A Practical Model for Hair Mutual Interactions

Johnny T. Chang† **Jingyi Jin**† **Yizhou Yu**†

Department of Computer Science
University of Illinois at Urbana-Champaign

Abstract

Hair exhibits strong anisotropic dynamic properties which demand distinct dynamic models for single strands and hair-hair interactions. While a single strand can be modeled as a multibody open chain expressed in generalized coordinates, modeling hair-hair interactions is a more difficult problem. A dynamic model for this purpose is proposed based on a sparse set of guide strands. Long range connections among the strands are modeled as breakable static links formulated as nonreversible positional springs. Dynamic hair-to-hair collision is solved with the help of auxiliary triangle strips among nearby strands. Adaptive guide strands can be generated and removed on the fly to dynamically control the accuracy of a simulation. A high-quality dense hair model can be obtained at the end by transforming and interpolating the sparse guide strands. Fine imagery of the final dense model is rendered by considering both primary scattering and self-shadowing inside the hair volume which is modeled as being partially translucent.

Keywords: Hair Animation, Hair-Hair Interaction, Static Links, Collision Detection, Open Chain, Hair Rendering

1 Introduction

Hair is a crucial element of appearance. One of the many challenges in simulating believable virtual humans and animals has been to produce realistic looking hair. Creating realistic hair presents problems in all aspects of computer graphics technologies, i.e. shape modeling, dynamics and rendering [11; 22; 16; 26; 1; 4; 8; 28; 13; 9; 18; 10; 7; 17; 15; 21; 29]. Hair rendering and shape modeling of fur like short hair is becoming increasingly available to animators. However, shape modeling and dynamics of long hair has been difficult. The difficulties arise from the number of hair strands, their geometric intricacies and associated complex physical interactions such as collisions, shadowing and static charges. These interactions contribute to both static and dynamic appearances of hair. There has not been much work proposing solutions modeling these interactions.

In this paper, we focus on the dynamics of long hair, and hair mutual interactions in particular. Hair has highly anisotropic dynamic properties, i.e. hair strands are extremely hard to stretch but free to move laterally and interact with each other irregularly. Strands cannot penetrate each other when they intersect; yet, each strand does not have a fixed set of neighboring strands. These unique properties inform us that a custom designed dynamic model is necessary to achieve realistic results.

The dynamics of long hair involve three aspects. First, an individual hair strand can deform and interact with the scalp, cloth and other objects. Second, an initial hairstyle can usually be recovered after subsequent head movement and the application of external force fields. This means a hairstyle can memorize its original configuration. Slight movement does not erase this memory. However, radical movement may permanently damage this memory and no complete recovery is possible. Third, there are dynamic collisions among different strands. A real person can have as many as 100,000 hairs. Each hair can be modeled as dozens of hair segments. Directly detecting pairwise collisions among hair segments is neither necessary nor computationally practical. Hair usually forms clusters and layers. Because of static charges and other forces, hairs in the same cluster or layer stick to each other. Therefore, we should model hair collisions at a higher abstraction level.

In this paper, we design an integrated sparse model for hair dynamics considering the aspects mentioned above. Specifically, this model has the following features: i) an initial hair connection model that allows hairstyle recovery after minor movement, ii) a hair mutual collision model that considers the hair volume as a collection of continuous strips, iii) an adaptive hair generation scheme to complement our sparse hair model. Since we adopt a dynamic hair model consisting of layers and clusters, solving physical interactions among them is computationally efficient without losing much of the quality from a dense model.

1.1 Related Work

We limit the overview to the previous work on hair dynamics, focusing on explicit hair models. In these models, each hair strand is considered for shape and dynamics. They are more realistic and especially suitable for dynamics of long hair. Rosenblum *et al* [22] and Daldegan *et al* [4] used a mass-spring-hinge model to control the position and orientation of hair strands. Anjyo *et al* [1] modeled hair with a simplified cantilever beam and used one-dimensional projective differential equation of angular momentum to animate hair strand. Daldegan *et al* used sparse characteristic hairs to reduce computations. None of these previous attempts considered hair-hair interactions and hairstyle recovery after minor movement. Individual hair dynamics was approximated using simplified models.

Recently, Hadap and Magnenat-Thalmann [10] proposed a novel approach to model dense dynamic hair as continuum by using a fluid model for lateral hair movement. Hair-hair collision is approximated by the pressure term in fluid mechanics while friction is approximated by viscosity. Single strand dynamics is solved using the formulation of a multibody open chain. Hair-air interaction is considered by integrating hairs with an additional fluid system for

†Email: {jtchang, jingjin, yyz}@uiuc.edu
Website: www-faculty.cs.uiuc.edu/∼yyz/research/hair/

the air. This work presented a promising and elegant model for hair interactions. Nonetheless, it has some limitations; the gradient of the pressure may generate a collision force along a direction incompatible with the velocities of the colliding hairs because pressure is defined as a function of the local density which has no knowledge of the velocities.

Plante *et al* [21] proposed a wisps model for simulating interactions inside long hair. Hair strands are clustered into wisps and modeled as anisotropic viscous volumes. Each wisp volume consists of a skeleton and a deformable envelope. The skeleton captures the global motion of a wisp, while the envelope models the local radial deformation. However, There is a lack of coherence in motion among nearby wisps. Koh and Huang presented an approach by explicitly modeling hair as a set of 2D strips [15]. Collisions between hair strips are handled to create more realistic motion. One drawback with the 2D strip-based approach is that the volumetric aspect of the hair is not captured. Other researchers also tried to model and constrain hair using a single thin shell or multiple head hull layers [13; 17].

Recently, there has been a few feature films, such as Final Fantasy and Monsters Incorporated, with realistic hair simulations as well as some commercial software packages, such as Shave[24] and Shag[23], for hair simulations. Shave is considered as the best commercial hair modeling and simulation software in the industry. However, its single strand dynamics does not look realistic, and it does not have hair-hair collision. Final Fantasy is the film with the best simulations for long human hair. From the press releases, Aki's hair was modeled as a whole deformable exterior surface and some of the simulations were done using the Maya cloth plugin. That means hairs are constrained around the surface to enable very good hairstyle recovery, but much of the lateral freedom has been lost. In many situations, the hair flows like a piece of cloth instead of a set of individual stiff strands. On the other hand, Monsters Incorporated has long fur simulation [7]. Each hair is considered as particles linked in a chain by a set of stiff springs. A builder or a small snippet of code is used to generate the inbetween hairs. Hair-hair collision has not been considered.

2 Overview

Although this paper focuses on hair-hair interaction, modeling, simulation and rendering are three inseparable stages for the production of fine hair imagery. The input to our simulation algorithm is an initial sparse hair model with a few hundred strands generated from a previous hair modeling method [29]. Each strand from the sparse model has multiple segments connected by vertices. Each strand serves as the *guide hair* for a whole cluster and may have its distinct curly features. The sparse model is then equipped with structural elements needed for dynamic simulation. For example, each vertex is considered as a rotational joint with a hinge. Connections and triangular meshes among guide hairs are then built for simulating hair-hair interactions. Such an enhanced model is then ready for dynamic simulation. Note that these enhanced structures are "invisible", which means they are never visualized during hair rendering although the effects they produce are incorporated into hair motion. Once an animation sequence of the sparse model is generated, additional hairs are interpolated to produce a dense model for final rendering. In the rendering stage, we consider both diffuse and specular reflection as well as partial translucency of each strand by integrating volume density rendering with a modified version of the opacity shadow buffer algorithm [14].

3 Single Hair Strand Dynamics

There are few techniques developed on modeling single hair strand dynamics [22; 1; 4; 10]. Some of the previous work [22; 4] models a single strand as particles connected with rigid springs. A hair strand is approximated by a set of particles. Each particle has 3 degrees of freedom, namely one translation and two angular rotations. This method is simple and easy to implement. However, individual hair strand has very large tensile strength, and hardly stretches by its own weight and body forces. This property leads to stiff equations which tend to cause numerical instability unless very small time steps are used. We model each hair strand as a serial rigid multibody chain. There is a rotational joint between two adjacent segments, and translational motion is prohibited. A single chain can be considered as a simple articulated body with joint constraints. Dynamic formulations of articulated bodies are addressed in robotics [5; 20] as well as graphics literature [27]. Both constrained dynamics with Lagrange Multipliers [2] and generalized(or reduced) coordinate formulation [5] can be used equally efficiently. The dynamics of a serial multibody chain and its generalized coordinate formulation have recently been applied to single hair simulation by Hadap and Magnenat-Thalmann [10]. The main focus of our paper is on hair-hair interaction; therefore, we describe the formulation of the serial multibody chain and our adaptations briefly in this section.

3.1 Kinematic Equations

In our model, we assume that the twisting of a hair strand along its axis is prohibited. This reduces each rotational joint in a strand to have two degrees of freedom. A rotational joint can be decomposed into two cascading one-dimensional revolute joints each of which has a fixed rotation axis. The rotation angles at the 1D revolute joints represent the set of generalized coordinates in a multibody chain system. If a 1D revolute joint has a rotation axis ω along with a point q on the axis, the matrix transformation corresponding to a rotation around ω by an angle θ can be given by the exponential map $e^{\widehat{\xi}\theta}$ [20] where

$$\widehat{\xi} = \begin{bmatrix} \widehat{\omega} & v \\ 0 & 0 \end{bmatrix}, \widehat{\omega} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix},$$

and $v = -\omega \times q$. Suppose a hair segment has n preceding 1D revolute joints in the chain and a local frame is defined at the segment. Assume the local-to-world transformation for this frame when all preceding joint angles are zero is $g_{st}(0)$. The updated local-to-world transformation after a series of rotations at the n joints becomes

$$g_{st}(\Theta) = e^{\widehat{\xi_1}\theta_1} e^{\widehat{\xi_2}\theta_2} \cdots e^{\widehat{\xi_n}\theta_n} g_{st}(0) \quad (1)$$

Thus, given an arbitrary series of joint angles, the position of every vertex in the chain can be obtained using this product of exponentials of its preceding joints. The exponential map actually is just another way of formulating a 4×4 homogeneous matrix. It can be calculated in constant time [20]. Therefore, the whole chain can be evaluated in linear time.

3.2 Dynamics of Hair Strand

Given the mapping in Eq. 1 which is from the set of generalized coordinates (joint angles) to real 3D world coordinates, hair strand simulation can be solved by integrating joint angular velocities and accelerations. Forward dynamics of a single strand in terms of joint angular velocities and accelerations can be solved using the Articulated-Body Method [5] or Lagrange's equations for generalized coordinates [20]. The former method is more efficient with a linear time complexity.

Both external and internal forces are indispensable for single hair dynamics. In this paper, hair-hair interactions are formulated as external forces in addition to gravity. The actual form of these external forces will be discussed in Section 4. At each joint of the hair chain, there is also an internal actuator force to account for the bending and torsional rigidity of the strand. We model the actuator force as a hinge with a damping term as in [22]. Since our hair model may have curly hair strands which means the strands are not straight even without any external forces, we define a nonzero resting position for each hinge. Any deviation from the resting position results in a nonzero actuator force trying to reduce the amount of deviation. This setup enables a strand to recover its original shape after subsequent movement.

3.3 Strand-Body Collision

In order to simulate inelastic collision between the hair and human body, there is no repelling forces introduced by the human body. Once a hair vertex becomes close enough to the scalp or torso, it is simply stopped by setting its own velocity to be the same as the velocity of the human body while all the following vertices in the multibody chain are still allowed to move freely. Any acceleration towards the human body is also prohibited at the stopped vertices which, however, are allowed to move away from or slide over the human body. Frictional forces are added as well to those vertices touching the human body. Collision detection is handled explicitly by checking penetration of hair strand particles with the triangle mesh of the body parts.

This scheme cannot guarantee that the hair vertices do not penetrate other colliding surfaces in the middle of a time step. If penetration does occur, we need to move the part of the penetrating strand outside the surface in the same time step so that no penetration can be actually observed. It is desirable that the tip of the hair, if outside the surface, remains unchanged during this adjustment in order to introduce minimal visual artifacts. To achieve this goal, inverse kinematics [20] can be applied to adjust the positions of the intermediate vertices between the tip and the adjusted locations of the penetrating vertices. In our implementation we opt for a simpler method using iterative local displacements. Starting from the root, we move the first penetrated vertex p_1 to its nearest valid location p_1', and then propagate this displacement by moving the subsequent vertices. More specifically, assume the following vertex of p_1 is p_2, we compute the vector $v' = ||p_2 - p_1||\frac{v}{|v|}$, where $v = p_2 - p_1'$. The new location for p_2 after the adjustment is $p_2' = p_1' + v'$. We repeat this for all the vertices following p_1 until reaching the tip.

4 A Sparse Model for Hair-Hair Interaction

We devise a novel scheme to simulate only a sparse set of hair strands for complex hair-hair interactions. We first introduce an elastic model to preserve the relative positions of the hair strands. The static links model the interaction of the hair due to interweaving, static charges and hairstyling. Second, the hair-hair collision and friction is simulated using the guide hairs and a collection of auxiliary triangle strips. Third, an interpolation procedure is described for generating dense hair from our sparse hair model. Last, we provide an adaptive hair generation technique to complement our sparse hair model; additional guide strands are added on the fly to reduce the interpolation artifacts. The proposed method models the hair dynamics efficiently with good visual realism.

4.1 Static Links

It is evident that the hair strands tend to bond together with other strands in their vicinity because of cosmetics, static charges and the interweaving of curly hairs. As a result, the movement of each strand is on most part depended on the motion of other strands. These interactions can have relatively long range effects besides clustering in a small neighborhood. While hair local clustering is modeled by default using our sparse model, longer range interaction is not. Furthermore, slight head movements or external forces exerted on the hair do not change a hairstyle radically. This is partly because each hair strand has its internal joint forces and resting configuration. However, an individual hair's recovery capability is quite limited especially for long hairs. The bonding effect among hairs plays an important role. Dramatic movements can break the bonds created by hairstyling, static charges or interweaving.

To effectively model the bonding effect, we may view the hair as one elastically deformable volume. Traditional models for deformable bodies include 3D mass-spring lattice, finite difference, and finite element method [25; 30]. These models approximate the deviation of a continuum body from its resting shape in terms of displacements at a finite number of points called nodal points. Although the vertices of hair strands may serve as the nodal points inside this hair volume, directly applying traditional models is not appropriate for the following reasons. We are only interested in an elastic model for hair's lateral motion. Under strong external forces, the continuum hair volume may break into pieces which may have global transformations among them. Therefore, using one body coordinate system for the whole hair volume is inadequate.

We propose to build breakable connections, called static links, among hair strands to simulate their elastic lateral motion and enable hairstyle recovery. These connections are selected initially to represent bonds specific to a hairstyle since different hairstyles have different hair adjacency configuration. The static links enforce these adjacency constraints by exerting external forces onto the hair strands. Intuitively, one can use tensile, bending and torsional springs as bonds to preserve the relative positions of the hair strands. In practice, we opt for a simpler and more efficient method using local coordinates.

We introduce a local coordinate system to each segment of the hair strands. For each segment, we find a number of closest points on nearby strands as its reference points. To improve the performance, an octree can be used to store the hair segments for faster searching. We transform these points, which are in the world coordinates, to the segment's local coordinates (Fig. 1a). The initial local coordinates of these reference points are stored as part of the initialization process. Once strands have relative motion, the local coordinates of the reference points change and external forces are exerted onto these strands to recover their original relative positions (Fig. 1b). We model these external forces as spring forces with zero resting length. One advantage of using the local coordinates is that it eliminates the need for bending and torsional springs.

Let us consider a single hair segment h with m reference points. The initial local coordinates of these reference points are represented as $p^o_{h,i}, i = 1, ..., m$, while their new local coordinates are represented as $p^n_{h,i}, i = 1, ..., m$. The accumulated force this segment receives due to static links can be formulated as

$$\mathbf{f}_h = \sum_i \left[k^s_{h,i}|\mathbf{l}_i| - k^d \frac{\mathbf{v}_i \cdot \mathbf{l}_i}{|\mathbf{l}_i|} \right] \frac{\mathbf{l}_i}{|\mathbf{l}_i|} \tag{2}$$

We compute the spring force using the Hook's law in (2), where $k^s_{h,i}$ is the spring constant for the i-th reference point of segment h, and k^d is the universal damping constant. Since the resting length in our case is zero, $|\mathbf{l}_i = p^n_{h,i} - p^o_{h,i}|$ is multiplied by $k^s_{h,i}$ directly. $\mathbf{v}_i$ is the time derivative of $\mathbf{l}_i$.

Similar to the bonds of stylized hair, static links can be broken upon excessive forces. We set a threshold for each static link. If the length change of a static link is greater than the threshold, the static link breaks (Fig. 1c). Once a link is broken, the damage is permanent; the link will remain broken until the end of the simulation. To be more precise, we model the spring constant $k^s_{h,i}$ as shown in Fig. 2. As $|\mathbf{l}_i|$ increases beyond δ_1, the spring constant begins to decrease gradually and eventually becomes zero at δ_2 as the spring snaps. The spring constant will not recover even when $|\mathbf{l}_i|$ shrinks below δ_1 again. This nonreversible spring model would make the motion of the hair look less like a collection of rigid springs.

When external forces recede, the original hairstyle may not be recovered if some of the static links have been broken. New static

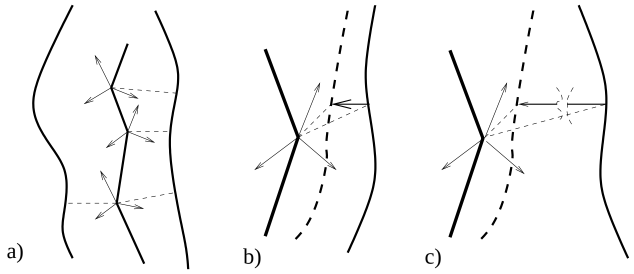

Figure 1: Each hair segment has its own local coordinate system where the forces from all static links (dashed lines) are calculated.

links may form for the new hairstyle with updated neighborhood structures.

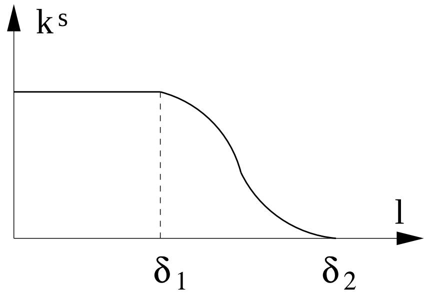

Figure 2: Spring constant $k_{h,i}^s$ vs displacement graph.

4.2 Dynamic Interactions

Elastic deformation only introduces one type of hair-hair interactions. Hairs also interact with each other in the form of collision. To effectively simulate hair-hair collision and friction using a sparse hair model, we need to have a dynamic model that imagines the space in between the set of sparse hairs as being filled with dense hairs. Collision detection among the guide hairs only is much less accurate. Let us consider a pair of nearby guide hairs. The space between them may be filled with some hairs in a dense model so another strand cannot pass through there without receiving any resistance. To model this effect, we can either consider the guide hairs as two generalized cylinders with large enough radii to fill up the gap between them, or build an auxiliary triangle strip as a layer of dense hair between them by connecting corresponding vertices. The triangular mesh can automatically resize as the guide hairs move, but it is trickier to resize the generalized cylinders. Therefore, we propose to construct auxiliary triangle strips between pairs of guide hairs to approximate a dense hair distribution. If we consider the set of dense hairs collectively as a volume, a triangle strip represents a narrow cross section of the volume. A number of such cross sections can reasonably approximate the density distribution of the original hair volume.

Since the distance between a pair of vertices from two hairs may change all the time during simulation, we decide to use the distance among hair roots. A triangle strip is allowed as long as two guide hairs have nearby hair roots. Each triangle only connects vertices from two guide hairs, therefore is almost parallel to them. Note that the triangle strips may intersect with each other. This does not complicate things because each triangle is treated as an independent patch of hair during collision detection. The triangles are only used for helping collision detection, not considered as part of the real hair geometry during final rendering. They do not have any other dynamic elements to influence hair movement. However, some triangles may have nearby static links which can help them resist deformation. The triangle edges are not directly constructed as static links because static links only connect nearby hair segments while not all the segments connected by triangles are close to each other.

As in standard surface collision detection, two different kinds of collision are considered, namely, the collision between two hair segments and the collision between a hair vertex and a triangular face. Since each guide hair represents a local hair cluster with a certain thickness, a collision is detected as long as the distance between two hair elements falls below a nonzero threshold. Once a collision is detected, a strongly damped spring force is dynamically generated to push the pair of elements away from each other [3]. Meanwhile, a frictional force is also generated to resist tangential motion. A triangle redistributes the forces it receives to its vertices as their additional external forces. Both the spring and frictional forces disappear when the distance between the two colliding elements becomes larger than the threshold. The spring force in effect keeps other hairs from penetrating a layer corresponding to a triangle strip. An octree is used for fast collision detection. All the moving hair segments and triangles are dynamically deposited into the octree at each time step. An octree node has a list of segments and triangles it intersects with.

Hair also exhibits strong anisotropic dynamical properties. Depending on the orientation of the penetrating hair vertex and the triangular face, the repelling spring force might vary. For example, hair segments of similar orientation with the triangle strip should experience weaker forces. We scale the repelling spring force according to the following formula.

$$\mathbf{f}_r = \lambda(1 - |\mathbf{a} \cdot \mathbf{b}|)\mathbf{f}_s \tag{3}$$

The original spring force $\mathbf{f}_s$ is scaled in Eq. (3), where $\mathbf{a}$ is the normalized tangential vector of the hair at the penetrating vertex, $\mathbf{b}$ is the interpolated hair orientation on the triangular face from its guide hair segments, and λ is a scale factor. When $\mathbf{a}$ and $\mathbf{b}$ are perfectly aligned, the scaled force $\mathbf{f}_s$ becomes zero. On the contrary, when they are perpendicular, the spring force is maximized. The collision force between two hair segments can be defined likewise.

The hair density on each hair strip is also modeled as a continuum. It can be dynamically adjusted during a simulation. If there is insufficient hair on a strip, the strip can be broken. This would allow other hair strands to go through broken pieces of a hair strip more easily. This is reasonable because sometimes there is no hair between two hair clusters while at the other times, there may be a dense hair distribution. In our current implementation, the length of the triangle edges serves as the indicator for when the hair density on a strip should be adjusted. If a triangle becomes too elongated, it is labeled as broken. If a triangle is not broken, the magnitude of the collision force in Eq. (3) is made adaptive by adjusting the scale factor λ according to the local width of the triangle strip to account for the change of hair density on the triangle. Unlike the static links, this process is reversible. Once the two guide hairs of a strip move closer to each other again, indicating the hair density between them is increasing, the generated collision force should also be increased, and the triangle strip should be recovered if it has been broken. If every triangle strip in our method is modeled as broken from the beginning, our collision model becomes similar to the wisp model in [21] since every guide hair in our method actually represents a wisp.

It may not be necessary to build triangle strips among all pairs of nearby strands. For a simple brush in Fig. 3a, we can only insert triangle strips between horizontally and vertically adjacent guide hairs. For human hair, we sometimes find it practically good enough to build triangle strips between guide hairs with horizontally adjacent hair roots (Fig. 3b). This is because hairs drape down due to gravity, and the thickness of the hair volume is usually much

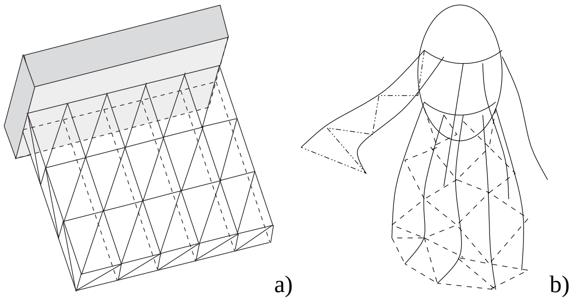

Figure 3: a) For a brush, triangle strips can be inserted between horizontally and vertically adjacent guide hairs. b) For a human scalp, triangle strips are inserted only between horizontally adjacent guide hairs

smaller than the dimensions of the exterior surface of the hair volume. In such a situation, using triangles to fill the horizontal gaps among guide hairs becomes more important.

4.3 Adaptive Hair Generation

Initially, we select the guide hairs uniformly on the scalp. However, it is not always ideal to pick the guide hairs uniformly. During a run of the simulation, some part of the hair may be more active than the other parts. For example, when the wind is blowing on one side of the hair, the other side of the hair appears to be less active. As a result, some computation is wasted for not so active regions. For not so active regions, fewer guide hairs combined with interpolation (Section 5) is sufficient. However, for more active regions, it is desirable to use more guide hairs and less interpolation for better results. We design an adaptive hair generation method to complement our sparse hair model.

We generate additional guide strands adaptively during the simulation to cover the over interpolated regions. The distribution and the initial number of guide strands are determined before the simulation. However, as the simulation proceeds, more hair strands can be added. The hair model may become more and more computationally intensive if hairs can only be inserted. We notice that the inserted hairs may become inactive again later in the same simulation. Therefore, we also allow them to be deleted if necessary. To keep our hair strands relatively sparse, we may also set a limit on how many adaptive hair strands can exist at the same time. Picking the right place to generate adaptive guide hair is important.

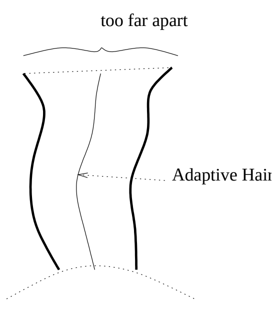

Figure 4: Adaptive hair generation

We use a simple technique to detect where to add and remove adaptive guide hairs. For each pair of guide strands, we compute the distance between all pairs of corresponding vertices of the strands. If any pair of vertices become farther away than a threshold, it indicates that the hair in between these two guide strands is relying too much on the interpolation. We then add an adaptive guide hair half-way between these two strands (Fig. 4). At the same time, we examine the adaptive guide hairs from the last step of the simulation. If some of the guide strands are no longer needed (when the two neighboring strands are close enough), we remove those strands and save them for future hair generation. When an adaptive hair is generated, its initial vertex positions and velocities are obtained by interpolating from those of the two initiating guide hairs. If there was a triangle strip between these two initiating hairs, it should be updated to two strips with the new hair in the middle. The new adaptive hair then follows its own dynamics from the next time step, colliding with nearby strands and triangle strips. To avoid discontinuous motion on the rest of the hairs, a new adaptive hair does not spawn static links with other strands.

5 Hair Interpolation

5.1 Interpolating a dense set of hair strands

Since only a sparse set of hair strands in the order of few hundreds is simulated, a procedure must be used to interpolate the dynamics of the remaining hair strands. We designed our interpolation procedure to complement our sparse hair model to produce believable hair animation efficiently. Each hair from the sparse model serves as a guide hair. The remaining hair strands in the dense set are interpolated from the guide hairs. Intuitively, one could imagine a simple procedure by averaging the position of the neighboring strands. However, this approach tends to group strands together into unnatural clusters. We come up with a more sophisticated method that produces better interpolation results. It requires an approach for defining a local coordinate system at each potential hair root. A typical scheme for this uses a global UP vector, such as the vertical direction, and the local normal orientation. Our interpolation procedure works as follows:

- Find the nearest root of a guide hair and transform the segments of that guide hair from the world coordinates to its local coordinates. Name this transformation M_1.
- Take these segments in the local coordinates and transform them back to the world coordinate using the local-to-world coordinate transformation defined at the root of the interpolated strand. Name this transformation M_2^{-1}.

$$M_2^{-1} M_1 p \qquad (4)$$

The procedure is summarized as equation (4), where p is the location of the guide hair in the world coordinate, M_1 and M_2 are the two transformations described previously. More than one nearby guide hairs can be used together to achieve smoother results by merging the multiple transformed guide hairs with some averaging scheme. Local clustering effects can be removed by interpolation from multiple guide hairs. In summary, our procedure generates better results by taking into account the round shape of the scalp and considering both rotation and translation between local coordinate systems.

Since small objects may miss all the guide hairs, but still hit some of the strands in the dense model, we decide to run hair-object collision detection for each hair in the dense model. Although this involves a certain amount of computation, the computing power available nowadays on a single processor workstation has already become sufficient to perform this task in a very short amount of time. If a hair penetrates an object, the scheme described in Section 3.3 can be used to adjust the hair.

5.2 Hermite Spline interpolation

The smoothness of a hair strand can be improved by Hermite Spline interpolation. We observe that a relatively coarse strand model with ten to fifteen segments combined with the spline interpolation is sufficient for normal hairstyles.

6 Hair Rendering

Although the primary focus of this paper is on hair animation, we will discuss briefly our approach to rendering realistic hair. The kind of physical interaction considered here includes self-shadowing and scattering. Hair strands are not completely opaque. Therefore, the interaction between light and hair leads to both reflection and transmission. When a dense set of hairs is present, light gets bounced off or transmitted through strands multiple times to create the final exquisite appearance. Basically, we can view a dense hair as a volume density function with distinct density and structures everywhere. The hair density is related to the local light attenuation coefficient while the structures including the local hair orientation are related to the phase function during scattering. In this section, we discuss how to efficiently render animated sequences of hair with high visual quality by considering the above factors.

While secondary scattering can improve the rendering quality, primary scattering and self-shadowing are considered much more important. Since the rendering performance is our serious concern when generating hair animations, we decide to simulate the latter two effects only. This is equivalent to solving the following integral equation

$$L(x, \vec{\omega}) = \int_{x_0}^{x} \tau(x', x)\sigma(x') \sum_l f(x', \vec{\omega}_l, \vec{\omega}) I_l(x') dx' \quad (5)$$

where $L(x, \vec{\omega})$ represents the final radiance at x along direction $\vec{\omega}$, $f(x', \vec{\omega}_l, \vec{\omega})$ is the normalized phase function for scattering, $I_l(x')$ is the attenuated light intensity from the l-th light source, and $\tau(x', x) = \exp(-\int_{x'}^{x}(\alpha(\xi) + \sigma(\xi))d\xi$ where $\alpha(x)$ is the absorption coefficient and $\sigma(x)$ is the scattering coefficient. Thus, our hair rendering is similar to traditional volume rendering techniques [12]. That is, the final color of a pixel can be approximated as the alpha-blending of the colors at a few sample points along the eye ray going through that pixel. To perform alpha-blending correctly, the sample points need to be depth-sorted. In terms of hair, the sample points can be the set of intersections between the eye ray and the hair segments. Note that the input to the rendering stage is a large number of hair segments resulted from the discretization of the spline interpolated dense hairs mentioned in Section 5. In order to obtain the set of intersections at each pixel efficiently, scan conversion is applied to the segments and a segment is added into the depth-sorted list of intersections at a pixel once it passes that pixel. Antialiasing by supersampling each pixel can help produce smoother results.

To finish rendering, we still need a color for alpha-blending at each of the intersections. It should be the reflected color at the intersection. The reflectance model we use is from [8]. It is a modified version of the hair shading model in [11] by considering partial translucency of hair strands. Since other hairs between the light source and the considered hair segment can block part of the incident light, the amount of attenuation is calculated using the opacity shadow maps [14] which can be obtained more efficiently than the deep shadow maps [19]. Basically, the algorithm in [14] selects a discrete set of planar (opacity) maps perpendicular to the lighting direction. These maps are distributed uniformly across the volume being rendered. Each map contains an approximate transmittance function of the partial volume in front of the map. Thus, the approximate transmittance of the volume at any point can be obtained by interpolating the transmittance at corresponding points on the two nearest opacity maps. In our implementation, exponential interpolation has been used since the attenuation of light through a volume is exponential. The exponential interpolation can be written as

$$\exp(-\alpha_1 \frac{d_2}{d_1 + d_2} - \alpha_2 \frac{d_1}{d_1 + d_2})$$

where $\exp(-\alpha_1)$ and $\exp(-\alpha_2)$ are the attenuation at the two nearest maps, and d_1, d_2 are the distance from the point to the two maps, respectively.

When the hair is rendered together with other solid objects, such as the head and cloth, which we assume to be completely opaque, the color of the solid objects needs to be blended together with the hair's during volume rendering. The solid objects also have their separate shadow buffer for each light source. Anything in the shadow of the solids receives no light while those solids in the shadow of the hair may still receive attenuated light.

7 Results

We have successfully tested our hair dynamic model in a few animations. In our experiments, we used around 200 initial guide hairs during the animation of the sparse model with 15 segments for each strand. During each time step, a strand is interpolated with a Hermite spline and discretized into around 50 smaller segments. Based on this set of resampled sparse hairs, a dense hair model with 50,000 strands is generated on the fly at each frame for the final rendering. The guide hair animation stage takes about one second per frame on a Pentium III 800MHz processor. Hair interpolation, hair-object collision detection and antialiased rendering takes another 20 seconds per frame on a Pentium 4 2GHz processor. Fig. 5 shows a sparse hair model with static links along with the dense interpolated model. Fig. 8 shows synthetic renderings of animated hair.

7.1 Comparison with Ground Truth

A synthetic head shaking sequence is compared with a real reference sequence in Fig. 9. The hair strands in the real sequence obviously have mutual connections since they move together. We use relatively strong static links to simulate this effect. The head motion in the synthetic sequence was manually produced to approximate the real motion. Nonetheless, the synthetic hair motion reasonably matches the real one.

7.2 Dynamic Collision

To demonstrate the effectiveness of our hair collision strategy, we built a simple braided hair model and let it unfold under gravity. There are basically two sets of guide hairs in the model, and static links and triangle strips are only built among hairs from the same set. A comparison is given between images from two synthetic sequences in Fig. 6, one with collision detection and the other without. In the simulation without collision detection, hairs go through each other. But in the sequence with collision detection, hairs unfold correctly in a spiral motion.

7.3 Hair-Air Interaction

Hair-air interaction is traditionally modeled as air drag which only considers the force exerted on the hair from the air. However, the velocity field of the air is also influenced by the hair. The method in [10] can be adapted to our model for hair-air interaction. That is, the air is simulated as a fluid and it generates a velocity field. Each hair vertex receives an additional external force from the air. This force can be modeled as a damping force using the difference between the velocity of the air at the vertex and the velocity of the hair vertex itself. The force exerted from the hair back to the air can be modeled similarly. If the air is simulated using a voxel grid [6], the velocity of the hair at each grid point can be approximated using the velocities of the nearby hair vertices and auxiliary triangles.

Fig. 10(top) shows images from a hair animation with a wind. The wind velocity field is driven by an artificial force field with a changing magnitude and direction. The head and torso are considered as hard boundaries in the wind field while the wind can go through hairs with a certain amount of attenuation.

7.4 Brush Simulation

In addition to human hair interactions, we simulate the dynamics of brushes. Fig. 7 shows images from a sequence with a sphere colliding with a synthetic brush. The mutual interactions are weak when only a small number of hairs drape down behind the sphere. However, when more and more hairs drape down, they stabilize much faster because of the collisions.

7.5 Hair Rendering with An Artistic Flavor

An artistic flavor can also be added to the images by rendering the hair with increased translucency and specularity. Fig. 10(bottom) shows some re-rendered images from one of the wind blowing sequences.

8 Discussions and Conclusions

In this paper, we presented an integrated sparse model for hair dynamics. Specifically, the model can perform the following functions: the static links and the joint actuator forces enable hairstyle recovery; once the static links are broken under external forces, hairs have the freedom to move laterally; hair-hair collision becomes more accurate by inserting triangle strips and performing collision detection among strands as well as between strands and triangle strips; stable simulation of individual strands is provided by the formulation for multibody open chains. Although our model is not originally designed for hairs without obvious clustering effects, with our multiple hair interpolation scheme, visual results for this kind of hairs turned out quite reasonable.

Note that for curly hair, we have two levels of details. The sparse or interpolated hair model only has large-scale deformations without fine curly details. Each strand in these models serves as the spine of its corresponding curly strand. Curliness can be added onto the interpolated dense hair model before rendering as in [29].

Acknowledgments

This work was supported by National Science Foundation CAREER Award CCR-0132970 and start-up funds from the University of Illinois at Urbana Champaign. We would like to thank Mike Hunter and Sheila Sylvester for their help with the real hair sequence, and the anonymous reviewers for their valuable comments.

References

[1] K. Anjyo, Y. Usami, and T. Kurihara. A simple method for extracting the natural beauty of hair. In *Proc. of SIGGRAPH'92*, pages 111–120, 1992.

[2] D. Baraff. Linear-time dynamics using larange multipliers. In *Proc. of SIGGRAPH'96*, pages 137–146, 1996.

[3] D. Baraff and A. Witkin. Large steps in cloth simulation. In *Proc. of SIGGRAPH'98*, pages 43–54, 1998.

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

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

[6] R. Fedkiw, J. Stam, and H.W. Jensen. Visual simulation of smoke. In *SIGGRAPH 01 Conference Proceedings*, pages 15–22, 2001.

[7] M. Fong. Animating monster fur. In *SIGGRAPH course 36 notes*, 2001.

[8] D. Goldman. Fake fur rendering. In *Proc. of SIGGRAPH'97*, pages 127–134, 1997.

[9] S. Hadap and N. Magnenat-Thalmann. Interactive hair styler based on fluid flow. In *Computer Animation and Simulation 2000. Proceedings of the Eleventh Eurographics Workshop*, 2000.

[10] S. Hadap and N. Magnenat-Thalmann. Modeling dynamic hair as continuum. In *Eurographics Proceedings. Computer Graphics Forum, Vol.20,No.3*, 2001.

[11] J. Kajiya and T. Kay. Rendering fur with three dimensional textures. In *Proc. of SIGGRAPH'89*, pages 271–280, 1989.

[12] J.T. Kajiya and B.P. von Herzen. Ray tracing volume densities. *Computer Graphics (SIGGRAPH 84 Proceedings)*, 18(3), 1984.

[13] T.-Y. Kim and U. Neumann. A thin shell volume for modeling human hair. In *Proc. of IEEE Computer Animation*, 2000.

[14] T.-Y. Kim and U. Neumann. Opacity shadow maps. In *Proc. of Eurographics Workshop on Rendering*, pages 177–182, 2001.

[15] C. K. Koh and Z. Huang. A simple physics model to animate human hair modeled in 2d strips in real time. In *Proceedings of Eurographics Computer Animation and Simulation*, 2001.

[16] A.M. LeBlanc, R. Turner, and D. Thalmann. Rendering hair using pixel blending and shadow buffers. *Journal of Visualization and Computer Animation*, pages 92–97, 1991.

[17] D.-W. Lee and H.-S. Ko. Natural hairstyle modeling and animation. *Graphics Models and Image Processing*, 63:67–85, 2001.

[18] J. Lengyel. Real-time hair. In *Proc. of Eurographics Workshop on Rendering*, pages 243–256, 2000.

[19] T. Lokovic and E. Veach. Deep shadow maps. In *Proc. of SIGGRAPH'00*, pages 385–392, 2000.

[20] R.M. Murray, Z. Li, and S.S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[21] E. Plante, M.-P. Cani, and P. Poulin. A layered wisps model for simulating interactions inside long hair. In *Proceedings of Eurographics Computer Animation and Simulation*, 2001.

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

[23] Shag (plugin for 3d studio max). home.abac.com/ddag/hair.html.

[24] Shave (plugin for lightwave). www.joealter.com/software.html.

[25] D. Terzopoulos, J.C. Platt, and A.H. Barr. Elastically deformable models. In *Proceedings of SIGGRAPH'87*, pages 205–214, 1987.

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

[27] J. Wilhelms. Using dynamic analysis for realistic animation of articulated bodies. *IEEE CG&A*, 7(2):12–27, 1987.

[28] X.D. Yan, Z. Xu, J. Yang, and T. Wang. The cluster hair model. *Graphics Models and Image Processing*, 1999.

[29] Y. Yu. Modeling realistic virtual hairstyles. In *Proceedings of Pacific Graphics*, pages 295–304, 2001.

[30] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method: Solid and Fluid Mechanics Dynamics and Non-Linearity*. McGraw-Hill Book Company, 1989.

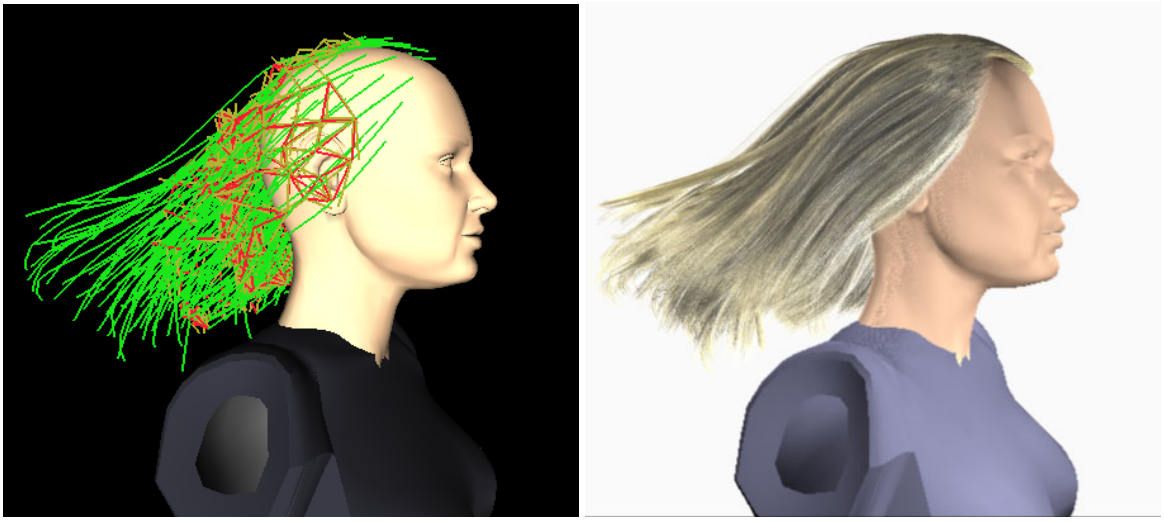

Figure 5: Left: a sparse hair model displayed with static links. Right: a rendered image of the interpolated dense model.

Figure 6: A comparison between two hair animations with and without collision detection. Top row: braided hair unfolds correctly in a spiral motion because of the collision detection. Bottom row: hairs penetrate each other when there is no collision detection.

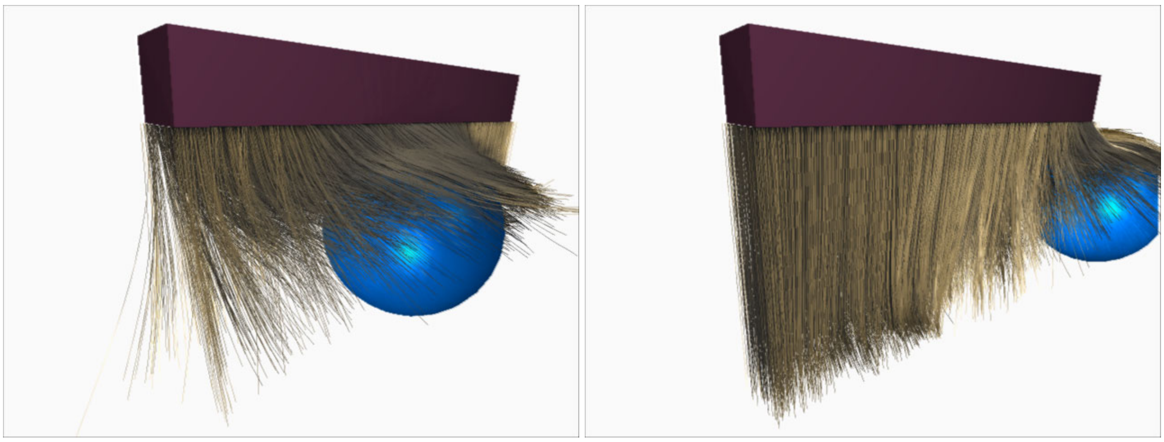

Figure 7: Two images from a sequence with a sphere colliding with a brush.

Figure 8: Two synthetic renderings of animated hair.

Figure 9: A comparison between a simulated hair animation and a real video. The hair motion is caused by the underlying head motion. Top row: images from the simulated hair motion sequence. Bottom row: images from the real video. The simulated hair motion approximately matches the real hair motion in the video.

Figure 10: Top row: short hair in a changing wind. Bottom row: hair rendering with increased translucency and specularity to convey an artistic flavor.