Kelly Ward*
Walt Disney Feature Animation
Burbank, CA 91521

Nico Galoppo
Ming Lin
Department of Computer Science
University of North Carolina at
Chapel Hill
Chapel Hill, NC 27599

Interactive Virtual Hair Salon

Abstract

User interaction with animated hair is desirable for various applications but difficult because it requires real-time animation and rendering of hair. Hair modeling, including styling, simulation, and rendering, is computationally challenging due to the enormous number of deformable hair strands on a human head, elevating the computational complexity of many essential steps, such as collision detection and self-shadowing for hair. Using simulation localization techniques, multi-resolution representations, and graphics hardware rendering acceleration, we have developed a physically-based virtual hair salon system that simulates and renders hair at accelerated rates, enabling users to interactively style virtual hair. With a 3D haptic interface, users can directly manipulate and position hair strands, as well as employ real-world styling applications (cutting, blow-drying, etc.) to create hairstyles more intuitively than previous techniques.

1 Introduction

User interaction with animated hair is useful for various applications. For example, virtual environments depicting human avatars require a hair modeling system capable of animating and rendering hair at interactive rates. Due to the performance requirement, many interactive hair modeling algorithms tend to lack important, complex features of hair, including hair interactions, dynamic clustering of hair strands, and intricate self-shadowing effects. Often, realistic hair appearance and behavior are compromised for real-time interaction with hair.

Because the shape of the hair model and other properties are typically dictated by hairstyling, it is an important step to modeling hair. Virtual environments created for interactive hairstyling are becoming key methods for understanding and specifying these hair properties and are useful for several applications including cosmetic prototyping, the entertainment industry, and cosmetologist training. Accurate virtual hairstyling requires both high performance simulation and realistic rendering to allow interactive use and the incorporation of fine details.

Hairstyles result from physical properties of hair and hair mutual interactions. Thus, hair dynamics must be incorporated to mimic the process of real-world hairstyle creation. Moreover, in the real world people are accustomed to hairstyling by touching hair directly. An intuitive virtual hairstyling tool should incorporate user interaction with dynamic hair, to allow for direct skill transfer

Presence, *Vol. 16, No. 3, June 2007, 237–251*

*Correspondence to wardk@cs.unc.edu.

from training in the virtual environment to hairstyling in the physical world. Until recently, the complexity of animating and rendering hair had been too high to accurately model all of these essential features at desired rates that would allow real-time interaction and styling of hair. As a result, many hairstyling methods ignore dynamic simulation and/or user interaction, which creates an unnatural styling process in comparison to what would be expected in practice.

1.1 Main Results

In this article, we present a physically based virtual hairstyling system that mimics real-world hairstyling processes and requires no knowledge other than common hair manipulation techniques. By using simulation localization and multi-resolution simulation techniques and exploiting graphics hardware rendering acceleration, we developed a physically based virtual hair salon system that animates and renders hair at accelerated rates, allowing users to interactively style virtual hair in a natural manner. With an intuitive 3D haptic interface, users can directly manipulate and position hair strands, as well as employ real-world styling applications (e.g., cutting, wetting, applying styling products) to create hairstyles as they would in the physical world. The main characteristics of our system are the following.

1.1.1 Direct 3D Hair Manipulation with a Haptic Interface. We use a commercially available haptic device to provide an intuitive 3D user interface, allowing the users to directly manipulate the hair in a manner similar to real-world hairstyling.

1.1.2 Visually Plausible Rendering. By exploiting the capability of programmable graphics hardware and the multi-resolution representations, we can render plausible hair appearance due to self-shadowing, self-occlusion, wet surfaces, and so on in real time on current commodity desktop PCs.

1.1.3 Multi-resolution Hairstyling. We achieve interactive hair simulation by using level-of-detail representations and simulation localization techniques, which accelerate dynamics computation and enable adaptive hair clustering and subdivision on the fly. This approach also leads naturally to efficient modeling of complex mutual interactions of hair.

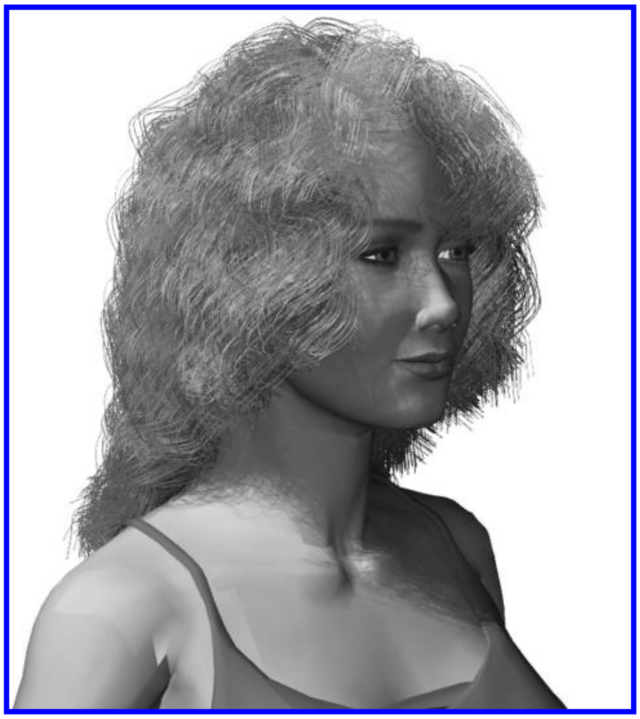

Figure 1. *Hairstyle interactively created with our system.*

1.1.4 Physically-based Interaction. By modeling the hair properties and dynamic behavior in the presence of water and styling products, we introduce the ability to interactively apply hairspray, wet, blow-dry, cut, and manipulate hair as in the physical world, like no other systems can at present.

We demonstrate the flexibility and effectiveness of our system. Figure 1 illustrates a hairstyle created by a naive user using our virtual hair salon system in less than 10 minutes.

1.2 Organization

The rest of the article is organized as follows. Related work is briefly reviewed in Section 2. Section 3 presents an overview of our system. The dynamic simulation and rendering of hair are described in Sections 4 and 5, respectively. Details of user interaction and application features are discussed in Section 6. Finally, we conclude with some results and possible future research directions in Section 7.

2 Related Work

Modeling hair involves hair shape modeling, dynamic hair simulation, and hair rendering. An overview of work in these areas can be found in Magnenat-Thalmann, Hadap, and Kalra, 2000. We have limited our summary of related work to techniques for interactive hair shape modeling (or hairstyling), accelerated hair animation, and rendering techniques using graphics hardware.

2.1 Interactive Hairstyling

Most hairstyling systems involve the manipulation of static hairs. A common approach uses clusters or wisps that couple strands of hair into groups, usually in the form of cylinders, to speed up the styling process (Watanabe & Suenaga, 1992; Chen, Saeyor, Dohi, & Ishizuka, 1999; Yang, Xu, Wang, & Yang, 2000; Xu & Yang, 2001). Generally in these methods a curve defines the shape of the cluster and the user manipulates the shape of the curve to attain a desired hairstyle.

Kim and Neumann (2002) extended the use of generalized cylinders into a multi-resolution control structure allowing the user to edit the hair geometry at global and local levels. Another approach is to use fluid or vector fields to control the direction of hair strands. Hadap and Magnenat-Thalmann (2000) used streamlines of fluid flow to create static hair shapes. Static 3D vector fields were used by Yu (2001) to model curly hairstyles.

Recently, Choe and Ko (2005) developed a method that uses vector fields and trajectories to create complex hairstyles. The trajectories defined by the user constrain the global positions of hair strands and extra parameters are provided to define curliness or randomness within a group of strands to add localized detail. Hair shapes, including ringlet curls, have also been modeled by considering the natural curliness of hair. A recent model uses potential energy minimization of static elastic rods to generate static hair shapes (Bertails, Audoly, Querleux, Leroy, Lévêque, & Cani, 2005).

While many of these methods have produced convincing hairstyles, they do not model the dynamic behaviors of hair and, therefore, lack intuitive user control for hairstyling. Physically based hair modeling methods (such as Anjyo, Usami, & Kurihara, 1992) employ forces, such as gravity, to manage the shape of dynamic hair. On the other hand, the existing physically based modeling methods tend to be too slow to allow the users to interact with the hair to create a desirable style (see Section 2.2). To the best of our knowledge, there exists no method prior to our work that enables the user to interact and style virtual dynamic hair.

2.2 Accelerated Hair Simulation

Recent work in hair animation has used multiresolution, or level-of-detail (LOD), methods to model hair. Many of these techniques accelerate the simulation of hair as well as achieve dynamic clustering effects of hair strands. Based on the observation that strands of close proximity behave similarly, numerous methods have grouped hair strands into fixed sets, or wisps (Watanabe & Suenaga, 1992; Kurihara, Anjyo, & Thalmann, 1993; Daldegan, Magnenat-Thalmann, Kurihara, & Thalmann, 1993; Plante, Cani, & Poulin, 2002) to accelerate hair simulation. Typically a single strand is simulated and the strands located near it follow the same motion.

Bertails, Kim, Cani, and Neumann (2003) created an adaptive control structure called the adaptive wisp tree that splits and merges wisps starting at the tips of the strands to give more detail where it is needed. The works of Ward, Lin, Lee, Fisher, and Macri (2003) and Ward and Lin (2003) couple a similar adaptive clustering technique with three LOD representations for hair: strands, clusters, and strips. An adaptive method was also used by Chang, Jin, and Yu (2002) that generated guide strands on the fly to overly interpolated regions. While many of these techniques were able to accelerate hair simulation, none of these methods was able to simulate hair at fast enough rates to allow a user to actually interact and directly manipulate the hair. Hair strips have been used to approximate complex hair geometry (Koh & Huang, 2001). Though these methods can achieve real-time hair simulation, the resulting hairstyles and motions are limited due to the approximated geometry. Recently, Volino and Magnenat-Thalmann (2004) developed a method for real-time animation of complex hairstyles using a volume free-form deformation scheme. The goal of our work is to interact with hair at "localized" levels while capturing the dynamic clustering effects found in natural hair, making hair modeling schemes that treat hair as a volume inappropriate. Section 3.2 gives an overview of the multi-resolution approach we have adopted and our extensions to it.

2.3 Hair Rendering using Graphics Hardware

Recent advancements in graphics hardware have made it possible to render hair efficiently and accurately. Hair self-shadowing algorithms using GPU features have been developed (Kim & Neumann, 2001; Koster, Haber, & Seidel, 2004; Mertens, Kautz, Bekaert, & Reeth, 2004). Furthermore, Heidrich and Seidel (1998) developed an efficient implementation of anisotropic lighting using texture maps. Section 5 provides a more thorough overview of GPU techniques for hair rendering and the methods we have adapted for our system exploit the newest GPU features.

3 Overview

We first give an overview of the user interface for our interactive virtual hair salon and the multi-resolution methods that are the basis of our hair modeling system.

3.1 User Interface

It is important to have an intuitive user interface that allows a user to work with the system with little training or requirement of outside knowledge. Many other virtual hairstyling methods require its users to have an understanding of complex physical or mathematical formulations, such as fluid dynamics (Hadap & Magnenat-Thalmann, 2000). Meanwhile, in the real world there are many applications performed on hair that small children are capable of performing (e.g., wetting hair). The virtual hair salon system described here uses a SensAble Technologies PHANToM as a 3D user input device. The real-time display of the PHANToM input is rendered using a commercial haptic toolkit called GHOST. The position and orientation of the device are updated and rendered at each frame. Figure 2 illustrates a user operating the system with the PHANToM stylus.

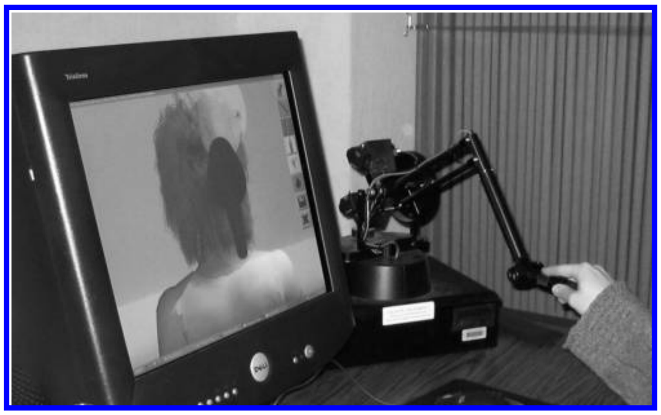

Figure 2. *User interface: PHANToM provides 3D user input and 2D menu buttons are labeled with icons to show applications.*

The virtual hair salon is designed to allow its user to switch between virtual tools (e.g., scissors, hairdryer) using the PHANToM stylus in a seamless manner. For this purpose, a 2D menu is projected onto the 3D scene containing the avatar and hair model. The 2D menu contains graphical icons depicting each tool available to the user. The user interacts with both the 3D scene and the 2D menu using the PHANToM stylus in a seamless fashion. By positioning the stylus over a 2D menu icon and pushing the stylus button, the user chooses a desired application. The position, orientation, and area of influence (the space the application will affect) of the current application is depicted in the scene with a semitransparent cone or triangle (depending on the application). As the user moves the stylus, the area of influence interactively follows the position and orientation of the user's hand in 3D.

3.2 Multi-resolution Hair Modeling

We use the three LOD representations and adaptive grouping and splitting process (Ward et al., 2003; Ward & Lin, 2003) coupled with a simulation localization scheme (see Section 4.3) to achieve interactive hair animation. Figure 3 shows the three LOD representations. A hair hierarchy (Ward & Lin, 2003) is constructed by the continual subdivision of strips, clusters, and strand groups resulting in a hierarchy of LODs of differing representations and resolutions. Throughout the simulation, the hair hierarchy is traversed on the fly to find the appropriate representation and resolution for a given section of hair.

In Ward et al. (2003) and Ward and Lin (2003), an appropriate LOD representation was chosen based on the hair's importance to the viewer. Sections of hair that could be viewed well were simulated and rendered with high detail. Similarly, as a section of hair moves rapidly, a high LOD was used to capture the intricate detail. These criteria aided in accelerating simulation and rendering while losing little visual fidelity.

In this work, our LOD selection criteria differ from prior criteria (Ward et al., 2003; Ward & Lin, 2003) to include areas of user interaction and we expand upon the hair motion criterion. Section 4 describes our LOD selection and areas of activation as determined by the user interaction. Our method is able to achieve greater performance increase over previous LOD methods as we will show in Section 7.

4 Interactive Dynamic Simulation

Hairstyling in the natural world is performed by focusing on specific sections of hair and executing a desired task to a given area. This focus correlates naturally with use of multi-resolution techniques to simulate hair. The level-of-detail hair framework described in Ward et al. (2003) and Ward and Lin (2003) is able to automatically determine the areas of the hair volume that are most significant to the simulation based on the hair's visibility, viewing distance, and motion. Now the user's interaction with the hair defines a new criterion for measuring a hair section's significance. The areas of the hair that the user is directly manipulating (such as by cutting or wetting) is naturally going to draw the user's focus. Moreover, the manipulation by the user will change the hair's physical structure, behavior, and/or visual appearance, implying that more computational resources should be allocated towards these areas to capture such detail.

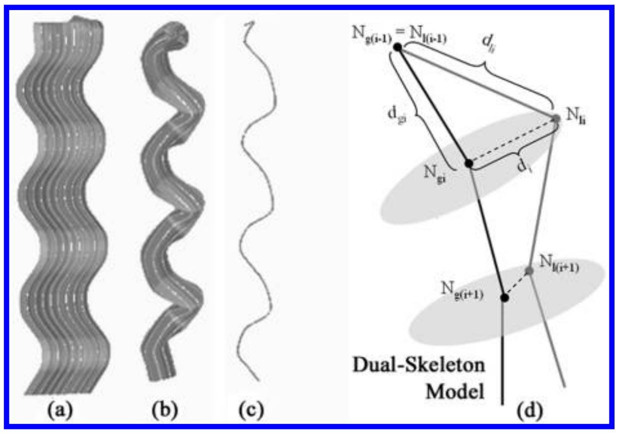

Figure 3. *Level-of-detail (LOD) representations for modeling hair. (a) Hair strip, (b) hair cluster, (c) hair strand, and (d) the dual-skeleton setup for simulating hair.*

In this section, the dynamics model and collision detection techniques we used for accelerated simulation is explained. We also introduce a simulation localization technique based on spatial decomposition that is used to rapidly locate the areas of highest activity. These areas are primarily based on the user's interaction with the hair and are subsequently simulated with high detail while the simulation resolution of the remaining hair sections is significantly reduced.

4.1 Dynamics and Collision Detection

Each LOD representation follows the same dynamics model for motion. We use the dual-skeleton system introduced by Ward, Galoppo, and Lin (2004) for modeling the influences of water and hairspray on hair, as shown in Figure 3d. In this method, two skeletons are used to model the hair. First, there is the global skeleton that is a chain of point masses linked by damped springs. The global skeleton controls the bending of hairs and the stretching of curls. Second, there is the local skeleton that provides a positional guide for the rendered strands. The local skeleton gives a reference for the actual position of the hair sections at any given time and is crucial to the localized hair modeling system. The dual-skeleton system is compelling in that it can capture the details of typical dry hair as well as wet hair and hair with hairspray, or some other styling products applied.

We have utilized the localized collision detection model of Ward et al. (2004) that is based on the dual-skeleton setup and the family of swept sphere volumes (SSVs; Larsen, Gottschalk, Lin, & Manocha, 2000). A swept sphere volume is created by taking a core shape, such as a point, a line, or a rectangle, and growing outward by some offset. The SSVs encapsulate the hair geometry (of any type or resolution LOD) and are used as bounding volumes for collision detection. Each line segment between the control points of the local skeleton constitutes the core line for a line swept sphere (LSS), for strand groups, and clusters. Rectangular swept spheres (RSSs) are created in a similar way for strips by using each rectangular section between control points of the local skeleton that create a strip.

Both hair-hair and hair-object collision detection is performed by checking for intersection between corresponding pairs of SSVs; this is done by calculating the distance between the core shapes and subtracting the appropriate offsets. Hair-object collisions are handled by moving the hair section outside of the body and that hair section is restricted to only move tangential to or away from the object, based on the object's normal direction. Hair-hair collisions are processed by pushing the hair sections apart based on their respective orientations as explained in previous methods (Plante et al., 2002; Ward & Lin, 2003).

4.2 Implicit Integration

Since the hair is interactively manipulated by the user, the numerical integration must be stable under all possible conditions. We use an implicit integration scheme to achieve greater stability. The bending force of a global skeleton is governed by the set of ordinary differential equations (Anjyo et al., 1992):

$$I_i \frac{d^2\theta_i}{dt^2} + \gamma_i \frac{d\theta_i}{dt} = M_{\theta i},\ I_i \frac{d^2\phi_i}{dt^2} + \gamma_i \frac{d\phi_i}{dt} = M_{\phi i}, \quad (1)$$

where I_i is the moment of inertia for the ith control point of the skeleton, i is the damping coefficient, and $M_{\theta i}$ and $M_{\phi i}$ are the θ_i and ϕ_i torque components, respectively.

The torque due to the angular spring forces is computed as:

$$M_\theta = -k_\theta(\theta_i - \theta_{i0}),\ M_\phi = -k_\phi(\phi_i - \phi_{i0}), \quad (2)$$

where k_θ and k_ϕ are the spring constants for θ and ϕ, respectively. The variables θ_0 and ϕ_0 are the specified rest angles and θ and ϕ are the current angle values.

We can rewrite the first equation in Equation (1) as a first-order differential equation by substituting the variable $\omega = \theta$. We get the following formulations for $\Delta\theta$ and $\Delta\phi$ when using the explicit forward Euler method, where $\Delta\theta = \theta(t_0 + h)–\theta(t_0)$ and $\Delta\omega = \omega(t_0 + h)– \omega(t_0)$ and h is the time step value:

$$\begin{pmatrix}\Delta\theta \\ \Delta\omega\end{pmatrix} = h\begin{pmatrix}\omega_0 \\ -\ k_\theta(\theta - \theta_0)\end{pmatrix}$$

By using Taylor series expansion to obtain the first-order approximation, we get

$$\begin{pmatrix}\Delta\theta \\ \Delta\omega\end{pmatrix} = h\begin{pmatrix}\omega_0 + \Delta\omega \\ -\ k_\theta(\theta - \theta_0) - k_\theta\Delta\theta\end{pmatrix}$$

Rearranging the equation and substituting back to Equation (1), the change in angular velocity for the θ-component of a skeleton node point, $\Delta\omega_\theta$, becomes

$$\Delta\omega = \frac{-\,hk_\theta(\theta - \theta_0) - h^2 k_\theta \omega_{\theta 0}}{1 + h^2 k_\theta}$$

where $\omega_{\theta 0} = \omega_\theta(t_0)$ is the angular velocity at time t_0. Here, $\Delta\omega_\theta = \omega_\theta(t_0 + h)–\omega_\theta(t_0)$. Once we have calculated $\Delta\omega_0$, we calculate the change in angular position $\Delta\theta$ from $\Delta\theta = h(\omega_0 + \Delta\omega)$. The same process can be applied to the ϕ-component of the angular position and angular velocity for each control point of a global skeleton. Similarly, implicit integration is used to control the elongation of a global skeleton using linear spring forces between control points.

4.3 Simulation Localization

We use a spatial decomposition scheme to rapidly determine the high activity areas of the hair; these areas are then simulated with finer detail. We use a uniform grid consisting of axis-aligned cells that encompass the area around the hair and human avatar (see Figure 4). Spatial decomposition schemes have been utilized previously for hair-hair interaction methods where sections of hair that are located in the same cell will be tested against each other for overlap. We have extended this process to all features of hair simulation, not just collision detection.

4.3.1 Insertion into the Grid. The polygons of the avatar, or other objects, are placed into the grid to determine potential collisions with the hair. Object positions only need to be updated within the grid if the object is moving, otherwise the initial insertion is sufficient. Every time a section of hair moves, or the skeleton for simulation is updated, its line swept spheres (LSSs) or rectangular swept spheres (RSSs) are inserted into the grid. An SSV is inserted into the grid by determining which cells first contain the core shape of the SSV (line or rectangle), then the offset of the SSVs are used to determine the remaining inhabited cells. Figure 4a shows the grid cells that contain hair geometry.

When the user employs an application (e.g., spraying water, grabbing the hair) the grid is used to indicate which portions of the hair are potentially affected by the user's action. As the user moves the PHANToM stylus, its position and orientation are updated. Each application has an area of influence that defines where in space its action will have an effect. This area is defined as a triangle for the cutting tool and a cone for the remaining tools. The cone of influence is defined by the application's position, orientation (or direction pointed), length (how far it can reach), and cutoff angle (determining its radius along its length). These properties define the cone's position in the grid. Inserting the cone becomes similar to inserting an LSS, but the offset becomes a variable of distance along the core line (an SSV has a constant offset along its core shape). The triangle for cutting is defined by the space between the open blades of a pair of scissors.

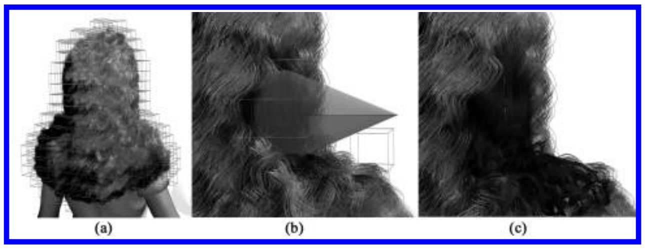

Figure 4. *(a) All of the grid cells that contain hair geometry. (b) Highlights the cells that will be affected by the current application (applying water). (c) Water is applied to some hair; grid allows us to localize each application.*

4.3.2 Retrieval from the Grid. Once information has been inserted or updated in the grid, it is retrieved to determine where to check for potential collisions and user interaction. Grid cells that contain both impenetrable triangles (from the avatar or another object in the scene) and hair geometry are marked to be checked for hair-object collision. Only these cells contain a potentially colliding pair. Similarly, any grid cell containing more than one section of hair is marked to be checked for hair-hair collisions. Likewise, the grid maintains a list of grid cells where the user interaction cone or triangle has been inserted. Any of these grid cells that contain hair geometry are returned and the sections of hair within the cell are independently checked to see if they fall within the area of influence; (see Figure 4).

4.4 Multi-Resolution Simulation with the Grid

The grid aids us to localize our simulation on the areas of highest importance to the user. These areas are defined based on their distance from the viewer, visibility, motion, and the user's interaction with the hair. We adopted the method for choosing a level of detail based on distance and visibility created previously by Ward et al. (2003), but have used our grid-based system to expand on the motion criteria and to include the user's interaction with the hair.

The motion of the hair is highly pertinent to the amount of detail needed to simulate the hair. Most applications performed on hair are localized to a small portion of the hair; most of the hair thus lies dormant. The sections of hair that are dormant are modeled with a lower LOD representation and resolution, determined by comparison against velocity thresholds, but we have gone a step further by effectively turning off simulation for areas where there is no activity. Each grid cell keeps track of the activity within the cell, tracking the hair sections that enter and exit the cell. When the action in a given cell has ceased and the hair sections in the cell have a zero velocity, there is no need to compute dynamic simulation due to gravity, spring forces, or collisions. The positions of the hair sections are thus frozen until they are reactivated. The cell is labeled dormant and does not become active again until either the user interacts with the cell or until a new hair section enters the cell.

As the previous section explained, we can determine which grid cells the user is focusing on by inserting the cone or triangle of influence into the grid. These grid cells are then activated and each hair section in the cells become active as well. The actions of the user can influence the activity in cells they are not directly focused on (for example, blow-drying hair will push sections of hair into new cells). It then remains important to continually track the motion of hair sections to determine active or dormant cells. As curly hair becomes wet, the curls can elongate creating activity below the actual placement of water. By tracking the motion of wet hair caused by its changing mass, the control points located below the water placement, and the cells they occupy, can be activated even if these cells were previously dormant. Neighboring control points share spring forces that control their respective bending and stretching or compressing. An active control point can then awaken its dormant neighboring control point if there is a force that warrants it.

Rapid determination of the active cells and hair sections allows us to place the computational resources towards dynamic simulation for the hairs of highest interest to the user.

5 Real-Time Rendering

In order to guarantee a convincing and interactive experience for the user, our rendering algorithm has to run in real time and produce realistic images at the same time. We have taken advantage of the newest features on graphics hardware to implement a physically plausible hair rendering model with very efficient self-shadowing.

5.1 Realistic Hair Lighting

We model the hair strand reflections by anisotropic light scattering from a cylindrical surface as proposed by Kajiya and Kay (1989), augmented by recent observations made by Marschner, Jensen, Cammarano, Worley, and Hanrahan (2003) that one must also account for the multiple modes of scattering that occur inside and on the surface of the cylinders. They showed that only a limited amount of colored reflection comes from diffuse scattering, whereas there is a major contribution of uncolored, specular reflection. We implemented two separate specular highlights, due to the multiple modes of scattering inside and on the surface of the hair fibers. The primary highlight is highly specular and shifted slightly towards the hair root, whereas the secondary highlight has a wider falloff and is shifted towards the hair tip. We compute both specular terms by shifting the hair tangent in Kayija's original formulation (Kajiya & Kay, 1989) towards the hair root and towards the hair tip respectively, applying separate falloff exponents.

Using the shifted tangents in the formulation proposed in Heidrich and Seidel (1998), the new twofold specular term of the hair shading thus becomes:

$$\begin{aligned} \mathrm{l}_{\text{specular}} &= (k_s^1\langle V,R^1\rangle^{n_1} + k_2^2\langle V,R^2\rangle^{n_2}) \cdot I_i\langle V,R^i\rangle \\ &= \sqrt{1 - \langle L,T^i\rangle^2}\,\sqrt{1 - \langle V,T^i\rangle^2} - \langle L,T^i\rangle\langle V,T^i\rangle \end{aligned}$$

The tangents are shifted in the plane formed by the hair tangent and the light direction. We take into account hair fiber roughness due to tilted surface scales by multiplying the second highlight shift with a noise pattern. All operations were performed in a fragment program for efficiency.

5.2 Real Time Self-Shadowing

Realistic hair self-shadowing effects are hard to implement efficiently, due to the large amount of dynamic geometry and due to the fineness of the hair strands. Regular shadow maps fail because the high frequency in the geometry forces us to use sampling rates beyond realistic values (both in terms of performance and memory usage). We make use of two recently introduced features of graphics hardware to achieve real-time self-shadows. The first, multiple render targets, allows us to output 16 floating point values in one rendering pass. The second, floating point blending, makes high-precision blending of shadow values possible.

Opacity shadow maps (OSMi Kim & Neumann, 2001) were a first attempt at efficient self-shadow generation for hair, but the algorithm is still not quite interactive. In this technique, self-shadows are generated by accumulating the opacity α of the strands hit by the light rays along the light direction in the hardware framebuffer. Unfortunately, because of the iterative nature of the algorithm, an expensive multiple pass rendering algorithm with an expensive read-back to the CPU is required. Moreover, the accumulation of opacity through blending in the framebuffer was limited to 8-bit integers. This leads to artifacts due to limited precision. For 10 opacity shadow maps, this technique achieves a rate of less than one frame per second.

Our algorithm is based on recent GPU features (NVIDIA, 2005); it generates 16 opacity shadow maps in only one pass with multiple render targets, plus an extra pass for the actual rendering, with no requirement for depth ordering. As in Kim and Neumann (2001), the opacity maps are placed at uniformly sampled distances from the eye, orthogonal to the view direction. Each of the four render targets holds four opacity maps, one in each 16-bit floating point component. In the opacity map generation pass, a GPU fragment program compares the distance of the incoming hair fragment from the eye to the distance of each of the 16 maps from the eye. If the hair fragment is closer to the eye than a particular opacity map, it means that it contributes to the opacity of the opacity map in question, thus we give it a positive contribution. Otherwise, the contribution is zero. By enabling 16-bit floating point blending, we achieve high-precision accumulation of the opacity of all the fragments into the appropriate opacity maps.

In the rendering pass, a vertex program first computes weights according to the relative position of the vertex to each of the opacity maps. This corresponds to determining the weights of a hat reconstruction filter. These weights are then used in the fragment program for reconstruction. In effect, it interpolates linearly between the opacities of the two closest opacity maps. The weights are computed as follows:

$$w_p^n = \frac{\max(0, 1 - |z_n - z_p|)}{\Delta}$$

where w_p^n is the weight of point p for the nth opacity map, z_n and z_p are the distances of the nth opacity map and point p to the eye. The variable Δ is the distance between the opacity maps.

Our method has an additional advantage over the technique proposed by Kim and Neumann (2001): instead of assigning the hair geometry to depth bins on the CPU, we perform the depth comparisons at the fragment level (in fragment programs on the GPU), therefore avoiding the expensive read-back to the CPU. In addition, we can use the the opacity generation pass to generate a regular shadow map for the head, with no overhead incurred. The blocking of light by the head is then easily determined by regular shadow mapping techniques, and allows us to cast hair shadows on the head and the body. Simply sampling the OSM does not work, because the OSM contains no information about whether the fragment is visible from the light or not, similar to projected textures. Instead, we combine the OSMs with visibility information obtained from the regular shadow map, and take the maximum of both values.

5.3 Limitations

This approach can incorrectly make fully lit fragments that lie between opacity maps appear darker because of the linear interpolation between the lit and non-lit opacity map plane. We solve this partially by virtually pushing the opacity map planes back, so that the fully lit fragment preferentially samples lit planes. Another approach would be to use quadratic interpolation.

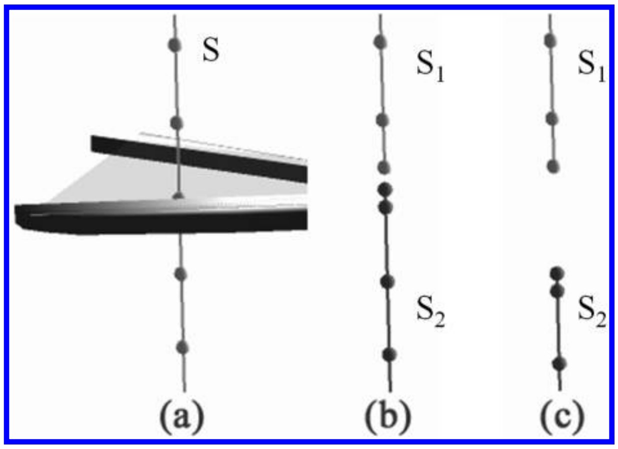

Figure 5. *(a) Open blades of scissors define cutting triangle. (b) and (c) Top skeleton remains attached to scalp, bottom skeleton falls away after cut.*

6 User Interaction and Applications

Given our system for simulating and rendering hair described in the previous sections, a user can now directly interact with hair through the 3D user interface and use operations commonly performed in hair salons. The operations supported in our system include applying water, hair spray, and mousse to the hair, grabbing and moving sections of hair, using a hair dryer, and cutting the hair. The rest of this section describes each of these hair modeling features and their influences on hair dynamics and LOD transitioning.

6.1 Haircutting

Cutting hair is crucial for changing a hairstyle. A common approach for specifying hair length has been to control the length of the hair through a parameter that either makes the hair grow or shorten. Some works have allowed the length parameter to change locally through either curve drawing (Xu & Yang, 2001; Kim & Neumann, 2002), or via a 2D length map that correlates to the distribution of strands on the scalp (Choe & Ko, 2005). Alternatively, Lee and Ko (2001) cut hair with a cutting surface; hairs that intersect the surface are clipped to the surface shape. Our cutting method builds on the words of Lee and Ko (2001) to model cutting of hair performed with scissors as is used in a salon. We model all the features of the cut, including capturing the hair that falls away or that is cut off. The location for cutting is defined by a triangle formed by the space between the open blades of scissors. Hair skeletons that intersect this triangle are then cut. At the cutting location, the skeleton S is split into two separate skeletons, S_1 and S_2, S_1 remains attached to the scalp, while the latter, S_2, falls away.

At the intersection of skeleton S and the cutting triangle two new control points are created. One control point becomes the last point of skeleton S_1, while the second becomes the first point of S_2 that falls away after the cut; see Figure 5. The geometry of the fallen hairs remains consistent with the geometry of the hair below the sever point before the cut is performed; curliness, wetness, hair distribution, and other properties are maintained in the fallen hair segments. Skeleton S_2 is simulated with two free endpoints; dynamics, including collision detection and response, are still enforced.

6.2 Applying Water

Wet hair is modeled using the technique described in Ward et al. (2004). When water is applied to the hair,

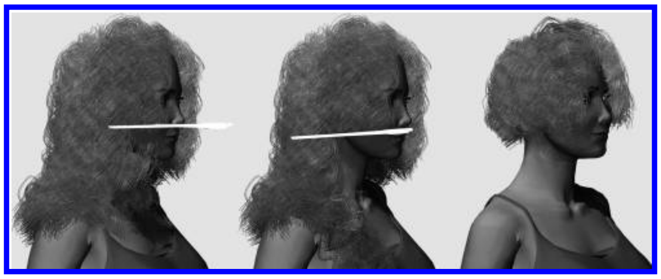

Figure 6. *Example of haircutting, far right shows final style.*

the mass points of the global skeleton become heavier with the mass of the water. The overall motion of the hair is limited due to the extra weight and if the hair is curly, the global-skeleton will stretch under the extra weight and the curls will lengthen as expected. The volume of the hair in the areas where water is applied is decreased by constricting the radius of the current hair representation (strand grouping, cluster, or strip); these hair segments are then rendered to show the wet appearance (see Figure 6).

6.3 Applying Hairpsray and Mousse

Hair spray is simulated on the hair by increasing the spring constants of the global skeleton where it is applied. Moreover, dynamic bonds (Ward et al., 2004) are added between sections of hair that are in contact when the hair spray is applied. Dynamic bonds are extra spring forces that model the adhesive quality of hairspray to make the hair move as a group throughout subsequent motions.

Though the effects of mousse vary by brand, we have chosen to model a "volume-defining" mousse, which adds volume to hair. We can inject volume into our hair model by growing the radii of the hair sections it affects. This process makes the hair fuller without adding more strands or skeleton models.

6.4 Grabbing and Moving Hair

More complex hairstyles can be created by grabbing areas of hair and moving them. Typically when hair is clasped in the real world, a group of strands are selected at a local point along the strands. In the simulation, the user presses the stylus button and the control points that fall within the cone of influence are decided. Among these selected control points, only one control point per dual skeleton is permitted to be grabbed; this rule correlates to the localized grabbing of a hair cluster when styling hair. If multiple control points of a single dual skeleton fall within the cone, the point that comes closest to the cone's center will be chosen.

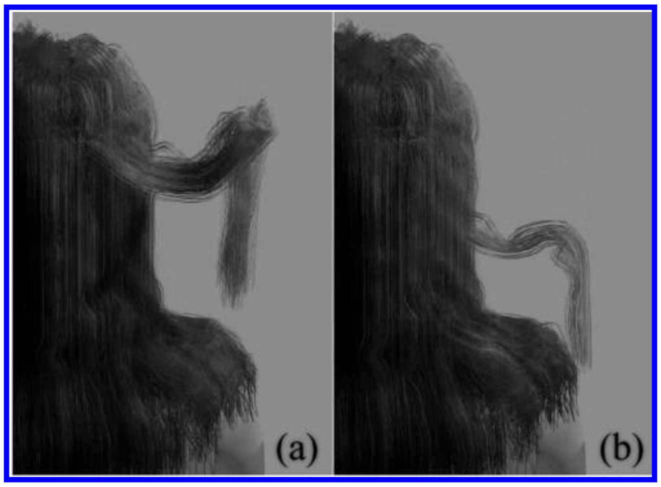

Figure 7. *(a) User grabs and pulls a section of hair. (b) User releases grip and hair falls back into place.*

At this point, control points are in a grabbed state and are referred to as grabbed points. In the grabbed state, as the user moves the stylus, the grabbed point will follow the motion of the stylus. A grabbed point cannot be pulled beyond its normal reach span (decided by its position in the dual skeleton). The length of the hair is always maintained so that the lengths above the grabbed point and below it are of consistent length while the point is moving.

When the user releases the grabbed point(s), he or she releases the button of the stylus and the former grabbed points will fall due to gravity. Figure 7 shows the user grabbing and releasing a section of hair.

6.5 Hair Dryer

Hair dryers are one of the most common tools in a hair salon. When the stylus button is pressed, a strong constant force is applied in the direction of its orientation. Any control points that fall within the cone of influence receive this strong force. Moreover, if a wet control point (see Section 6.2) is influenced by the hair dryer, the control point will "dry"; the amount of water will decrease over the length of exposure dependent on the strength of the hair dryer force. The hair dryer also has the ability to break the dynamic bonds created by hair spray (Section 6.3).

When two control points linked by a dynamic bond are moved apart past a given threshold, the bond will break. The stiffened spring forces, however, remain strong and can only be broken by the application of water.

6.6 Hierarchy Inheritance

The applications described in the previous section change the properties of the hair whether permanently or temporarily; therefore a method for propagating these properties through LOD switching has to be employed. The effects of water, hair spray, mousse, and cutting are propagated up the hair hierarchy by averaging the values of the children dual skeletons when they transform into their parent. Therefore, water values, spring stiffness, thickness, and length values are averaged. A parent skeleton also takes over the same dynamic bonds of its children; any hair sections connected to a child skeleton then become connected to the parent.

Traveling down the hierarchy, child skeletons inherit the same values as their parents. For example, a parent control point having 80% wetness will pass on 80% wetness to each of its children. The water value is not divided among the children because water would be lost throughout the system. The values are averaged when going up the hierarchy because the water is spread throughout all of the hairs. A hair section with 0% wetness combines with a section of 100% wetness and the water spreads throughout both resulting in 50% wetness for its parent. Similarly, spring forces, thickness, and lengths are passed onto the children when traveling down the hierarchy. Any dynamic bonds that are created on the parent will be passed on to its children. Moreover, if hair spray has been applied to a parent control point, when it splits into its children dynamic bonds then connect each child.

Cutting hair has the most significant impact on the hair hierarchy because the number of control points per skeleton may be altered. Inheriting a cut through adaptive splitting is a simple process. If a hair group loses the last *x* number of controls points from a cut, during a transition, each child skeleton will lose the same *x* number of control points and all the child hair groups will be the same length as the parent. In the opposite direction, adaptive merging for simplifying the simulation is more constrained. As child skeletons prepare to transition into their parent, the number of control points on each child is compared. If all the child skeletons have been cut and still contain the same number of control points, then their lengths are averaged and the transition into the parent occurs. However, if the child skeletons have been cut but do not contain the same number of control points, the transition is not permitted. Using the simulation localization technique described in Section 4, the dynamic simulation of the hair groups can still be accelerated if they lie in dormant cells.

7 Results and Discussion

Our virtual hair salon system is a prototype system that demonstrates the usefulness of our multi-resolution techniques for interactive hair modeling and was implemented in C++. The initial hairstyles are loaded as a preprocess.

7.1 Discussion

Our virtual hair salon system has enabled physically based user interaction with dynamic hair, while modeling several common hair salon applications. Figure 8 shows a comparison of real hair under the influence of common hair modeling applications with our virtual salon results under the same conditions. Level-of-detail representations coupled with our simulation localization scheme have accelerated the animation of hair so that a user can actually interact with it.

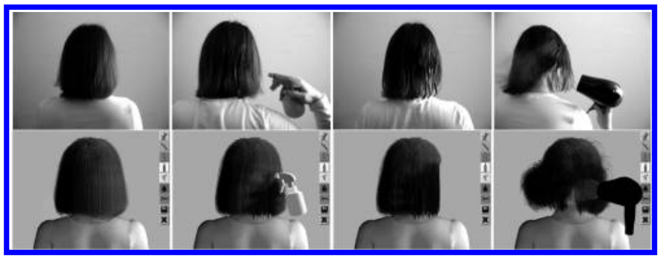

Figure 8. *Comparison between real (top) and virtual use of common hair salon activities (from left to right). (1) Normal dry hair, (2) applying water, (3) some wet, some dry hair, (4) blow-drying hair.*

Dynamic simulation, including implicit integration, LOD selection, hair applications (wetting, cutting, etc.), and collision detection, to create the hair model shown in Figure 1 ran at an average of 0.092 sec per frame. This figure comprised between 37 to 296 skeleton models, determined on-the-fly throughout the simulation, with an average of 20 control points each. At the finest resolution, the model contained 8,128 rendered strands; throughout the simulation the rendering LOD contained between 6K and 1,311K rendered vertices. Lighting and shadow computations on the GPU were performed in 0.058 sec/frame on average. Using a fixed geometry of 304 K rendered vertices, our rendering algorithm can run at approximately 15 fps for 16 opacity shadow maps. The benchmarks were measured on a desktop PC equipped with an Intel Xeon 2.8 GHz processor with 2.0 GB RAM and an NVIDIA GeForce 6800 graphics card. Figure 9 shows a detailed performance comparison over the course of an entire simulation between wisps (which are used as the baseline of comparison), LODs alone, and our LODs coupled with simulation localization. The performance of the LODs with simulation localization varies over time due to the user performing different applications on the hair. However, it is clear that the LODs with simulation localization are able to outperform wisps alone as well as LODs alone.

7.2 Limitations and Future Work

Our prototype system demonstrates the usefulness of our multi-resolution techniques for interactive hair modeling. There are several advancements that can be performed as future work. The research areas we are interested in pursuing include:

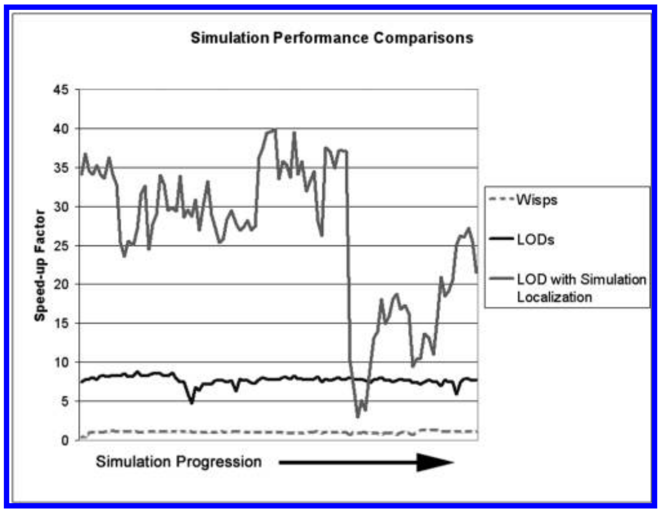

Figure 9. *Simulation performance comparison. Shows the factor of speed-up for LODs with simulation localization and LODs over wisps alone. Here, the average runtime of the wisps is used as the baseline for comparison (value of 1 on this chart). Over the course of this simulation, the camera remained at a consistent distance from the figure and the viewer primarily faced the back of the avatar—causing distance and occlusion tests to have a small overall impact on the LOD choice. Note that the LODs with simulation localization outperforms both wisps and LODs alone, though the simulation varies over time as the user employs different applications.*

- Clasping hair to mimic the effects of barrettes or bobby pins to allow for hairstyles such as ponytails with dynamic simulation.
- Experimenting with two-hand interaction with hair through the use of haptic gloves. More interesting interactions can be performed along these lines including the creation of braids and other more complex hairstyles. Furthermore, we would be able to model physically based methods for curling hair that involve rollers or curling irons where the hair is held in one hand and the roller or curling iron in the other.
- Integrating high-fidelity force-feedback into the haptic system for more realistic user interaction.

- Explore the use of adaptive grids so the resolution of the grid can vary along with the simulation, providing another level of multi-resolution interaction.

Moreover, our rendering approach could be combined with recent works (Mertens et al., 2004; Bertails, Menier, & Cani, 2005) that focus on rendering animated hair for further rendering acceleration.

8 Summary

We presented a system involving an intuitive 3D user interface and methods for animating and rendering hair that allows for a user to interactively manipulate hair through several common hair salon applications. This system provides a level of user interaction that has before been too complex to achieve. We are interested in exploring the use of two-handed haptic gloves with our system to provide higher fidelity force feedback to the user, while allowing for further interaction capability and the creation of even more complex hairstyles.

Acknowledgments

The authors would like to thank our reviewers for the useful feedback which helped to improve the work. This work was supported in part by ARO Contracts DAAD19-02-1-0390 and W911NF-04-1-0088, NSF awards 0400134, 0429583, and 0404088, DARPA/RDECOM Contract N61339-04-C-0043 and the Disruptive Technology Office.

References

Anjyo, K., Usami, Y., & Kurihara, T. (1992). A simple method for extracting the natural beauty of hair. *ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series,* 111–120.

Bertails, F., Audoly, B., Querleux, B., Leroy, F., Lévêque, J.-L., & Cani, M.-P. (2005). Predicting natural hair shapes by solving the statics of flexible rods. *Eurographics* (short papers).

Bertails, F., Kim, T.-Y., Cani, M.-P., & Neumann, U. (2003). Adaptive wisp tree—A multiresolution control structure for simulating dynamic clustering in hair motion. *ACM SIGGRAPH Symposium on Computer Animation,* 207–213.

Bertails, F., Menier, C., & Cani, M.-P. (2005). A practical self-shadowing algorithm for interactive hair animation. *Proceedings of Graphics Interface,* 71–78.

Chang, J. T., Jin, J., & Yu, Y. (2002). A practical model for hair mutual interactions. *ACM SIGGRAPH Symposium on Computer Animation,* 73–80.

Choe, B., & Ko, H.-S. (2005). A statistical wisp model and pseudophysical approaches for interactive hairstyle generation. *IEEE Transactions on Visualization and Computer Graphics, 11*(2).

Chen, L., Saeyor, S., Dohi, H., & Ishizuka, M. (1999). A system of 3D hairstyle synthesis based on the wisp model. *The Visual Computer, 15*(4), 159–170.

Daldegan, A., Magnenat-Thalmann, N., Kurihara, T., & Thalmann, D. (1993). An integrated system for modeling, animating and rendering hair. *Computer Graphics Forum, 12*(3), 211–221.

Hadap, S., & Magnenat-Thalmann, N. (2000). Interactive hair styler based on fluid flow. *Computer Animation and Simulation,* 87–100.

Heidrich, W., & Seidel, H.-P. (1998). Efficient rendering of anisotropic surfaces using computer graphics hardware. *Proceedings of Image and Multi-Dimensional Digital Signal Processing Workshop (IMDSP).*

Kurihara, T., Anjyo, K., & Thalmann, D. (1993). Hair animation with collision detection. In *Proceedings of Computer Animation,* 128–138.

Koh, C., & Huang, Z. (2001). A simple physics model to animate human hair modeled in 2D strips in real time. *Computer Animation and Simulation,* 127–138.

Koster, M., Haber, J., & Seidel, H.-P. (2004). Real-time rendering of human hair using programmable graphics hardware. *Computer Graphics International (CGI),* 248–256.

Kajiya, J., & Kay, T. (1989). Rendering fur with three dimensional textures. *ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series,* 271–280.

Kim, T.-Y., & Neumann, U. (2001). Opacity shadow maps. *Rendering Techniques,* 177–182.

Kim, T.-Y., & Neumann, U. (2002). Interactive multiresolution hair modeling and editing. *ACM SIGGRAPH, ACM Transactions on Graphics, 21*(3), 620–629.

Larsen, E., Gottschalk, S., Lin, M., & Manocha, D. (2000). Distance queries with rectangular swept sphere volumes.

Proceedings of IEEE International Conference on Robotics and Automation.

Lee, D.-W., & Ko, H.-S. (2001). Natural hairstyle modeling and animation. *Graphical Models, 63*(2), 67–85.

Marschner, S., Jensen, H. W., Cammarano, M., Worley, S., & Hanrahan, P. (2003). Light scattering from human hair fibers. *ACM SIGGRAPH, ACM Transactions on Graphics, 22* (3), 780–791.

Mertens, T., Kautz, J., Bekaert, P., & Reeth, F. V. (2004). A self-shadow algorithm for dynamic hair using density clustering. *Proceedings of Eurographics Symposium on Rendering.*

Magnenat-Thalmann, N., Hadap, S., & Kalra, P. (2000). State of the art in hair simulation. In *International Workshop on Human Modeling and Animation,* 3–9.

NVIDIA. (2005). Tech. report, NVIDIA. Available at http://www.nvidia.com/.

Plante, E., Cani, M.-P., & Poulin, P. (2002). Capturing the complexity of hair motion. *Graphical Models, 64*(1).

Volino, P., & Magnenat-Thalmann, N. (2004). Animating complex hairstyles in real-time. *ACM Symposium on Virtual Reality Software and Technology.*

Ward, K., Galoppo, N., & Lin, M. C. (2004). Modeling hair influenced by water and styling products. *Proceedings of the 17th International Conference on Computer Animation and Social Agents (CASA 2004),* 207–214.

Ward, K., & Lin, M. C. (2003). Adaptive grouping and subdivision for simulating hair dynamics. *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications,* 234–243.

Ward, K., Lin, M. C., Lee, J., Fisher, S., & Macri, D. (2003). Modeling hair using level-of-detail representations. *Proceedings of the 16th International Conference on Computer Animation and Social Agents (CASA 2003),* 41–47.

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

Xu, Z., & Yang, X. D. (2001). V-Hairstudio: An interactive tool for hair design. *IEEE Computer Graphics and Applications, 21*(3), 36–42.

Yu, Y. (2001). Modeling realistic virtual hairstyles. *Proceedings of Pacific Graphics,* 295–304.

Yang, X. D., Xu, Z., Wang, T., & Yang, J. (2000). The cluster hair model. *Graphics Models and Image Processing, 62*(2), 85–103.

This article has been cited by:

1. Yuki Horita, Haruka Yamashita. 2019. Bayesian network considering the clustering of the customers in a hair salon. *Cogent Business & Management* **6**:1, 1641897. [Crossref]
2. Yijie Wu, Yongtang Bao, Yue Qi. Spatial-Temporal Editing for Dynamic Hair Data 336-341. [Crossref]
3. Yun (Raymond) Fei, Henrique Teles Maia, Christopher Batty, Changxi Zheng, Eitan Grinspun. 2017. A multi-scale model for simulating liquid-hair interactions. *ACM Transactions on Graphics* **36**:4, 1-17. [Crossref]
4. Nicholas Michael, Maria Drakou, Andreas Lanitis. 2017. Model-based generation of personalized full-body 3D avatars from uncalibrated multi-view photographs. *Multimedia Tools and Applications* **76**:12, 14169-14195. [Crossref]
5. A. Galliano, M. Lheur, R. Santoprete. 2015. Analysing the movement of a hair swatch using video and image analysis: A promising technique for exploring the dynamic properties of hair. *International Journal of Cosmetic Science* **37**:1, 56-62. [Crossref]
6. Alexandre Derouet-Jourdan, Florence Bertails-Descoubes, Gilles Daviet, Joëlle Thollot. 2013. Inverse dynamic hair modeling with frictional contact. *ACM Transactions on Graphics* **32**:6, 1-10. [Crossref]
7. W. Rungjiratananon, Y. Kanamori, T. Nishita. 2012. Wetting Effects in Hair Simulation. *Computer Graphics Forum* **31**:7, 1993-2002. [Crossref]
8. Dan Wang, Xiujuan Chai, Hongming Zhang, Hong Chang, Wei Zeng, Shiguang Shan. A novel coarse-to-fine hair segmentation method 233-238. [Crossref]
9. Cem Yuksel, Scott Schaefer, John Keyser. 2009. Hair meshes. *ACM Transactions on Graphics* **28**:5, 1-7. [Crossref]
10. Ugo Bonanni, Nadia Magnenat-Thalmann, Petr Kmoch. Tool-based hairstyling metaphors 122-127. [Crossref]
11. Vinícius Jurinic Cassol, Fernando Pinho Marson, Soraia Raupp Musse. Procedural Hair Generation 185-190. [Crossref]
12. Andrew Selle, Michael Lentine, Ronald Fedkiw. 2008. A mass spring model for hair simulation. *ACM Transactions on Graphics* **27**:3, 1-11. [Crossref]
13. Jamie Wither, Florence Bertails, Marie-Paule Cani. Realistic Hair from a Sketch 33-42. [Crossref]
14. Kelly Ward, Florence Bertails, Tae-yong Kim, Stephen R. Marschner, Marie-paule Cani, Ming C. Lin. 2007. A Survey on Hair Modeling: Styling, Simulation, and Rendering. *IEEE Transactions on Visualization and Computer Graphics* **13**:2, 213-234. [Crossref]