Eurographics/ACM SIGGRAPH Symposium on Computer Animation (2005)
K. Anjyo, P. Faloutsos (Editors)

Simulating Complex Hair with Robust Collision Handling

Byoungwon Choe[1] Min Gyu Choi[2] Hyeong-Seok Ko[1]

[1]Seoul National University [2]Kwangwoon University

Abstract

We present a new framework for simulating dynamic movements of complex hairstyles. The proposed framework, which treats hair as a collection of wisps, includes new approaches to simulating dynamic wisp movements and handling wisp-body collisions and wisp-wisp interactions. For the simulation of wisps, we introduce a new hair dynamics model, a hybrid of the rigid multi-body serial chain and mass-spring models, to formulate the simulation system using an implicit integration method. Consequently, the simulator can impose collision/contact constraints systematically, allowing it to handle wisp-body collisions efficiently without the need for backtracking or sub-timestepping. In addition, the simulator handles wisp-wisp collisions based on impulses while taking into account viscous damping and cohesive forces. Experimental results show that the proposed technique can stably simulate hair with intricate geometries while robustly handling wisp-body collisions and wisp-wisp interactions.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically Based Modeling; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

The synthesis of hair has been a key stumbling block in generating realistic humans in computer graphics. The modeling of diverse hairstyles by controlling individual hair strands is a difficult problem. Moreover, simulating such hairstyles during dynamic movements in the presence of gravity, wind, and head motion is even more complicated.

An approach that has proved successful in generating a wide range of human hairstyles is to treat the hair as a collection of *wisps* rather than as individual strands [WS92]. This simplification makes the hairstyling process interactive and manageable because the simulator needs only to compute the behavior of a collection of interacting wisps rather than that of tens of thousands of individual strands [CK05a, KN00, KN02, XY01]. In this paper, we develop a new simulation algorithm based on the wisp concept. In developing this wisp-based hair simulator, we address the following three major problems: (1) simulating dynamic movements of wisps; (2) handling collisions between wisps and the body; and (3) modeling hair-hair interactions, which are treated in terms of wisp-wisp interactions.

A model that has been widely used to simulate hair strands is the mass-spring-hinge system pioneered by Rosenblum et al. [RCT91]. This model cannot, however, properly represent torsional motions of a strand. The two-dimensional projection method, introduced by Anjyo et al. [AUK92], has been another popular technique in hair simulation [DTKT93, LK01, WL03, WGL04]. However, this method also does not provide a true representation of three-dimensional motions. One method that can properly represent three-dimensional motions is the rigid multi-body serial chain model, employed by Hadap and Magnenet-Thalmann [HMT01]. In this model, high bending and torsional stiffness is required to maintain the curved rest shape under high gravity, and it calls for the use of implicit integration for stability. However, implicit integration is difficult to realize using conventional articulated rigid body dynamics, as described in [Had03]. To resolve this problem, here we introduce a new hair dynamics model that is a hybrid of the rigid multi-body serial chain and mass-spring models. In the proposed model, a linear spring and an angular spring are used between each pair of links instead of a hard-constrained joint. This model allows implicit integration, and thus it can rapidly and stably simulate hair structures with intricate geometries.

In most previous hair simulations, hair-body collision response has been implemented using a penalty method. However, penalty methods can give rise to instabilities in systems undergoing complex motions because they do not enforce constraints exactly and increase the stiffness of the system [BW98, Had03]. To accurately and robustly enforce the

Computer Animation 2005, 29–31 July 2005, Los Angeles CA

hair-body collision/contact, we employ a constraint-based method (instead of a penalty method) by incorporating velocity constraints into the implicit integration scheme. This constraint-based method also enables us to develop a compact algorithm that systematically handles collisions without sub-timestepping or backtracking.

Handling hair-hair interactions has been recognized as a critical issue in simulating hair. To handle hair-hair collisions efficiently, Lee and Ko [LK01] proposed the use of head-hull layers, and Chang et al. [CJY02] used triangle strips. Hadap and Magnenat-Thalmann [HMT01] formulated a continuum model to solve hair-hair collisions as well as hair-body and hair-air interactions within a single framework. In contrast to the above methods, Plante et al. [PCP02] and Bertails et al. [BKCN03] handled hair-hair collisions at the wisp level. Here we also handle hair-hair collisions through the modeling of wisp-wisp collisions. However, to more realistically model wisp interactions without undesired oscillations, we exploit the impulse (due to collision) through a phenomenological use of the coefficient of restitution. In addition, we take into account a viscous drag proportional to the hair density and a cohesive force to preserve the initial hairstyle.

Putting it all together, our framework for hair simulation is comprised of the following three components: the hybrid hair dynamics model, wisp-body collision response using constrained dynamics, and hair-hair interactions calculated using an empirical impulse model. The significance of this paper lies in our adaptation/generalization of well-established techniques for simulating solids and cloth to the simulation of hair, all within the existing concept of wisp-based hair simulation. This enables us to establish a powerful framework for simulating long complex hair. The new framework is based on backward Euler integration, and hence is stable in the simulation of the dynamic movements of complex hair. Moreover, it exhibits robust collision handling, owing to the use of a well-established constraint solver that computes global system states in accordance with complex collision situations.

2. Dynamic Simulation of a Wisp

Typical mass-spring models used in hair simulations are easy to implement and computationally efficient; however, they are ill-suited to representing torsional motions, which are intrinsic motions in the restoration of hair to its rest shape. On the other hand, rigid multi-body serial chain models can support torsional motions, but it is difficult to impose constraints such as contact with an object. Furthermore, the resulting systems are stiff due to the *hard* constraints at the joints, which precludes the use of large time steps.

Here we propose a new physically based technique for dynamic simulation of a thin flexible structure, represented as a rigid multi-body serial chain (See Figure 1). Instead of imposing hard constraints that require reduced coordinate formulations [Fea87] or Lagrange multipliers [Bar96], we impose *soft* constraints by attaching a linear spring and an angular spring between each pair of links. This hybrid scheme significantly improves simulation speed while supporting torsional motions because the resulting system of equations can be solved by an implicit backward Euler method.

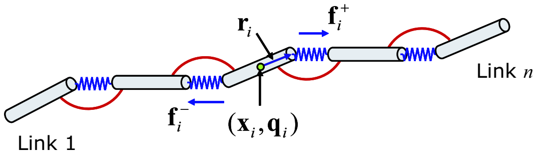


Figure 1: *The wisp model.*

In the present work, the above model is used only for simulating wisps (or more precisely, their skeletons); it could, however, be adapted to the simulation of individual strands by appropriate tuning of the dynamic parameters such as the mass and stiffness. Once we obtain the dynamic movement of a wisp skeleton, we generate the motion of the individual strands constituting the wisp using the statistical model proposed by Choe and Ko [CK05a], with a minor modification in Section 2.5.

2.1. Linear Springs

For the position constraint at the joint, we attach a linear spring between each pair of links. Let $\mathbf{x}_i$ and $\mathbf{R}_i$ be the center of mass (COM) and the rotation matrix of Link i, respectively. For simplicity, we assume that the COM lies at the center of the link, and denote the vector from the COM to the next joint by $\mathbf{r}_i$, as shown in Figure 1. Then, Link i is influenced by the two linear spring forces (except for the two extreme links):

$$\mathbf{f}_i^- = k_L\{(\mathbf{x}_{i-1} + \mathbf{R}_{i-1}\mathbf{r}_{i-1} - (\mathbf{x}_i - \mathbf{R}_i\mathbf{r}_i)\},$$
$$\mathbf{f}_i^+ = k_L\{(\mathbf{x}_{i+1} - \mathbf{R}_{i+1}\mathbf{r}_{i+1} - (\mathbf{x}_i + \mathbf{R}_i\mathbf{r}_i)\},$$

where k_L is a linear stiffness coefficient, $\mathbf{f}_i^-$ is the elastic force due to the spring connected to Link $(i-1)$, and $\mathbf{f}_i^+$ is the elastic force due to the spring connected to Link $(i+1)$. As a consequence, Link i experiences the force,

$$\mathbf{F}_i = \mathbf{f}_i^- + \mathbf{f}_i^+,$$

and also the accompanying torque,

$$\tau_i^L = (\mathbf{R}_i\mathbf{r}_i) \times (\mathbf{f}_i^+ - \mathbf{f}_i^-). \qquad (1)$$

2.2. Angular Springs

For torsional restitution we attach an angular spring between each pair of links, which produces a torque proportional to the angular difference. Let $\mathbf{q}_i$ be the quaternion representation of the rotation matrix $\mathbf{R}_i$. Then, Link i experiences the elastic torque

$$\tau_i^A = k_A\mathbf{R}_i\{\log(\mathbf{q}_i^{-1}\mathbf{q}_{i-1}) + \log(\mathbf{q}_i^{-1}\mathbf{q}_{i+1})\}, \qquad (2)$$

where k_A is an angular stiffness coefficient, and $\log(\cdot)$ is the logarithm map of the quaternion. Therefore, the total torque τ_i exerted on Link i is given by $\tau_i = \tau_i^L + \tau_i^A$ from Equations (1) and (2).

2.3. Governing Equations

To achieve numerical stability and increase the simulation speed, we adopt an implicit scheme by employing an incremental formulation of orientational changes, $\mathbf{q}_i(t+h) = \mathbf{q}_i(t)\exp(\mathbf{u}_i/2)$, which corresponds to the positional counterpart, $\mathbf{x}_i(t+h) = \mathbf{x}_i(t) + \Delta\mathbf{x}_i$. Let $\mathbf{M}_i$ and $\mathbf{H}_i$ be the mass matrix and the inertia matrix of Link i, respectively. Then, the governing equations for Link i can be written as

$$\mathbf{M}_i\dot{\mathbf{v}}_i = \mathbf{F}_i + \sum_j \left[\frac{\partial \mathbf{F}_i}{\partial \mathbf{x}_j}\Delta\mathbf{x}_j + \frac{\partial \mathbf{F}_i}{\partial \mathbf{u}_j}\Delta\mathbf{u}_j\right], \tag{3}$$

$$\mathbf{H}_i\dot{\omega}_i = \tau_i + \sum_j \left[\frac{\partial \tau_i}{\partial \mathbf{x}_j}\Delta\mathbf{x}_j + \frac{\partial \tau_i}{\partial \mathbf{u}_j}\Delta\mathbf{u}_j\right] + \mathbf{L}_i \times \omega_i, \tag{4}$$

where $\Delta\mathbf{x}_i = h(\mathbf{v}_i + \Delta\mathbf{v}_i)$ and $\Delta\mathbf{u}_i = h\mathbf{R}_i^{-1}(\omega_i + \Delta\omega_i)$. For brevity, we have omitted damping terms proportional to the linear velocity $\mathbf{v}_i$ of Link i and its angular velocity $\mathbf{w}_i$. Refer to [CK05b] for the detailed derivation of the system equations. Equations (3) and (4) are implicit forms of $\dot{\mathbf{P}}_i = \mathbf{F}_i$ and $\dot{\mathbf{L}}_i = \tau$, respectively, where $\mathbf{P}_i = \mathbf{M}_i\mathbf{v}_i$ and $\mathbf{L}_i = \mathbf{H}_i\omega_i$.

By manipulating the above equations, we obtain the following governing equation:

$$\mathbf{A}\mathbf{y} = \mathbf{b}, \tag{5}$$

where $\mathbf{y}_i = \left[\Delta\mathbf{v}_i^{\mathrm{T}} \quad \Delta\omega_i^{\mathrm{T}}\right]^{\mathrm{T}}$,

$$\mathbf{A}_{ij} = \begin{bmatrix} \mathbf{I} - h^2\mathbf{M}_i^{-1}\frac{\partial \mathbf{F}_i}{\partial \mathbf{x}_j} & -h^2\mathbf{M}_i^{-1}\frac{\partial \mathbf{F}_i}{\partial \mathbf{u}_j}\mathbf{R}_i^{-1} \\ -h^2\mathbf{H}_i^{-1}\frac{\partial \tau_i}{\partial \mathbf{x}_j} & \mathbf{I} - h^2\mathbf{H}_i^{-1}\frac{\partial \tau_i}{\partial \mathbf{u}_j}\mathbf{R}_i^{-1} \end{bmatrix}, \quad \text{and}$$

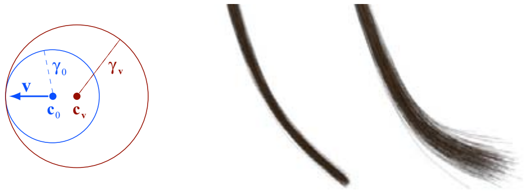
$$\mathbf{b}_i = h\sum_j \begin{bmatrix} \mathbf{M}_i^{-1}(\mathbf{F}_i + h\frac{\partial \mathbf{F}_i}{\partial \mathbf{x}_j}\mathbf{v}_j + h\frac{\partial \mathbf{F}_i}{\partial \mathbf{u}_j}\mathbf{R}_i^{-1}\omega_j) \\ \mathbf{H}_i^{-1}(\tau_i + \mathbf{L}_i \times \omega_i + h\frac{\partial \tau_i}{\partial \mathbf{x}_j}\mathbf{v}_j + h\frac{\partial \tau_i}{\partial \mathbf{u}_j}\mathbf{R}_i^{-1}\omega_j) \end{bmatrix}.$$

Note that the (6×6) block matrix $\mathbf{A}_{ij}$ is non-zero if and only if $j \in [i-1, i+1]$. As a consequence, the linear system is (6×6) block tri-diagonal, and hence it can be solved directly in $\mathrm{O}(n)$, where n is the number of links. The computation of jacobian matrices is addressed in Appendix A.

2.4. Constraints

In our implicit formulation, incorporation of linear and angular velocity constraints is straightforward. To impose the velocity change of Link i to be $\mathbf{z}_i = \left[\Delta\mathbf{v}_i^{\mathrm{T}} \quad \Delta\omega_i^{\mathrm{T}}\right]^{\mathrm{T}}$, we modify the inverses mass matrix $\mathbf{M}_i^{-1}$ and the inverse inertia matrix $\mathbf{H}_i^{-1}$ following the approach of Barafff and Witkin; refer to [BW98] for the details of the modifications. The linear system that incorporates the constraints can ultimately be written as

$$\hat{\mathbf{A}}\mathbf{y} = \hat{\mathbf{b}} + \mathbf{z}, \tag{6}$$



(a) cross sections (b) *w/o* modification (c) *w/* modification

Figure 2: *Modification of the wisp shape according to the velocity.*

where $\hat{\mathbf{A}}$ and $\hat{\mathbf{b}}$ are the modified system matrix and vector, respectively, from Equation (5).

2.5. Shape of the Wisp in Motion

In forming a *static* wisp, Choe and Ko [CK05a] generated individual strands by replicating the shape of the master strand with some statistical variations. A *moving* wisp obviously calls for further variations, without which the results do not look acceptable. Generating such motions should, in principle, be based on physically based approaches—e.g., computing hair-air interactions [HMT01]. However, such approaches are incompatible with the use of wisps; hence in the present work we resort to a simple method.

We account for the effect of the velocity by making the wisp occupy an expanded volume as shown in Figure 2. We use the formula $\gamma_{\mathbf{v}} = (1+\sigma|\mathbf{v}|)\gamma_0$ for the radius $\gamma_{\mathbf{v}}$ of the cross section of a wisp moving at velocity $\mathbf{v}$, where γ_0 is the radius at the rest state, and σ is a user-controlled function increasing monotonically towards the tip of the wisp. We also move back the center of the cross section using $\mathbf{c}_{\mathbf{v}} = \mathbf{c}_0 - (\gamma_{\mathbf{v}} - \gamma_0)\mathbf{v}/|\mathbf{v}|$, so that the two circles coincide at the front.

3. Hair-Body Collision Handling

We propose a compact hair-body collision handling method that exploits the simple nature of hair, compared to solids or cloth [GBF03, BFA02, BWK03]. Distinctive features of our method are that (1) it does not employ sub-timestepping, (2) it does not involve any backtracking, and (3) it executes either a collision (colliding contact) response step or a contact (resting contact) response step but *not both*.

Our collision handling algorithm is designed so that interpenetrations do not occur in principle. To achieve this, we predict whether the collisions occur in the next time step, and resolve them by running the constrained dynamics proposed by Baraff and Witkin [BW98] when computing the system states of the next time step. In contrast to the penalty method, our approach finds the next system state by solving the constrained dynamics globally; therefore, we obtain more accurate and stable behavior. As a result, unexpected interpenetrations occur only in truly exceptional situations, as is evident in the results obtained during the production of the accompanying animations (see Table 1).

Animation	#1	#2	#3	#4
#segments	2,687	3,136	3,583	5,672
N_1	8.8 (0.33%)	5.6 (0.18%)	10.0 (0.28%)	7.3 (0.13%)
N_2	2.9 (0.11%)	0.6 (0.02%)	3.2 (0.09%)	2.2 (0.04%)

Table 1: *Summary of interpenetrations that occurred during the production of the accompanying animations: N_1 is the average number of segments that undergo unexpected interpenetrations; N_2 is the average number of segments that undergo interpenetrations of depth > 0.2 cm.*

Given the rarity of such interpenetrations, the fact that in most cases the interpenetration depth is sufficiently shallow (< 0.2 cm; see Table 1) that the interpenetration cannot be discerned by the viewer, and that unexpected interpenetrations are mostly hidden by the hair stacked above them, we chose to handle the unexpected interpenetrations at the *rendering stage* by drawing the wisps somewhat squashed. This approach removes the complications associated with interpenetration from the collision handling procedures, and thus allows us to achieve the distinctive features outlined at the beginning of this section. Moreover, those interpenetrations are resolved when processing the next time step without causing any further trouble by the above collision handling algorithm based on the constrained dynamics.

3.1. Overview

We assume that the gross body motion is already given, and treat hair movement as a secondary motion. The algorithm starts by first moving the gross body to the pose of the next time step.

To check whether the wisp segment will be involved in collisions in the next time step, we first *predict* the velocity and position of the wisp segments by a simplified dynamics:

$$\tilde{\mathbf{v}}_i^1 = \frac{h}{m_i}\mathbf{F}_{total}^0 + \mathbf{v}_i^0, \quad \tilde{\mathbf{x}}_i^1 = h\tilde{\mathbf{v}}_i^1 + \mathbf{x}_i^0$$

where $\mathbf{F}_{total}^0$ is the summation of all the forces acting on Segment i, and m_i is the mass of Segment i.[†] In this system state estimation for collision handling, Equation (4) is disregarded and the torque is not considered. The result of this estimation is used to identify imminently occurring collisions or contacts in the next time step, thus determining the collision mode as one of Free, Colliding, or Contacting.

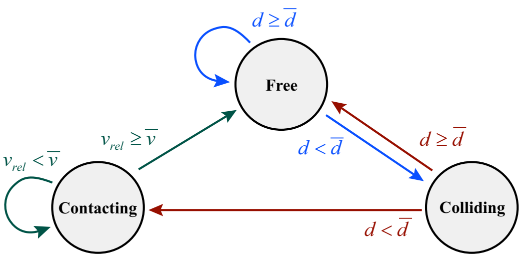


Figure 3: *Three collision modes and their transitions.*

Based on the current collision mode and the relative position/velocity of the wisp segment with respect to the body, we determine the next collision mode. The transitions among the modes are illustrated in Figure 3, in which d and v_{rel} are the proximity distance and the relative velocity, respectively; both are measured along the direction normal to the body surface, for example $v_{rel} = (\tilde{\mathbf{v}}^1 - \mathbf{v}_{obj}^1) \cdot \mathbf{n}$, where $\mathbf{v}_{obj}^1$ and $\mathbf{n}$ are the velocity and normal at the body contact point. $\bar{d}$ and $\bar{v}$ are the user-specified threshold values of d and v_{rel}, respectively. To find the proximity distance, we measure the distance by coupling the efficient use of distance fields and the accurate computation of the distance to the mesh surface. If the next mode turns out to be either Colliding or Contacting, then the collision handling procedure (Section 3.2) or the contact handling procedure (Section 3.3) is executed.

3.2. Wisp-Body Collision

When a segment is in Colliding mode, we resolve it by applying an impulse to the center of the segment. When the coefficient of restitution is ε, the effect of an impulse in the normal direction is described by

$$v_{rel}^+ = -\varepsilon v_{rel}^-, \tag{7}$$

where v_{rel}^- and v_{rel}^+ are the relative velocities before and after the collision. Ideally the velocity should be changed at the exact moment of collision, which can only be achieved by taking sub-timesteps. Instead of using sub-timesteps, we make the approximations

$$\begin{array}{l} v_{rel}^- = v^0 - v_{obj}^0 \\ v_{rel}^+ = v^1 - v_{obj}^1 \end{array}, \tag{8}$$

where all the variables are normal components of the velocities. Substituting Equation (8) into Equation (7), we obtain the recommendation for the normal velocity change Δv_A that accounts for the collision that will occur by

$$\Delta v_A = v^1 - v^0 = \varepsilon(v_{obj}^0 - v^0) + v_{obj}^1 - v^0.$$

The above collision response does not guarantee that the segment evolves to a collision free location at the next time step. In the backward Euler scheme, we can guarantee it by setting Δv as

$$\Delta v = max\{\Delta v_A, \Delta v_B\},$$

[†] In this paper we use the following notational convention when it is applicable. A boldface letter as in $\mathbf{p}_i^*$ represents a vector in which the subscript represents the segment number and the superscript represents the time steps (0 means the current and 1 means the next time step). A non-boldface letter as in p_i^* represents a scalar, usually the normal component of $\mathbf{p}_i^*$. A bar on a scalar as in $\bar{p}$ indicates that it is the threshold value.

where Δv_B is the minimum velocity update for the segment to lie outside of the surface and is computed by

$$\Delta v_B = \frac{\overline{d} - d}{h} - v^0. \tag{9}$$

Finally, we realize the above velocity change Δv by setting $\mathbf{z}_i = \Delta v \cdot \mathbf{n}$ and making appropriate modifications to $\mathbf{M}_i^{-1}$ in Equation (6). By these arrangements, the normal movements occur as constrained while the movements along the other unconstrained directions are in accordance with the current dynamic situation.

3.3. Wisp-Body Contact

To resolve the Contacting mode, we use the same procedure as is used for resolving the Colliding mode, with the difference that in this case we directly compute Δv using Equation (9) to make the segment maintain contact with the body surface. This constraint is realized using the same procedure as that described in the previous section.

In the Contacting mode, we must additionally consider friction. We formulate the friction by a damping force proportional to the tangential component of the velocity:

$$\mathbf{F}_{friction} = -\mu\left(\mathbf{v}_{rel} - (\mathbf{v}_{rel} \cdot \mathbf{n})\,\mathbf{n}\right),$$

where μ is the friction coefficient. Note that the effect of the friction force is realized by adding this force and its Jacobians to Equation (3). The Coulomb friction force would be the most physically reasonable force to use, but the Jacobians of this force cannot be obtained analytically. Thus, we opt instead for the damping force to ensure stability.

4. Hair-Hair Interactions

Hair strands undergoing dynamic movements are constantly interacting with each other. Thus, if this behavior is not correctly modeled, the result of the simulation will not look like real hair. Modeling the individual strand-strand interactions is obviously intractable. In this paper, we model only wisp-level collisions. Plante et al. [PCP02] also proposed a wisp-based collision handling technique using a deformable wisp envelope; however their method gave rise to instabilities in the mass-spring system used to model the wisp envelope. Instead, we introduce the *admissive cylinder model*, and handle the collisions based on impulses. We also introduce a viscous drag force related to the density of the hair medium involved in the collisions and a cohesive force to preserve the initial configuration.

4.1. Wisp-Wisp Collisions

Collisions occurring between two objects can be viewed as a conflict to occupy the same location in space. In rigid bodies, the conflict is resolved by moving them away from each other. In deformable solids, the conflict is momentarily resolved by deforming the shape. For the case of two colliding wisps, we propose the *admissive cylinder model* that handles the collision by allowing the colliding segments to overlap rather than by deforming them, as shown in Figure 4.

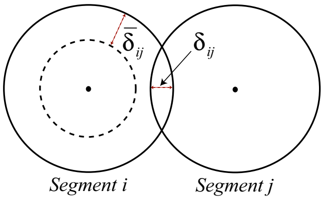


Figure 4: *The admissive cylinder model.*

In the admissive cylinder model, we must decide upon the permitted interference depth $\overline{\delta_{ij}}$ between Segments i and j. It has been observed that the interference is deeper when the two colliding wisps are in a parallel rather than a perpendicular configuration with respect to one another [PCP02, CJY02]. In fact, making $\overline{\delta_{ij}}$ a function of the relative orientations of the segments produces visually better results than setting it to a constant value. Here we use

$$\overline{\delta_{ij}} = \left(1 + g|\mathbf{t}_i \cdot \mathbf{t}_j|\right)\lambda, \tag{10}$$

where $\mathbf{t}_i$ and $\mathbf{t}_j$ are the tangential directions of the interfering segments, and g and λ are user-provided coefficients.

When a set S of segments is overlapping Segment i, we resolve the situation by applying *penalty forces*. Let δ_{ij} be the current interpenetration depth of Segment j with Segment i. According to δ_{ij}, we partition S into S_1 and S_2 and compute the penalty forces in different ways, where S_1 consists of the segments such that $\delta_{ij} \leq \overline{\delta_{ij}}$ (within the permitted threshold), and S_2 consists of the remaining segments (causing *undesired* collisions to be resolved immediately).

For each Segment j in S_1, we calculate the penalty force $\mathbf{F}_{ij}^{WW}$, which will be applied to Segment i, utilizing *collision impulses*; more precisely, we first compute the velocity change based on the collision impulse then compute the force from the velocity change. Let the normal vector be $\mathbf{n}_{ij} = \mathbf{d}_{ij}/|\mathbf{d}_{ij}|$ where $\mathbf{d}_{ij}$ is the vector from the center of Segment j to the center of Segment i. Then using the same assumptions in Equations (7) and (8), we can obtain $v_i^1 - v_j^1 = -\varepsilon(v_i^0 - v_j^0)$ along the normal direction. If we suppose that the masses of the Segments i and j are the same, we can get $v_i^1 + v_j^1 = v_i^0 + v_j^0$ from the law of momentum conservation. Derived from the above two equations, we can compute Δv_{ij}^{WW} the velocity change of Segment i due to Segment j by

$$\Delta v_{ij}^{WW} = v_i^1 - v_i^0 = \frac{1}{2}\left(1+\varepsilon\right)\left(v_j^0 - v_i^0\right).$$

Here, we use an empirical coefficient of restitution $\varepsilon = \delta_{ij}/\overline{\delta_{ij}} - 1$ that varies according to the current overlapping depth. The negative ε value gradually decreases the velocity of the interpenetrating segment as δ_{ij} approaches to $\overline{\delta_{ij}}$. Finally, we convert Δv_{ij}^{WW} into the form of a force using $\mathbf{F}_{ij}^{WW} = (m_i/h)\Delta v_{ij}^{WW} \cdot \mathbf{n}_{ij}$. Guendelman et al. [GBF03] previously used a varying restitution coefficient to handle contact between rigid bodies; however, whereas they used it to prevent incorrect rattling, here we use it to make the simulated hair appear non-rigid rather than to ensure convergence.

For the segments in S_2, the interpenetrations are deeper than the threshold $\overline{\delta_{ij}}$, and they should be resolved immediately. To achieve this, we use

$$\mathbf{F}_{ij}^{WW} = \frac{m_i}{2} \frac{\delta_{ij} - \overline{\delta_{ij}}}{h^2} \mathbf{n}_{ij}.$$

Then, the overall penalty force, which will be applied to Segment i, is computed by summing up all the forces from the overlapping segments by $\mathbf{F}_i^{WW} = \sum_{j \in S} \mathbf{F}_{ij}^{WW}$.

4.2. Viscous Drag

Although segment overlap is permitted, the movements of the mutually overlapping segments should not be as free as in the non-overlap case. This can be modeled by introducing viscous drag into the movement of the wisp. An object moving at a velocity $\mathbf{v}$ is known to experience a viscosity that is proportional to $\rho\mathbf{v}$, where ρ is the density of the medium. In our case, we consider that the segments in S create a dense medium that dampens the movement of Segment i. We apply the following viscous force

$$\mathbf{F}_{vis} = -k_{vis}(\rho_a + \rho_h)\mathbf{v}_i, \tag{11}$$

where ρ_a is the density of air, and ρ_h is the density created by the *hair medium*. We compute ρ_h (influencing the Segment i) using the approximation of

$$\rho_h = \sum_{j \in S} \frac{\rho_j}{4} \left(\frac{\delta_{ij}}{r_i} \right)^2,$$

where ρ_j is the density of Segment j at the initial state and r_i is the radius of the Segment i. Again, the effect of this force is implemented by putting the force and its Jacobians into Equation (3).

4.3. Cohesive Force

Human hair has a tendency to return to its initial configuration, due to factors such as static charge or hair spray. This tendency can be incorporated into the model by including a cohesive force between wisp segments. Based on the work of Chang et al. [CJY02], we apply a cohesive force that is proportional to the lateral distance of two wisps up to a threshold distance. As the distance increases beyond the threshold, the cohesive force starts to decrease linearly until it completely vanishes to zero.

5. Results

We implemented the technique presented in this paper on a computer with a 3.0 GHz Pentium IV processor and 1 GB of RAM. We performed simulations of straight, wavy, and curly hair undergoing various motions. In producing the accompanying animations (Animations #1∼#4 in Figure 5), we used a fixed simulation time step of 10 milliseconds. Final images were rendered using the method described in [CK05a]. Table 2 summarizes the performance statistics collected while producing the animations.

Animation	#1	#2	#3	#4
#(wisps)	142	142	274	274
#(segments)	2,687	3,136	3,583	5,672
#(triangles)	8,257	8,057	7,816	15,801
sec./frame	1.57	2.01	4.00	5.13
(wisp-body)	11.1%	13.5%	12.92%	7.97%
(wisp-wisp)	52.9%	58.9%	63.44%	70.58%
(solving)	21.5%	18.9%	14.55%	12.75%
(etc.)	14.5%	8.7%	9.09%	8.70%

Table 2: *Performance statistics: #(segments) = the total number of segments, #(triangles) = the number of triangles of the body; the lower part summarizes the amount of computation taken for each sub-task: wisp-body and wisp-wisp collision handling, solving equations, and other tasks such as updating the states.*

Animation #1 demonstrates that the technique produces natural movements of long wavy hair in a walking motion. To generate smooth interference of wisps, we set $\lambda = 0.5$ in Equation (10) and $k_{vis} = 2.5$ in Equation (11). Animation #2 is the result of simulating curly hair during a dancing motion. Because curly hair looks more natural when high curvature wisps are less intrusive, we set $\lambda = 0.1$ and $k_{vis} = 0.5$ for this simulation. For both animations, we used $g = 0.5$ in Equation (10). Animation #3 demonstrates that our hair simulation technique can robustly handle the collisions even when all of the wisps undergo a sudden and large configuration change. Animation #4 shows the result of a simulation in which the front part of the hair is fixed by a headband. The effects of accessories can be easily incorporated into the proposed simulation framework; all that is required is that additional constraints be imposed to fix the position and orientation of some wisp segments with respect to the head.

6. Conclusions

We have presented a new framework for simulating dynamic movements of human hair. Our framework, which treats the hair system as a collection of wisps, is a physically based technique that can simulate free motion of wisps as well as their constrained movements when they collide with the body while interacting with other wisps. By imposing soft constraints to a rigid multi-body serial chain, we have developed a new dynamic simulation system that can be formulated with an implicit integration method. The system is stable and fast while solving the dynamics of complex hair by accounting for the bending and torsional components. The implicit integration also enables wisp-body collision/contact constraints to be incorporated into the system equations so that the collisions can be handled systematically and robustly. Finally, by introducing an impulse-based collision force calculation as well as a viscous damping and cohesive force, the hair-hair interactions in the proposed framework are more stable than in previous methods.

In the proposed method, wisp-wisp collision handling is not as robust as wisp-body collision handling because the dynamics system is solved separately for each wisp. Another limitation of the proposed technique is that it may not properly simulate the movements of hairstyles in which wisp boundaries are not clear or strands shift between wisps over time. This drawback (inherent in wisp-based simulators) could potentially be rectified by employing the concept of the adaptive wisp tree [BKCN03] or level-of-detail representations [WLL*03]. Finally, we are actively searching for an algorithm to more efficiently handle wisp-wisp collision detection, which accounts for more than half of the simulation time in the current method (see Table 2). If significant computational savings could be made in this algorithm, it could potentially make possible the realtime simulation of hair [BCN03, BMC05, KH01, VMT04].

Acknowledgments

The authors wish to appreciate the valuable discussions with Dr. Kwang-Jin Choi during the development of this work. We also would like to thank Jae-Wang Lee for setting up the modeling and animation data for the experiments. This research was supported by the Ministry of Information and Communication, Republic of Korea. This work was also partially supported by the Automation and Systems Research Institute at Seoul National University and the Brain Korea 21 Project.

References

[AUK92] Anjyo K., Usami Y., Kurihara T.: A simple method for extracting the natural beauty of hair. In *Proceedings of SIGGRAPH 92* (July 1992), pp. 111–120.

[Bar96] Baraff D.: Linear-time dynamics using lagrange multipliers. In *Proceedings of SIGGRAPH 96* (Aug. 1996), pp. 137–146.

[BCN03] Bando Y., Chen B.-Y., Nishita T.: Animating hair with loosely connected particles. *Computer Graphics Forum (Eurographics 2003) 22*, 3 (2003), 411–418.

[BFA02] Bridson R., Fedkiw R. P., Anderson J.: Robust treatment of collisions, contact, and friction for cloth animation. *ACM Transactions on Graphics (Siggraph 2002) 21*, 3 (July 2002), 594–603.

[BKCN03] Bertails F., Kim T., Cani M.-P., Neumann U.: Adaptive wisp tree—a multiresolution control structure for simulating dynamic clustering in hair motion. In *ACM SIGGRAPH Symposium on Computer Animation* (2003), pp. 207–213.

[BMC05] Bertails F., Ménier C., Cani M.-P.: A practical self-shadowing algorithm for interactive hair animation. In *Graphics Interface* (May 2005).

[BW98] Baraff D., Witkin A. P.: Large steps in cloth simulation. In *Proceedings of SIGGRAPH 98* (July 1998), pp. 43–54.

[BWK03] Baraff D., Witkin A., Kass M.: Untangling cloth. *ACM Transactions on Graphics (Siggraph 2003) 22*, 3 (July 2003), 862–870.

[CJY02] Chang J. T., Jin J., Yu Y.: A practical model for hair mutual interactions. In *ACM SIGGRAPH Symposium on Computer Animation* (July 2002), pp. 73–80.

[CK05a] Choe B., Ko H.-S.: A statistical wisp model and pseudophysical approaches for interactive hairstyle generation. *IEEE Transactions on Visualization and Computer Graphics 11*, 2 (March/April 2005), 160–170.

[CK05b] Choi M. G., Ko H.-S.: *Dynamic Simulation of Thin Rods using Linear and Angular Springs*. Tech. Rep. SNU-EE-TR-2005-1, Seoul National University, June 2005.

[DTKT93] Daldegan A., Thalmann N. M., Kurihara T., Thalmann D.: An integrated system for modeling, animating and rendering hair. *Computer Graphics Forum (Eurographics '93) 12*, 3 (1993), 211–221.

[Fea87] Featherstone R.: *Robot Dynamics Algorithms*. Kluwer Academic Publishers, 1987.

[GBF03] Guendelman E., Bridson R., Fedkiw R. P.: Nonconvex rigid bodies with stacking. *ACM Transactions on Graphics (Siggraph 2003) 22*, 3 (July 2003), 871–878.

[Had03] Hadap S.: *Hair Simulation*. PhD thesis, MIRALab, CUI, University of Geneva, 2003.

[HMT01] Hadap S., Magnenat-Thalmann N.: Modeling dynamic hair as a continuum. *Computer Graphics Forum (Eurographics 2001) 20*, 3 (2001), 329–338.

[KH01] Koh C. K., Huang Z.: A simple physics model to animate human hair modeled in 2D strips in real time. In *Proceedings of the Eurographics Workshop on Computer Animation and Simulation* (2001), pp. 127–138.

[KN00] Kim T.-Y., Neumann U.: A thin shell volume for modeling human hair. In *Computer Animation 2000* (May 2000), pp. 104–111.

[KN02] Kim T.-Y., Neumann U.: Interactive multiresolution hair modeling and editing. *ACM Transactions on Graphics (Siggraph 2002) 21*, 3 (July 2002), 620–629.

[LK01] Lee D., Ko H.-S.: Natural hairstyle modeling and animation. *Graphical Models 63*, 2 (Mar. 2001), 67–85.

[PCP02] Plante E., Cani M., Poulin P.: Capturing the complexity of hair motion. *Graphical Models 64*, 1 (Jan. 2002), 40–58.

[RCT91] Rosenblum R. E., Carlson W. E., Tripp E.: Simulating the structure and dynamics of human hair: modeling, rendering and animation. *The Journal of Visualization and Computer Animation 2* (1991), 141–148.

[VMT04] Volino P., Magnenat-Thalmann N.: Animating complex hairstyles in real–time. In *ACM Symposium on Virtual Reality Sofrware and Technology* (Nov. 2004).

[WGL04] Ward K., Galoppo N., Lin M. C.: Modeling hair influenced by water and styling products. In *Proc. of Computer Animation and Social Agents* (2004).

[WL03] Ward K., Lin M. C.: Adaptive grouping and subdivision for simulating hair dynamics. In *Pacific Conference on Computer Graphics and Applications* (2003).

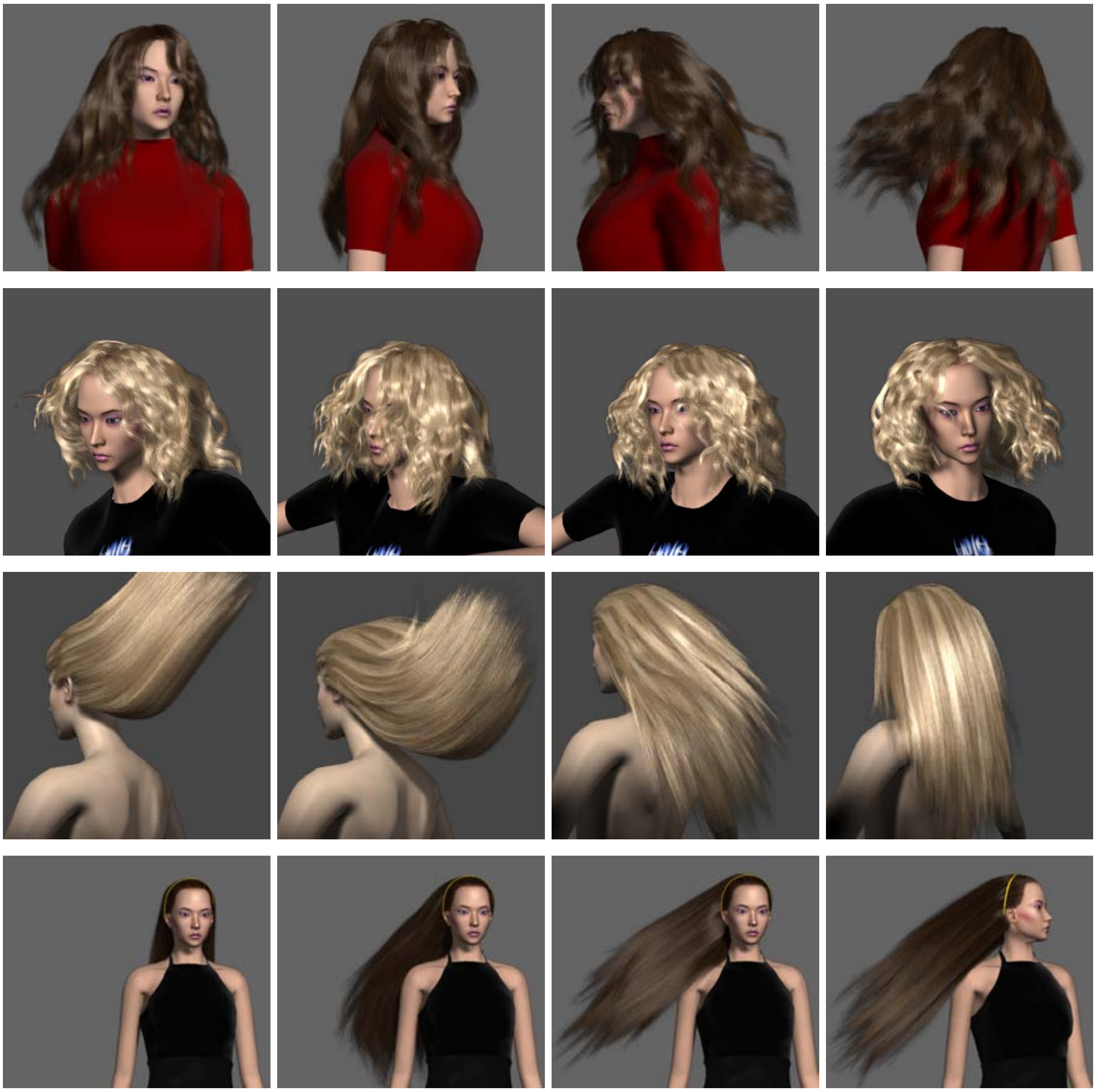


Figure 5: *Snapshots from Animations #1∼#4.*

[WLL*03] Ward K., Lin M. C., Lee J., Fisher S., Macri D.: Modeling hair using level-of-detail representations. In *Proc. of Computer Animation and Social Agents* (2003).

[WS92] Watanabe Y., Suenaga Y.: A trigonal prism-based method for hair image generation. *IEEE Computer Graphics & Applications 12*, 1 (Jan. 1992), 47–53.

[XY01] Xu Z., Yang X. D.: V-hairstudio: an interactive tool for hair design. *IEEE Computer Graphics & Applications 21*, 3 (2001), 36–42.

Appendix A: Computation of Jacobian Matrices

The implicit scheme described in Section 2.3 requires Jacobian matrices. The Jacobian matrices $\partial \mathbf{F}_i/\partial \mathbf{x}_j$ and $\partial \tau_i/\partial \mathbf{x}_j$ can be obtained analytically in a straightforward manner. For the Jacobian matrices $\partial \mathbf{F}_i/\partial \mathbf{u}_j$ and $\partial \tau_i^L/\partial \mathbf{u}_j$, we need to compute only $\partial(\mathbf{R}_i\mathbf{r}_i)/\partial \mathbf{u}_i$ (all other terms are zero because the local change of the j-th orientation has no effect on the i-th orientation when $i \neq j$). Its analytic form is $\partial(\mathbf{R}_i\mathbf{r}_i)/\partial \mathbf{u}_i = \mathbf{R}_i\partial(\mathscr{R}(\mathbf{u}_i)\mathbf{r}_i)/\partial \mathbf{u}_i = -\mathbf{R}_i\mathbf{r}_i\times$, where $\mathscr{R}(\cdot)$ is Rodrigues' formula that converts a rotation vector into its matrix form and $\mathbf{r}_i\times$ denotes the standard skew symmetric matrix of vector $\mathbf{r}_i$. However, it is difficult to analytically obtain the Jacobian of the torque given in Equation (2) with respect to $\mathbf{u}_j$. Thus, we calculate $\partial \tau_i^A/\partial \mathbf{u}_j$ numerically.

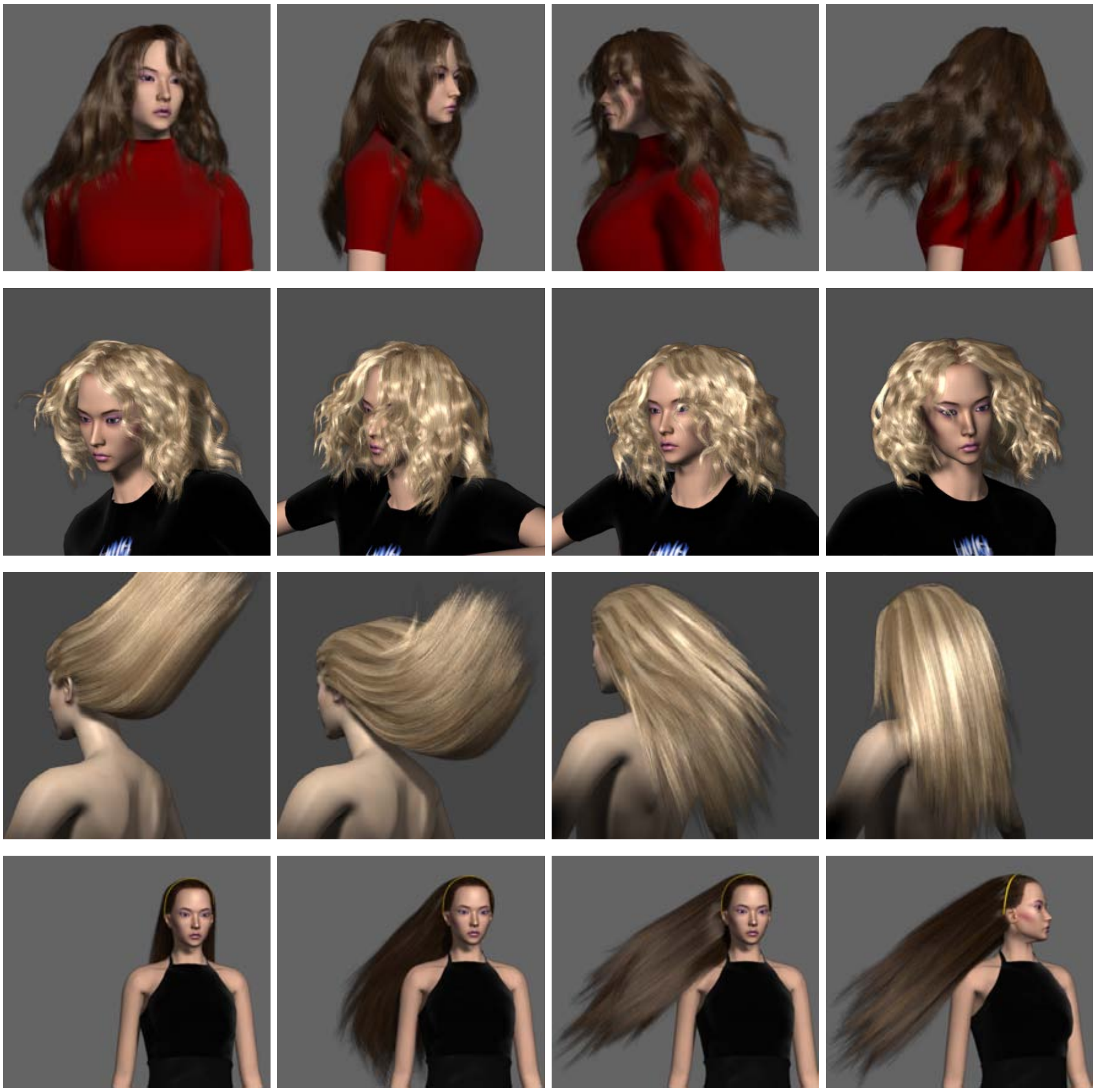


Figure 5: *Snapshots from Animations #1∼#4.*