Graphical Models **63,** 67–85 (2001)
doi:10.1006/gmod.2001.0547, available online at http://www.idealibrary.com on IDEAL®

Natural Hairstyle Modeling and Animation

Doo-Won Lee and Hyeong-Seok Ko

Graphics & Media Lab, School of Electrical Engineering, Seoul National University, Seoul 151-742, Korea

Accepted June 1, 2001

In this paper, we develop a hairstyle modeling and animation technique specifically designed for human hairs, and we report several experimental results. Using simplified cantilever beam model and one-dimensional projective differential equations of angular momenta, we give a practical solution to the problem of enormous complexity. Even though our hair animation algorithm is an approximate solution, it includes all the relevant dynamic elements such as gravity, wind, inertia, air-resistance, hair-to-head, and hair-to-hair friction forces. Collision is an important element that makes a collection of hair strands look like hair. We develop an accurate but efficient hair-to-head and hair-to-hair collision detection and treatment algorithm. The algorithm produces quite realistic results; still it runs at an interactive speed. An interesting contribution of our algorithm is that it unifies hairstyle modeling and animation into a single equation, so that (1) hairstyling can be done under the effects of gravity and other internal or external forces, and (2) original hairstyle is more or less restored even after the initial hair is tangled by the application of external forces or head movements.

Key Words: hairstyle modeling and animation; dynamic simulation; collision detection; cantilever beam.

1. INTRODUCTION

In animating human characters, one of the most difficult parts is hair. To model human hair realistically, we need to have about 20,000 hair strands, each of which should move in harmony with the movement of head, other strands, and external forces. Due to the thin structure of hair and enormity of the problem, algorithms for modeling, animating, and rendering human hair take a quite different form than the algorithms developed to deal with other parts of human body.

Several researchers have been studying the problem with considerable success [1, 2, 5, 6, 8, 10, 11, 13, 14]. However, there are still many hurdles to overcome to provide animators with a practical tool for modeling and animating human hair.

In this paper we propose a novel hairstyle modeling and animation method. It is an improvement over the previous methods in the following aspects:

1524-0703/01 $35.00

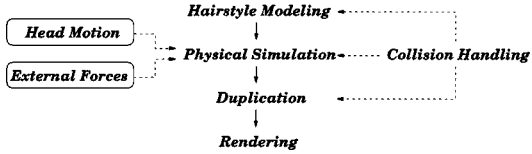

FIG. 1. Overview.

• **We unify hairstyle modeling and animation into a single principle:** in previous methods, hairstyle modeling was handled separately from animation. The result of hairstyle modeling was simply used as an initial condition of the animation. As soon as animation modifies the original style, therefore, it could not be restored afterward. In our method, both hairstyle modeling and animation are handled by the same equation. A (stationary) hairstyle is simply the result of animation when there is no external force or head movement. After hair was in motion, if the external forces go away and the head becomes stationary, the original style is more or less recovered by the *styling force*.

• **We handle hair-to-head and hair-to-hair collisions more effectively and accurately:** collision is an important element that makes a collection of hair strands look like hair. It is known to be a difficult problem to handle collisions among *so many* thin flexible structures. We propose an efficient method to handle both hair-to-head and hair-to-hair collisions that produces more accurate results than previously proposed methods.

• **We provide a rich set of hairstyling primitives:** a wide range of hairstyles can be modeled by applying our hairstyling primitives, *cut*, *waving*, and *hair-gel*.

• **Our physically based simulation includes all the physical aspects relevant to hair animation:** our hair animation employs the previous methods such as one-dimensional projective differential equations of angular momenta [1] and mass-spring-hinge system [13], which could generate the effect of *gravity* and/or *wind*. But we further generalize the dynamic equation to include the *inertia force*, hair-to-head or hair-to-hair *friction force*, and *air resistance*.

Figure 1 shows how our Hairstyle Modeling and Animation System (HMAS) works. First, a hairstyle is modeled. The model is then physically simulated based on head movement and external forces. For computational efficiency, only about 850 samples are simulated. When the simulation is complete, the result is duplicated to get the normal hair density. Finally, the hairs are rendered. Collision detection and treatment algorithm plays an important role during the hairstyle modeling, physical simulation, and duplication steps.

This paper is organized as follows: Section 2 reviews the previous work on hair modeling and animation. Section 3 explains the hair strand model for physical simulation. Section 4 introduces our new collision detection algorithm. Section 5 presents hairstyle modeling methods. Section 6 presents physical simulation of hair. Section 7 explains the rendering procedure. Section 8 reports experimental results. Finally, Section 9 concludes the paper.

2. RELATED WORK

Rosenblum *et al.* [13] used a mass-spring-hinge model to control the position and orientation of hair strand. Anjyo *et al.* [1] modeled hair with a simplified cantilever beam and used one-dimensional projective differential equations of angular momenta for dynamic hair animation. Our work is an extension of [13] and [1] in the following aspects: (1) the

FIG. 2. Our Hair Strand Model.

original hairstyle can be restored after subsequent animation, (2) the movement of head is accounted for, and (3) collision detection is more accuate and can handle hair-to-hair collisions.

Watanabe and Suenaga [14] modeled a hair strand as a chain of trigonal prisms. Using the wisp model, which generates a volume of hair by duplicating the simulated result, they generated a large number of hairs with a reduced amount of computation.

Daldegan *et al.* [4] modeled a hair as a chain of straight cylinders and used a simplified collision detection algorithm, in which human body was represented as a cylinder. They also used the wisp model to reduce calculations. For rendering hairs, they used ray tracing and shadow-buffer techniques.

Lieu-Hen Chen *et al.* [3] proposed a hair styling system that uses trigonal prism wisp model. Their model shaped a wisp using two-dimensional hair distribution map. They divided the head into eight sections to apply hairstyling primitives selectively.

Kong and Nakajima [9] proposed the *visible volume buffer* for efficient rendering. They divided the space containing hairs into cubic cells, checked how much each cell was occluded by hairs, and stored the information into the visible volume buffer. Using this buffer, they classified hairs into two layers: outermost surface hairs and inner background hairs. While the former was modeled in detail using thin hairs, the latter was modeled with coarse thick hairs. Also, they generated shadow using the opacity buffer.

3. OUR HAIR STRAND MODEL

We model a hair strand as articulated rigid segments of same length (one centimeter) and mass as shown in Fig. 2. Therefore, a longer hair consists of more segments than a shorter hair. A strand may not be an exact multiple of one centimeter. Hence the last segment may be shorter than one centimeter. Each joint has two DOFs; we assume there is no twist.

In the rendering stage, joint positions are interpreted as the control points of a B-spline curve (*RiCurve* in *RenderMan* software). In the final result, therefore, there is no discontinuities along the strand.

In this work, we use a polar coordinate system $\{H\}$ as the local coordinate system embedded in the head as shown in Fig. 3a. ϕ and θ range $[0, 2\pi]$ and $[0, \pi]$, respectively. The coordinate system is quite useful for encoding the pore positions.

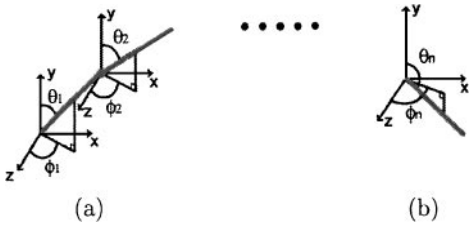

FIG. 3. Polar coordinate systems for locating pore positions and representing joints. (a) The polar coordinate system {H} embedded in the head; (b) polar coordinate systems for representing joints.

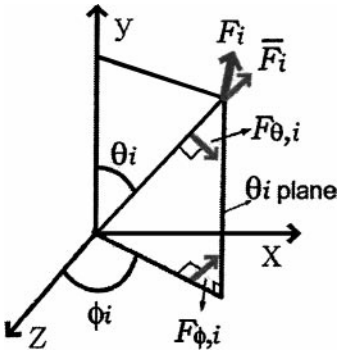

FIG. 4. Decomposing a Cartesian force into projective force components.

The joint angles between segments are also encoded by polar coordinates as shown in Fig. 3b. In the figure, ϕ_i $(i = 1, \ldots, n)$ ranges $[0, 2\pi]$, and θ_i $(i = 1, \ldots, n)$ ranges $[0, \pi]$. Origin of the coordinate system is placed at the proximal end of the strand segment. However, its orientation is always aligned with $\{H\}$.

To produce a plausible movement of hair we must account for the effects of gravity, wind, air resistance, inertia of head movement, or even the direct manipulation such as hand-stroke. Also, we must produce a tendency that the original hairstyle more or less retains through the animation steps. Considering different kinds of physical aspects to simulate and the number of hairs to animate, a physically based method seems to be a natural choice for our purpose.

Lagrangian dynamics is the normal approach to model the motion of a serial chain accurately. Considering the enormous number of hairs and the number of segments (30∼50) in each hair strands, however, Lagrangian dynamics is not a practical approach for our problem. Therefore, like in Anjyo *et al.*'s work [1], we use the simplified cantilever beam model and one-dimensional projective differential equations of angular momenta, in which the force components are considered at each segment independently. We will call the method *projective dynamics* in short. The original version in [1] simulates gravity and wind along with hair-to-head collision detection. We extend the formulation to include internal restoration property of hair (spring force), hair-to-head and hair-to-hair friction, hand-stroke, inertia, and air resistance. In the new version, hairstyling is unified with dynamic simulation so that, for example, hairstyle modeling can be done under gravity. Also, we add algorithms to deal with both hair-to-head and hair-to-hair collisions.

A Cartesian[1] force $\mathbf{F}_i$ applied at (the distal end of) Segment i can be decomposed into three components: $\bar{F}_i$ along (θ_i, ϕ_i) direction and two projective components $F_{\theta,i}$ and $F_{\phi,i}$ along the directions shown in Fig. 4. Since the force is acting on a rigid segment, only $F_{\theta,i}$ and $F_{\phi,i}$ have effects on the shape of the strand. From $\mathbf{F}_i$, the two projective components (scalar values) are obtained by

$$F_{\theta,i} = [\cos\theta_i \sin\phi_i - \sin\theta_i \cos\theta_i \cos\phi_i] \cdot \mathbf{F}_i, \tag{1}$$

$$F_{\phi,i} = [\cos\phi_i \; 0 - \sin\phi_i] \cdot \mathbf{F}_i. \tag{2}$$

Then, the relationship between the force applied at a segment and the angular

[1] We use boldface letters to denote vectors in Cartesian space.

displacement (θ_i, ϕ_i) are given by

$$\frac{d^2\theta_i}{dt^2} = \beta_i F_{\theta,i}(\theta_i, t), \tag{3}$$

$$\frac{d^2\phi_i}{dt^2} = \beta_i F_{\phi,i}(\phi_i, t), \tag{4}$$

where β_i is the reciprocal number of the inertia moment of the segment.

We use the mass-spring-hinge model to model the raw material property of a hair strand (i.e., the hair before any styling is applied). Excluding effects of all the external forces, the applied spring force $F_{\theta,i}^{SPRING}$, $F_{\phi,i}^{SPRING}$ and displacement θ_i, ϕ_i in each projective direction are related by

$$F_{\theta,i}^{SPRING} = -\kappa_s\theta_i + \kappa_d\dot{\theta}_i, \tag{5}$$

$$F_{\phi,i}^{SPRING} = -\kappa_s\phi_i + \kappa_d\dot{\phi}_i. \tag{6}$$

κ_s and κ_d are spring and damping constants, respectively, and are given by

$$\kappa_s = \begin{cases} \kappa_s^{pore} & \text{if } i = 1, \\ \kappa_s^{nonpore} & \text{if } i = 2, \ldots, n \end{cases} \tag{7}$$

$$\kappa_d = \begin{cases} \kappa_d^{pore} & \text{if } i = 1, \\ \kappa_d^{nonpore} & \text{if } i = 2, \ldots, n \end{cases} \tag{8}$$

Normally we use a bigger value for κ_s^{pore} than $\kappa_s^{nonpore}$.

On the raw strand above, we apply gravity force

$$\mathbf{F}_i^{GRAVITY} = m_i\mathbf{g}, \tag{9}$$

where m_i is the mass of ith segment (it is M except for the last segment), and $\mathbf{g}$ is the gravitational acceleration. To account for hair-to-head and hair-to-hair collision we include the friction force $\mathbf{F}_i^{FRICTION}$ and a synthetic force $\mathbf{F}_i^{SYNTHETIC}$ (Section 4). Then, we apply the styling force $\mathbf{F}_i^{STYLING}$ to deform the hair into a desirable shape (Section 5). For animation, we now include all the external forces (Section 6): $\mathbf{F}_i^{INERTIA}$, $\mathbf{F}_i^{RESISTANCE}$, $\mathbf{F}_i^{WIND}$. Thus Eqs. (3) and (4) become

$$\frac{d^2\theta_i}{dt^2} = \beta_i\left(F_{\theta,i}^{TOTAL}(\theta_i, t) + F_{\theta,i}^{FRICTION}(\theta_i, t)\right), \tag{10}$$

$$\frac{d^2\phi_i}{dt^2} = \beta_i\left(F_{\phi,i}^{TOTAL}(\phi_i, t) + F_{\phi,i}^{FRICTION}(\phi_i, t)\right), \tag{11}$$

where

$$\begin{aligned}\mathbf{F}_i^{TOTAL} = {} & \mathbf{F}_i^{SPRING} + \mathbf{F}_i^{GRAVITY} + \mathbf{F}_i^{SYNTHETIC} + \mathbf{F}_i^{STYLING} \\ & + \mathbf{F}_i^{INERTIA} + \mathbf{F}_i^{RESISTANCE} + \mathbf{F}_i^{WIND}.\end{aligned} \tag{12}$$

Equations (10) and (11) are then integrated over time steps to generate the angular variations $\theta_i(t)$ and $\phi_i(t)$.

4. COLLISION HANDLING

One of the significant issues in hair animation is detecting and handling collisions. Without it, hairs may not produce a volumetric feeling. An accurate solution can never be practical due to the enormous number and thin structure of hairs. Therefore we must find a way to approximate the situation that produces results that *look* natural. In this section, we present a new algorithm to handle both hair-to-head and hair-to-hair collisions. The algorithm is based on the *head-hull distance map* and *head-hull layering*.

4.1. *Hair-to-Head Collision*

Anjyo *et al.* [1] approximated the head hull with an ellipsoid in considering collision between hair and head. Although the method saved computation cost significantly, since it disregarded geometrical details of the head, the result could be unnatural when the ellipsoid is finally replaced with the original head.

We propose a new collision detection algorithm that uses the *hull distance map*. It is a two dimensional array H_0 of 180×360 floating point numbers that stores the displacements of the grid points (in polar coordinates) from the origin of $\{H\}$ as shown in Fig. 5.

To check a collision between a hair strand and head, we check each segment against the head; i.e., we compare the displacement d of the segment end and the value $H_0[\theta][\phi]$ of the *hull distance map* in that direction. If $d \geq H_0[\theta][\phi]$, the segment does not collide with the hull. Otherwise, the segment end is inside the hull. Once the hull distance map is prepared in off-line for the given head model, the collision can be checked with a small amount of computation.

When a collision is identified, we kinematically bend the joint (i.e., project out the segment to the hull surface with respect to the hull center) as shown in Fig. 6 so that the segment comes out of the hull. Since we are processing hair in the physical simulation context, the above kinematic adjustment at each frame should be done at the end of the dynamic simulation loop.

When hair contacts the head, it experiences friction. The friction force $\mathbf{F}_i^{FRICTION}$ is given by

$$\mathbf{F}_i^{FRICTION} = \min\{\mathbf{q}_1, \mathbf{q}_2\}, \tag{13}$$

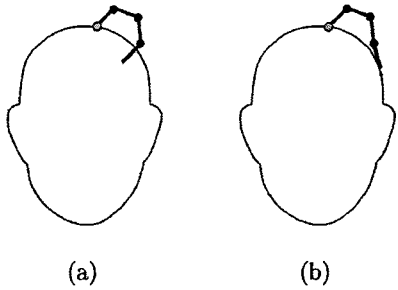

FIG. 6. Hair-to-head collision handling: (a) before collision is processed, (b) after collision is processed.

with

$$\mathbf{q}_1 = \mu\left(\mathbf{F}_i^{TOTAL} \cdot \mathbf{N}\right) \frac{\mathbf{F}_i^{TOTAL} - \left(\mathbf{F}_i^{TOTAL} \cdot \mathbf{N}\right)\mathbf{N}}{\left|\mathbf{F}_i^{TOTAL} - \left(\mathbf{F}_i^{TOTAL} \cdot \mathbf{N}\right)\mathbf{N}\right|}, \tag{14}$$

$$\mathbf{q}_2 = -\left(\mathbf{F}_i^{TOTAL} - \left(\mathbf{F}_i^{TOTAL} \cdot \mathbf{N}\right)\mathbf{N}\right), \tag{15}$$

where μ is the friction coefficient, and $\mathbf{N}$ is the unit surface normal at the current segment end. $\mathbf{q}_1$ and $\mathbf{q}_2$ are along the same direction. The min operator in Eq. (13) picks the one with the smaller magnitude. When $\mathbf{q}_2$ is picked, it is the case when the friction prevails other external forces, and the hair is stalled.

Thus the whole simulation can be summarized into the following pseudo code, in which `T` is the duration of simulation, `L` is the number of the strand samples simulated, and `n(s)` is the number of segments in strand `s`.

```
for(t=0;t<=T;t+=Δt
  for(s=1;s<=L;s++)
    for(i=1;i<=n(s);i++) {
      apply F_i^TOTAL tocurrent segment;
      check collision;
      if(collision) {
        apply F_i^FRICTION;
        do kinematic_adjustment();
      }
    }
```

4.2. *Hair-to-Hair Collision*

Detecting collision among hairs is almost impossible. We note that the most important visual effect resulting from collision among hairs is *hair volume*. Therefore, instead of trying to detect collisions, we look for a simple method that can generate the hair volume effect.

We achieve the goal by having *multiple head hull layers*. The idea is based on the observation that, unless a heavy artificial styling is applied, the hair strands whose pores are located at upper latitude generally cover the ones whose pores are located at the lower part (Fig. 7). Such phenomenon can be simulated by performing *hair-to-head collision* algorithm with hulls of different sizes; i.e., the strands in the upper region should be checked with a larger hull. We used the following equation to obtain the larger hull H_θ from the original head-hull H_0:

$$H_\theta[\theta][\phi] = \begin{cases} \left(1 + \alpha_V \frac{90-\theta}{90}\right) H_0[\theta][\phi], & \text{if } 0 \le \theta \le 90, \\ H_0[\theta][\phi], & \text{if } 90 < \theta < 180, \end{cases} \tag{16}$$

where θ is the latitude of the pore position. α_V is the *hair volume scaling factor*. By adjusting the factor we can generate compact or expanded hair shapes. Figure 7b shows a result produced by the multiple head hull layering.

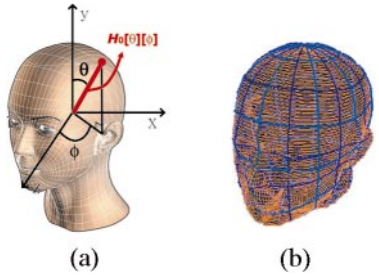

FIG. 5. Hull distance map: (a) distance in each direction, (b) hull distance map.

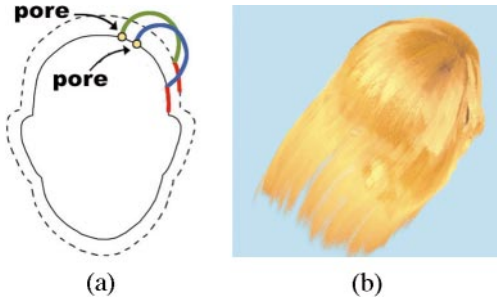

FIG. 7. Multiple head hull layering, (a) the idea, (b) an example result.

FIG. 8. Simulating collision between hair and hand.

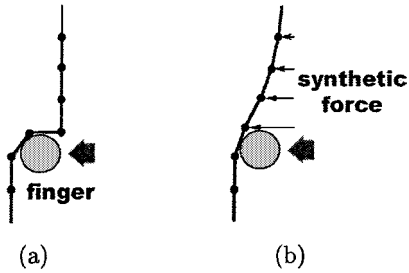

FIG. 9. Synthetic force in hair-to-object collision: (a) without synthetic force, (b) with synthetic force.

4.3. *Collision with Other Objects*

Hairs can collide also with objects other than the head. For example, a long hair might contact with the shoulder or back; hairs can be moved by hand as shown in Fig. 8. Our hull distance map in Section 4.1 can handle the situation with a small modification. We construct the hull distance map for each nonhead object and proceed similarly as the hair-to-head collision handling algorithm.

In this case, however, simply projecting the segments out of the object surface can produce an unnatural result as shown in Fig. 9a. Accurate handling of the situation (that consists of 30∼50 segments) calls for Lagrangian dynamics along with a significant amount of computation, which is an offshoot from the initial setting—projective dynamics at individual segments. In order to stay within the projective dynamics, we fix the problem by applying some synthetic forces to the segments between the contact point and the pore, as shown in Fig. 9b. If Segment j is in collision, the synthetic force $\mathbf{F}_i^{SYNTHETIC}(t)$ for Segment i $(i \le j)$ in our implementation is

$$\begin{aligned} \mathbf{F}_j^{SYNTHETIC}(t) &= k_c \dot{\mathbf{x}}^{obj}(t), \\ \mathbf{F}_i^{SYNTHETIC}(t) &= \sqrt{\frac{1}{|i-j|+1}} \cdot \mathbf{F}_j^{SYNTHETIC}(t), \end{aligned} \tag{17}$$

where $\dot{\mathbf{x}}^{obj}(t)$ is the velocity of the object pushing the hair, and k_c is a constant which is adjusted internally using the shooting method so that the contact is made between Segment j and the object surface. Figure 8 shows the result produced by Eq. (17).

5. HAIRSTYLE MODELING

Figure 10 illustrates the overview of our hairstyle modeling procedure. First, we select the region to plant hair strands (Fig. 10a). Then, by specifying the length and density, and by adding gravity, we can get the initial hair model in Fig. 10b, in which hair is covering the face in all directions. So far, we have prepared the basic hair model to which we can apply our styling primitives. The next steps are real styling steps. We can *wave* selected hairs as shown in Fig. 10c, and *cut* some portions to obtain the hairstyle shown in Fig. 10d. Also, we developed a styling primitive that simulates a hair-gel effect. The details of each styling primitive are presented in the subsequent sections.

A hairstyle is an initial state, which starts to change as it experiences head movement, external forces, or collision. However, our daily experience tells us that hairstyle modeling

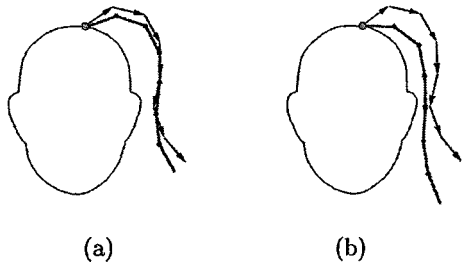

FIG. 11. Effect of styling forces: the arrows represent styling forces and connected solid line segments show the resulting hair shape, (a) when styling forces are strong, (b) when styling forces are weak.

and animation are not separate issues. The original style is more or less retained no matter how strongly or how long the original hairstyle is influenced by the dynamic parameters. To accomplish the above requirement, algorithms that integrate both hairstyling and animation should be equipped with a mechanism to *remember* and *restore* the initial hairstyle.

In this paper, hairstyle modeling is realized by the *styling force* $\mathbf{F}_i^{STYLING}$. As shown in Fig. 11, the styling force basically resembles the initial shape (or direction) of each segment. Since it is a Cartesian force, the component along the segment direction does not count in actual simulation. When the gravity and spring forces are included, the final shape of the strand changes accordingly. Magnitude of the styling force reflects the rigidity of the style. Figure 11 shows how the final shape is affected by different styling force magnitudes. Even though it is drawn with arrows of same size, the styling forces in Fig. 11a are stronger by a constant scale than the forces in Fig. 11b. Therefore, the final shape of the hair in Fig. 11a is closer to the styling force shape than the shape in Fig. 11b.

In hairstyling modeling, only spring, gravity, and styling forces are considered, with the collision handling algorithm running in the background. Once styling is complete, $\mathbf{F}_i^{STYLING}$ remains constant throughout the simulation.

Another way to realize hairstyling is to impose the initial shape with the spring model; i.e., instead of Eqs. (5) and (6), we might have used

$$F_{\theta,i}^{SPRING} = -\kappa_s(\theta_i - \theta_{i0}) + \kappa_d\dot{\theta}_i, \tag{18}$$

$$F_{\phi,i}^{SPRING} = -\kappa_s(\phi_i - \phi_{i0}) + \kappa_d\dot{\phi}_i. \tag{19}$$

Then the spring force will constantly try to restore the original shape. There are two reasons why we used styling force rather than the shape parameters θ_{i0} and ϕ_{i0}.

- **Styling force has more expressive power:** Shape parameters cannot represent hair strands that pinch or push toward the head in the initial hairstyle. On the other hand, such a situation can be quite easily represented by styling forces. Even though hairs do not penetrate the head-hull in both representations due to our collision handling mechanism, differences start to show up as head motion or external forces are exerted.
- **Styling force allows a hairstyle to be modified more intuitively:** Hairstyling modeling is an interactive process. Therefore it is crucial to provide hairstyle modelers with intuitive means to modify a previously designed hairstyle. When styling force is used, we can raise or move hairs at selective regions or over the whole area by scaling or incrementing the styling force by a constant value. The same job is much more complicated if we use the shape parameters in the joint space.

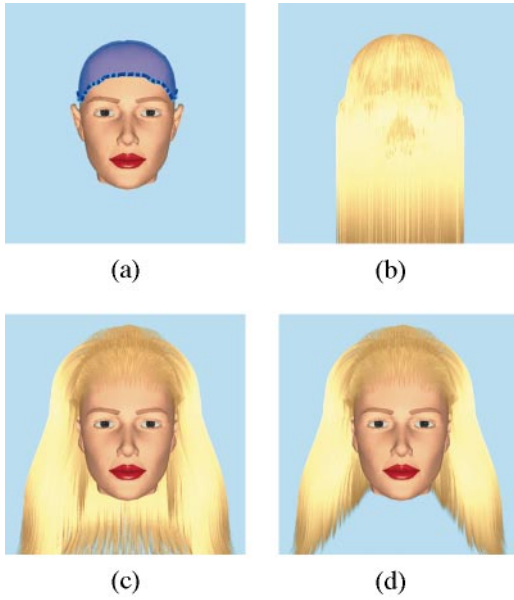

FIG. 10. Hairstyle modeling procedure: (a) planting, (b) adding gravity, (c) waving, and (d) cutting.

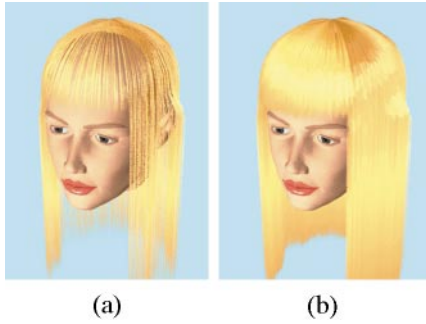

FIG. 12. Duplicating strands: (a) with 850 strands, (b) with 21,250 strands.

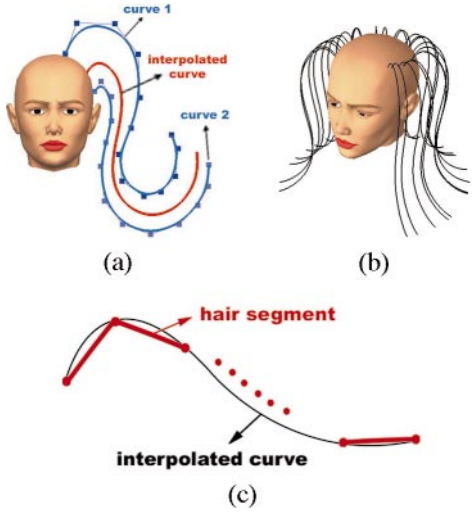

FIG. 13. Revolutionary waving: (a) boundary B-splines, (b) the result; (c) converting to line segments.

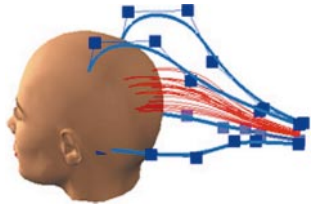

FIG. 14. Truncated-pyramid waving.

Even though restoring original style is a desirable feature, perfect restoration looks quite unnatural. If there was a strong wind, even if it is gone, we would get a tangled hair. But according to the algorithm described above, the styling force will eventually put the hairs completely back to the original configuration. The friction force $\mathbf{F}_i^{FRICTION}$ in Eq. (13) prevents such an abnormal behavior.

5.1. *Planting*

Our Hairstyle Modeling and Animation System (HMAS) allows animators to interactively select the region to plant hairs and specify the density. Then HMAS internally plants hairs according to Poisson distribution so that the hairs are uniformly distributed. In hairstyle modeling and simulation stage, we put a much smaller number (e.g., 850) of hair samples than real hair to reduce the amount of computation (Fig. 12a). The backside is not significant most of time. So we use an even smaller number of samples for the back. When the result is finally rendered, we duplicate the simulated result to about 20,000 hairs (Fig. 12b).

5.2. *Waving*

Waving together with cutting is the most basic but important primitive in contemporary hairstyling. The waving algorithm in this section can be used to model both the innate and artificially generated curly hair.

We use B-spline curves to model waved hair. We do not form a B-spline curve for each strand. Instead, we form B-spline curves to define the boundary of hair volume. Then those curves are interpolated to generate the shape of a waved strand according to the pore location.

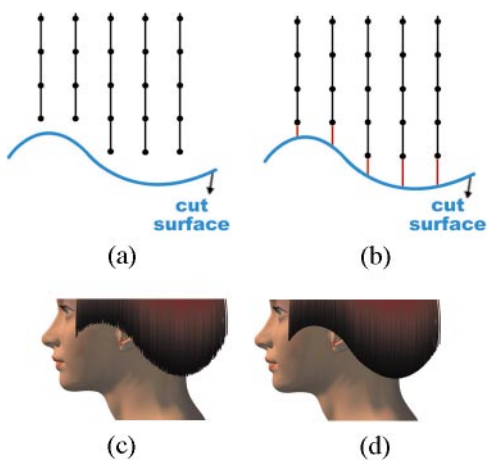

FIG. 15. Cut modes and results: (a) rough cut, (b) clean cut, (c) a rough cut result, (d) a clean cut result.

Since B-spline is an affine transformation, the curve interpolation can be achieved simply by interpolating the control points.

We provide two kinds of waving methods: *revolutionary waving* and *truncated-pyramid waving*. The revolutionary waving is initiated by specifying two curves B^1 and B^2 on a θ-plane as shown in Fig. 13a, which are revolved around the y axis of $\{H\}$ to form a hair volume. Then the shape of a hair at latitude θ (pore position) is defined by the B-spline curve B with control points B_i, where

$$B_i = \left(\frac{\theta_{B^2} - \theta}{\theta_{B^2} - \theta_{B^1}}\right) B_i^1 + \left(\frac{\theta - \theta_{B^1}}{\theta_{B^2} - \theta_{B^1}}\right) B_i^2. \tag{20}$$

Here, θ_{B^1} and θ_{B^2} are the latitudes of B^1 and B^2, respectively, and B_i^1 and B_i^2 are the control points of B^1 and B^2, respectively. Of course, the hairs with $\theta > \theta_{B^1}$ or $\theta < \theta_{B^2}$ are not affected. Figure 13b shows an example.

Note that, in the hairstyle modeling and animation stage, a strand is represented by a series of line segments. Therefore, the resulting curve must be converted to the line segments of equal length as shown in Fig. 13c.

Since revolutionary waving produces a symmetric hair, the result itself is not natural. Therefore the method is normally used to set up an initial style so that further styling primitives can apply to.

The second waving method, the *truncated-pyramid waving* is useful for more specific control. The truncated pyramid waving is initiated by specifying four cubic B-spline curves B^1, B^2, B^3, and B^4 as shown in Fig. 14. The shape of a hair at (θ, ϕ) is defined by the B-spline curve B with the control points B_i, where

$$\begin{aligned} B_i &= \sum_{k=1}^{4} w_k B_i^k, \\ w_k &= \frac{\left(\frac{1}{d_k}\right)^2}{\sum_{j=1}^{4}\left(\frac{1}{d_j}\right)^2}, \quad (1 \le k \le 4) \\ d_j &= \sqrt{(\theta - \theta_{B^j})^2 + (\phi - \phi_{B^j})^2}, \quad (1 \le j \le 4). \end{aligned} \tag{21}$$

d_j is the distance between (θ, ϕ) and $(\theta_{B^j}, \phi_{B^j})$ in polar coordinates. Therefore, w_k gives a bigger weight to a closer curve (in the polar coordinates) among the four. Of course, hairs with (θ, ϕ) outside the quadrilateral—the base of the pyramid—are not affected.

5.3. *Cutting*

We cut hair using a surface formed by projecting or revolving a B-spline curve. Once a cut surface is formed, we check if each hair intersects with the surface. When a segment of a hair strand intersects with it, the segments below the intersection point are removed.

We provide two cutting modes: the *rough cut* and *clean cut*. As shown in Fig. 15a, the rough cut simply deletes the segments below the intersection point (inclusive). Therefore the cut surface is a little rough (Fig. 15c). On the other hand, the clean cut makes the cut at the exact intersection point (Fig. 15b). Therefore, the cut surface is clean as shown in Fig. 15d.

We provide two kinds of cut surfaces: *projective surface* and *revolutionary surface*. Projective surface is formed by projecting a B-spline curve in Cartesian coordinates as shown in Fig. 16. When we cut the hair from the front, side, top views, we form a B-spline

curve in the xy plane, yz plane, zx plane, and project it along the z axis, x axis, y axis, respectively. We can cut at a selective region by controlling the size of the cut surface.

The revolutionary cut surface, which is shown in Fig. 17, is formed by defining a B-spline curve in the xy plane and revolving it around the y axis of $\{H\}$. As in waving, revolutionary cut produces a symmetric thus unnatural result. Therefore it is normally used to set up an initial model so that further styling primitives can apply to.

5.4. *Hair-Gel Effect*

When hair is pushed by a force (e.g., from hand or comb), the styling force $\mathbf{F}_{\theta,i}^{STYLING}$ pushes it back to the original shape as soon as the force disappears. But in some cases, for example, when hair-gel is applied, we might want to retain the deformed shape. We can get such an effect by updating the styling force during simulation by

$$\mathbf{F}_i^{STYLING}(t+\Delta t) = \mathbf{F}_i^{STYLING}(t) + \alpha \mathbf{F}_i^{SYNTHETIC}(t+\Delta t), \tag{22}$$

where α is a value between [0, 1]. $\alpha = 1$ produces a strong hair-gel effect, and $\alpha = 0$ produces zero hair-gel effect. The force components other than $\mathbf{F}_{\theta,i}^{SYNTHETIC}$ can also be included. According to the experiments, however, only $\mathbf{F}_{\theta,i}^{SYNTHETIC}$ turns out directly relevant to hair-gel effect.

6. DYNAMIC SIMULATION

Now we present the algorithm to animate the hairstyle prepared above. As described in Section 3, movement of individual strand is obtained by integrating Eq. (12) over time steps. The five force components $\mathbf{F}_i^{SPRING}$, $\mathbf{F}_i^{GRAVITY}$, $\mathbf{F}_i^{FRICTION}$ $\mathbf{F}_i^{SYNTHETIC}$, and $\mathbf{F}_i^{STYLING}$ have already been explained in the previous sections. In this section we explain three more force components: the inertia force $\mathbf{F}_i^{INERTIA}$, air resistance $\mathbf{F}_i^{RESISTANCE}$, and wind $\mathbf{F}_i^{WIND}$.

6.1. *Inertia Force*

Since hair is attached to the head, head motion directly affects hair movement. Among the complications, we first consider the inertia force. It is an important element to get a realistic result especially when the head makes a big motion. Inertia force is a relative concept. It greatly depends from which frame the motion is viewed. In our work, we use the coordinate system $\{H\}$ for the reference frame (at this time we consider Cartesian coordinates). When it is viewed from $\{H\}$, every Segment i in a strand experiences same inertia force which is given by

$$\mathbf{F}_i^{inertia}(t) = -m_i \ddot{\mathbf{x}}^{head}(t) - I_i \ddot{\omega}^{head}(t), \tag{23}$$

where m_i and I_i are the mass and moment of inertia of ith segment, respectively. $\ddot{\mathbf{x}}^{head}(t)$ and $\ddot{\omega}^{head}(t)$ are the translational and rotational acceleration of the head, respectively.

6.2. *Air Resistance*

Every object moving in the air experiences air resistance. This force component is not negligible when we deal with light objects such as hair. If an object is not in a very high speed, the resistance is reported to be proportional to the velocity of the object, and in the reverse direction of its movement. Therefore the resistance $\mathbf{F}_i^{RESISTANCE}$ at Segment i can be

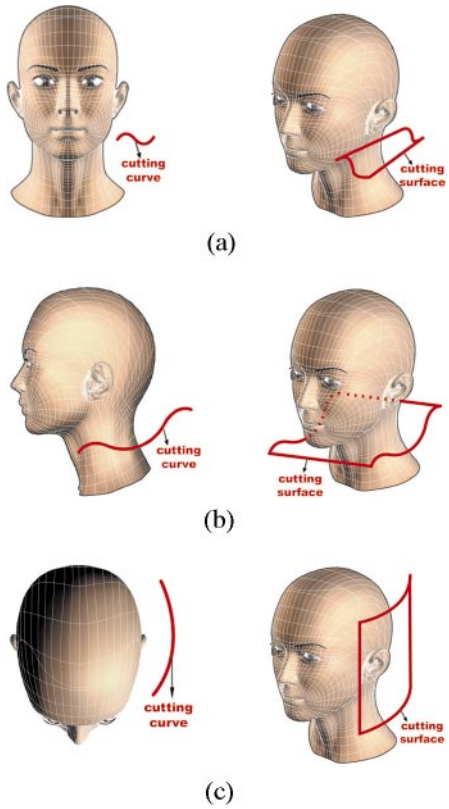

FIG. 16. Projective cutting: (a) front-view cutting, (b) side-view cutting, (c) top-view cutting.

written as

$$\mathbf{F}_i^{RESISTANCE} = -k_r \dot{\mathbf{x}}_i(t), \tag{24}$$

where k_r is the constant of air resistance, and $\dot{\mathbf{x}}_i(t)$ is the velocity of ith segment.

6.3. *Wind*

Finally, the force component $\mathbf{F}_i^{WIND}(t)$ contributed from wind is given from animators as an input.

7. RENDERING

For rendering hairs, we used *RenderMan* RiCurves primitive [12], which is a 3D curved ribbon that keeps its flat side facing the camera. It is much more efficient than using round tubes. If we light it using a standard illumination model, it does not look like hair, since a flat ribbon does not reflect light the same way a tube does. Pixar's hair shader [12] solves the problem by employing Blinn's specular model [7], and using nonuniform colors along the strand.

To obtain a more realistic result, we added two things to Pixar's hair shader:

- **We used nonuniform alpha values along the strand:** Around the hair line (the boundary between the face and hair), especially when hairs are heading upward direction,

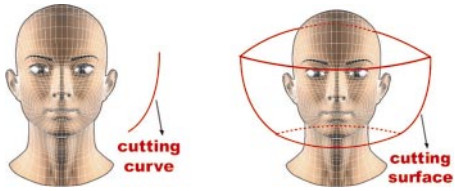

FIG. 17. Revolutionary cutting.

we can see the skin through the hairs. To account for that, we used a smaller alpha value (0.5) near the pore, and used a bigger value (0.9) at other places. The improvement due to nonuniform alpha values is shown in Fig. 18.

- **We used nonuniform reflectance along the strand:** According to observation, the mid-part of a strand is in general more reflective. So we gave bigger reflectance value at the mid-part than the end parts. The results are shown in Figs. 18a and 18b.

As pointed out in earlier sections, the simulated sample strands are duplicated to their neighbor to get a normal hair density. Duplication is done by simply copying the simulated result with necessary displacements. However, such copy-with-displacements can produce new hair-to-head or hair-to-hair collision. Since it is visually too severe, we perform the collision handling procedure again for the duplicated hairs. So the rendering algorithm can be summarized into the following pseudo code, in which `q(s)` is the number of duplication for strand `s`.

```
for(s=1;s<=L;s++)
  for(d=1;d<=q(s);d++) {
    x=duplicate_strand(s);
    for(i=1;i<=n(s);i++) {
      check collision;
      if(collision)
        do kinematic_adjustment();
    }
    render_strand(x);
  }
```

FIG. 18. Improvement due to nonuniform alpha values: (a) when the uniform alpha value 0.9 is used, (b) when nonuniform alpha values in [0.5, 0.9] are used.

FIG. 19. GUI of our hairstyle modeling and animation system.

8. RESULTS

The algorithms for hairstyle modeling, dynamic simulation, and rendering described in the previous sections were implemented on a Silicon Graphics Octane workstation. Figure 19 shows the graphical user interface of our Hairstyle Modeling and Animation System. The system was used to produce the clips (Clips#1∼#6) that are placed at `http://graphics.snu.ac.kr/research/nhma/`.

Clip#1 shows a short animation that summarizes the styling primitives: hair growing, waving, and cutting.

Clips#2 and #3 show the hair movement during a hand-stroke. Clip#2 is the result when the styling forces are weak; Clip#3 is the result when the styling forces are strong.

Clip#4 shows the hair-gel effect. α value used for the animation was 1.

In Clip#5, a woman is riding a hairy balloon. It shows how the hair movements of both woman and balloon are affected by the wind.

Clip#6 shows the hair movement during a dance motion. The dance motion was captured from a live performance. The clip demonstrates how the inertia effect is simulated by our HMAS.

9. CONCLUSION

In this paper we developed a hairstyle modeling and animation technique specifically designed for human hairs. Contributions of this paper can be summarized below:

- **We gave an interactive but practical and realistic solution to a problem of enormous complexity:** Animating a single strand of hair is already a very complex problem if accurate dynamic behavior is to be simulated. Our HMAS is an approximate/interactive solution, but all the relevant dynamic elements are in play, and results are quite realistic.

• **Our algorithm unified hairstyle modeling and animation into a single equation:** Due to the unification, (1) hairstyling now can be done under the effects of gravity, spring force, other forces generated from collisions; (2) original hairstyle is more or less restored even after the hair is tangled by the application of external forces or head movements.

• **The styling force provided an increased modeling power:** Direct control of shape parameters or geometrical shape of strands cannot model a style in which hairs have pushing or pinching forces toward the face. Our styling force can model such a case quite easily.

• **The styling force enabled intuitive hairstyle modification:** Since styling force is given in the Cartesian space, it can handle the styling operations such as raising or shifting a portion of hair much more intuitively than the shape parameters which are given in the Joint space.

• **We developed an effective hair-to-head and hair-to-hair collision detection and treatment algorithm:** Collision handling is one of the most challenging problem in dealing with hairs. Our algorithm for collision runs at an interactive speed, still it produces quite realistic hair volume effect.

• **We proposed a more realistic way to render human hair:** Nonuniform alpha and reflectance values along the strand produced more realistic result.

Projective dynamics, which considers the force components acting on each segment of a strand independently, proved quite effective in dealing with the dynamic behavior of hair strands. However, we should note it has a limitation. Intrinsically, it is not suitable for handing collisions or contacts. It is suitable to predict the behavior of a long flexible structure in which one end (pore) is fixed and all the rest part is open in space. Therefore the force components $\mathbf{F}_i^{FRICTION}$ and $\mathbf{F}_i^{SYNTHETIC}$, which were considered to handle hair-to-head or hair-to-hair collision are based on heuristics. A simple extension of Eq. (12) for simulating hair-band or hairpin effect may not work, since the case is too different from the designed projective dynamics.

ACKNOWLEDGMENTS

This work was supported by Brain Korea 21 Project. This work was also partially supported by ASRI (Automation and Systems Research Institute), Seoul National University.

REFERENCES

1. K.-I. Anjyo, Y. Usami, and T. Kurihara, A simple method for extracting the natural beauty of hair, in *Computer Graphics (SIGGRAPH '92 Proceedings), July 1992*. pp. 111–120.
2. A. Bruderlin, A method to generate wet and broken-up animal fur, in *Pacific Graphics '99, 1999*, pp. 242–249.
3. 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* **4**, 1999.
4. A. Daldegan, N. Magnenat Thalmann, T. Kurihara, and D. Thalmann, An integrated system for modeling, animating and rendering hair, in *Eurographics '93, 1993*, pp. 211–221.
5. A. Van Gelder and J. Wilhelms, An interactive fur modeling technique, in *Graphics Interface '97 Proceedings, May 1997*, pp. 181–188.
6. D. B. Goldman, Fake fur rendering, in *Computer Graphics (SIGGRAPH '97 Proceedings), 1997*, pp. 127–134.
7. J. Blinn, Models of light reflection for computer synthesized pictures, in *Computer Graphics (SIGGRAPH '77 Proceedings), July 1977*, pp. 192–198.
8. J. T. Kajiya and T. L. Kay, Rendering fur with three dimensional textures, in *Computer Graphics (SIGGRAPH '89 Proceedings), July 1989*, pp. 271–280.

9. W. Kong and M. Nakajima, Visible volume buffer for efficient hair expression and shadow generation, in *Computer Animation '99, May 1999.*
10. A. M. LeBlanc, R. Turner, and D. Thalmann, Rendering hair using pixel blending and shadow buffers, *J. Visual. Comput. Anim.* **2**, 1991, 92–97.
11. K. Perlin and E. M. Hoffert, Hypertexture, in *Computer Graphics (SIGGRAPH '89 Proceedings), July 1989*, pp. 253–262.
12. Pixar, *PhotoRealistic RenderMan Toolkit Application* Note No. 19, Using the RiCurves Primitive.
13. R. E. Rosenblum, W. E. Carlson, and E. Tripp III, Simulating the structure and dynamics of human hair: Modelling, rendering and animation, *J. Visual. Comput. Anim.* **2**, 1991, 141–148.
14. Y. Watanabe and Y. Suenaga, A trigonal prism-based method for hair image generation, *IEEE Comput. Graphics Appl.* **12**(1), Jan 1992, 47–53.