 2001.

[RCT91] 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*, pages 141–148, 1991.

[WL03] Kelly Ward and Ming C. Lin. Adaptive grouping and subdivision for simulating hair dynamics. In *Proc. of the 11th Pacific Conference on Computer Graphics and Applications*, pages 5–9, 2003.

[WLL+03] Kelly Ward, Ming C. Lin, Joohi Lee, Susan Fisher, and Dean Macri. Modeling hair using level-of-detail representations. In *Proc. of the 16th International Conference on Computer Animation and Social Agents (CASA 2003)*, page 41, 2003.

[YXWY00] Xue Dong Yang, Zhang Xu, Tao Wang, and Jun Yang. The cluster hair model. *Graphical Models*, 62(2):85–103, 2000.