A Survey on Hair Modeling: Styling, Simulation, and Rendering

Kelly Ward Florence Bertails Tae-Yong Kim Stephen R. Marschner Marie-Paule Cani Ming C. Lin

***Abstract*— Realistic hair modeling is a fundamental part of creating virtual humans in computer graphics. This paper surveys the state of the art in the major topics of hair modeling: hairstyling, hair simulation, and hair rendering. Because of the difficult, often unsolved, problems that arise in all these areas, a broad diversity of approaches are used, each with strengths that make it appropriate for particular applications. We discuss each of these major topics in turn, presenting the unique challenges facing each area and describing solutions that have been presented over the years to handle these complex issues. Finally, we outline some of the remaining computational challenges in hair modeling.**

***Index Terms*— Hair modeling, physically-based simulation, hardware rendering, light scattering, user-interaction, collision handling**

I. Introduction

Modeling hair is essential to computer graphics for various applications; however, realistically representing hair in structure, motion and visual appearance is still an open challenge. Hair modeling is important for creating convincing virtual humans for many diverse CG applications.

Hair modeling is a difficult task primarily due to the complexity of hair. A human head typically consists of a large volume of hair with over 100,000 hair strands. However, each individual hair strand is quite small in diameter. Considering this duality, researchers have examined whether hair should be treated as an overall volume or as individual interacting hair strands. Currently, there is no method that has been accepted as the industry standard for modeling hair.

In the real world, the structure and visual appearance of hair varies widely for each person, making it a formidable task for any one modeling scheme to capture all diversities accurately. Moreover, due to the high complexity of hair the algorithms that provide the best visual fidelity tend to be too computationally overwhelming to be used for interactive applications that have strict performance requirements. The diverse applications that incorporate hair modeling each possess their own challenges and requirements, such as appearance, accuracy, or performance. Additionally, there are still unknown properties about real hair, making the creation of a physically correct modeling scheme elusive at this time.

In this survey, we will discuss the primary challenges involved with modeling hair and also review the benefits and limitations of methods presented in the past for handling these complex issues. Furthermore, we will give insight for choosing an appropriate hair modeling scheme based on the requirements of the intended application.

Walt Disney Feature Animation
EVASION-INRIA, Grenoble, France
Rhythm & Hues Studio
Cornell University
EVASION/INRIA & INP Grenoble, France
University of North Carolina at Chapel Hill

A. Hair Modeling Overview

As illustrated by Magnenat-Thalmann *et al.* [1], hair modeling can be divided into three general categories: hairstyling, hair simulation, and hair rendering. Hairstyling, viewed as modeling the shape of the hair, incorporates the geometry of the hair and specifies the density, distribution, and orientation of hair strands. Hair simulation involves the dynamic motion of hair, including collision detection between the hair and objects, such as the head or body, as well as hair mutual interactions. Finally, hair rendering entails color, shadows, light scattering effects, transparency, and anti-aliasing issues related to the visual depiction of hair on the screen.

While there are several known techniques for hair modeling, hair research began by viewing hair as individual strands, or one-dimensional curves in three-dimensional space [2], [3]. Building on these foundations, researchers have focused on how these individual strands interact with each other to comprise the whole volume of a full head of hair. Though several paths have been followed, modeling a full head of hair remains an open challenge due to the geometric complexity and thin nature of an individual strand coupled with the complex collisions and shadows that occur among the hairs. We have considered the following general questions for analyzing these methods in several categories:

- **Hair Shape**: Can the method handle long, curly or wavy hair or is it limited to simpler short, straight styles?
- **Hair Motion**: Is the method robust enough to handle large, erratic hair motion that can cause dynamic grouping and splitting of hair clusters as well as complex hair collisions?
- **Performance vs. Visual Fidelity**: Is the primary focus of the method to model visually realistic hair, to model hair quickly and efficiently, or to offer a balance between performance speed and visual fidelity of the virtual hair?
- **Hardware Requirements**: Does the method rely on specific GPU features or other hardware constraints or does it have cross-platform compatibility?
- **User Control**: To what degree does the user have control over the hair? Is the control intuitive or burdensome?
- **Hair Properties**: Can the method handle various hair properties (e.g. coarse vs. fine, wet vs. dry, stiff vs. loose) and allow for these values to vary on the fly throughout the application?

Given the factors above, a hair modeling method may typically have strength in some areas, but little capability in addressing others. Future research endeavors strive to lessen the gap between these areas. The goal is to create an ideal unified hair modeling structure that can effortlessly handle various hair shapes, motions, and properties, while giving the desired level of intuitive user control in a manner that achieves a fast performance with photo-realistic hair. Presently, hair modeling is far from this ideal.

B. Applications and Remaining Problems

The future research in hair modeling may be driven by applications. Cosmetic prototyping desires an exact physical and chemical model of hair for virtually testing and developing products; currently, there is little measured data on the mechanical behaviors of hair to accurately simulate how a product will influence hair's motion and structure. As a result, there is no known hair modeling method that can simulate the structure, motion, collisions and other intricacies of hair in a physically-exact manner.

In contrast, in the entertainment industry, such as with feature animation, a physically correct hair modeling scheme is not necessarily desirable. In fact, it is frequently a goal to model a physically impossible hairstyle or motion. In these cases, a high degree of user control is needed to direct the hair in a desired way, which is a time-consuming and costly endeavor due to the magnitude of the hair volume. Methods to accelerate and ease this process would be valued additions to hair modeling research.

Another arena that requires hair modeling is interactive systems, such as virtual environments and videogames. In these applications, the performance speed of the virtual hair is the main emphasis over its appearance. Though recent efforts have increased the efficiency of hair modeling algorithms, there still remains a desire to heighten the quality of the resulting hair to capture more hair shapes, motions and properties.

The remainder of this paper is organized as followed. Hairstyling techniques are reviewed in Section II. Methods for simulating dynamic hair are presented in Section III. Section IV describes the properties of hair related to its interaction with light, followed by techniques for rendering hair. Finally, Section V presents new challenges facing hair research and applications in each of these categories.

II. Hairstyling

Creating a desired hairstyle can often be a long, tedious, and non-intuitive process. In this section, the main properties of real hair that control its final shape are explained, followed by the methods for styling virtual hair. Techniques for hairstyling can be categorized into three general steps: attaching hair to the scalp, giving the hair an overall or global shape, and managing finer hair properties.

A. Hair Structural and Geometric Properties

There is a diverse spectrum of hair shapes, both natural and artificial. Depending on their ethnic group, people can have naturally smooth or jagged, and wavy or curly hair. These geometric features can result from various structural and physical parameters of each individual hair strand, including the shape of its cross-section, its level of curliness, or the way it comes out of the scalp [4], [5]. Hair scientists categorize hair types into three main groups: Asian hair, African hair, and Caucasian hair. Whereas an Asian hair strand is very smooth and regular, with a circular cross-section, an African hair strand looks irregular, and has a very elliptical cross-section. Caucasian hair ranges between these two extrema, from smooth to highly curly hair.

Furthermore, most people typically cut and style their hair in various ways through bangs, ponytails, braids, etc. Cosmetic products can also modify the shape of hair, either temporarily (using gel, mousse, etc.), or permanently (through permanent waving, hair straightening, etc.), creating a wide variety of artificial hairstyles.

The majority of virtual styling methods used today actually do not consider the physical structure of real hair in their algorithms. Rather than trying to match the *process* of real-world hair shape generation, most virtual styling methods try to match the final *results* with the appearance of real-world hair. Consequently, virtual styling techniques are not appropriate for applications that may desire a physically-correct model for the structure of hair, but rather for applications that desire a visually-plausible solution. However, there have been recent efforts towards the creation of styling methods that more accurately reflect the real-world process of hairstyle generation by considering what is known about real physical hair properties [6] and by mimicking more natural user interaction with hair [7]. Though promising, these endeavors are still at early stages.

B. Attaching Hair to the Scalp

Due to the high number of individual hair strands composing a human head of hair, it is extremely tedious to manually place each hair strand on the scalp. To simplify the process, a number of intuitive techniques have been developed that employ 2D or 3D placement of hairs onto the scalp.

1) 2D Placement: In some styling approaches, hair strands are not directly placed onto the surface of the head model. Instead, the user interactively paints hair locations on a 2D map which is subsequently projected onto the 3D model using a mapping function. Spherical mappings to map the strand bases to the 3D contour of the scalp have been popular approaches [2], [8].

Alternatively, Kim *et al.* [9] define a 2D parametric patch that the user wraps over the head model, as illustrated in Figure 1. The user can interactively specify each control point of the spline patch. In the 2D space defined by the two parametric coordinates of the patch, the user can place various clusters of hair.

Placing hair roots on a 2D geometry is easy for the user and allows flexibility. But mapping 2D hair roots onto a 3D curved scalp may cause distortion. Bando *et al.* [10] use a harmonic mapping and compensate for the mapping distortion by distributing the root particles based on a Poisson disc

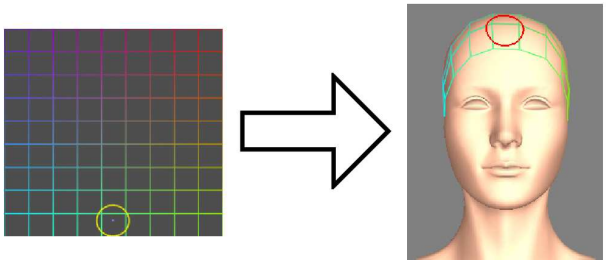

Fig. 1. 2D square patch wrapped onto the 3D model by the method of Kim *et al.* [9].

distribution using the distance between corresponding points on the scalp in world space rather than their 2D map positions.

2) 3D Placement: An alternative approach is to use direct 3D placement of the hair roots onto the scalp. Patrick *et al.* [11] present an interactive interface where the user can select triangles of the head model. The set of selected triangles defines the scalp, ie. the region of the head mesh where hair will be attached, and each triangle of the scalp is the initial section of a wisp.

3) Distribution of Hair Strands on the Scalp: A popular approach for placing hair strands uses uniform distribution over the scalp as it makes a good approximation of real hair distribution. Some wisp-based approaches randomly distribute hair roots inside each region of the scalp covered by the root wisps sections [12], [13], [14]. But if wisp sections overlap, a higher hair density is generated in the overlapping regions, which can produce distracting results. In order to guarantee a uniform hair distribution over the whole scalp, Kim *et al.* [9] uniformly distribute hair over the scalp and then assign each generated hair root to its owner cluster.

Some approaches also enable the user to paint local hair density over the scalp [15], [13]. Hair density can be visualized in 3D by representing density values as color levels. Controlling this parameter is helpful to produce further hairstyles such as thinning hair. Hernandez and Rudomin [15] extended the painting interface to control further hair characteristics such as length or curliness.

C. Global Hair Shape Generation

Once hair has been placed on the scalp, it has to be given a desired global shape which is commonly done through geometry-based, physically-based or image-based techniques, which are explained and evaluated in this section.

1) Geometry-Based Hairstyling: Geometric-based hairstyling approaches mostly rely on a parametric representation of hair in order to allow a user to interactively position groups of hair through an intuitive and easy-to-use interface. These parametric representations can involve surfaces to represent hair or wisps in the form of trigonal prisms or generalized cylinders.

a) Parametric Surface: Using two-dimensional surfaces to represent groups of strands has become a common approach to modeling hair [16], [17], [18]. Typically, these methods use a patch of a parametric surface, such as a NURBS surface, to reduce the number of geometric objects used to model a section of hair. This approach also helps accelerate hair simulation and rendering. These NURBS surfaces, often referred to as hair *strips*, are given a location on the scalp, an orientation, and weighting for knots to define a desired hair shape. Texture mapping and alpha mapping are then used to make the strip look more like strands of hair. A complete hairstyle can be created by specifying a few control curves or hair strands. The control points of these hair strands are then connected horizontally and vertically to create a strip. Though this method can be used for fast hairstyle generation and simulation, the types of hairstyles that can be modeled are limited due to the flat representation of the strip (see Figure 2, left).

In order to alleviate this flat appearance of hair, Liang and Huang [17] use three polygon meshes to warp a 2D strip into a U-shape, which gives more volume to the hair. In this method, each vertex of the 2D strip is projected onto the scalp and the vertex is then connected to its projection.

Fig. 2. Modeling hair using NURBS surfaces [16] (left). The Thin Shell Volume [19] (right)

Extra geometric detail can also be extracted from a surface representation. Kim and Neumann [19] developed a model called the *Thin Shell Volume*, or TSV, that creates a hairstyle starting from a parameterized surface. Thickness is added to the hair by offsetting the surface along its normal direction. Individual hair strands are then distributed inside the TSV (see Figure 2, right). Extra clumps of hair can be generated off a NURBS surface using the method of Noble and Tang [18]. Starting from a NURBS volume that has been shaped to a desired hairstyle, key hair curves are then generated along the isocurves of the NURBS volume. The profile curves that are extruded from the key hair curves create extra clumps, which can then be animated independently from the original NURBS surface. This approach adds more flexibility to the types of hair shapes and motions that can be captured using the surface approach.

b) Wisps and Generalized Cylinders: Wisps and generalized cylinders have been used as intuitive methods to control the positioning and shape of multiple hair strands in groups [14], [20], [21], [22], [13]. These methods reduce the amount of control parameters needed to define a hairstyle. A group of hair strands tend to rely on the positioning of one general space curve that serves as the center of a radius function defining the cross-section of a generalized cylinder, also referred to as a hair cluster. The cluster hair model is created from hair strands distributed inside of these generalized cylinders (see Figure 3). The user can then control the shape of the hair strands by

editing the positions of the general curve or curves.

Fig. 3. The cluster hair model [20] [21]

The clusters or wisps allow for the creation of many popular hairstyles from braids and twists of many African hairstyles [22] to constrained shapes such as ponytails. Some more complex hairstyles that do not rely on strands grouped into fixed sets of clusters are more difficult to achieve with these methods. Moreover, while they provide intuitive control to its users, the shaping of a hairstyle can often be tedious as the time to create a hairstyle is typically related to the complexity of the final style.

c) Multi-resolution Editing: Complex hair geometry can also be represented with a hierarchy of generalized cylinders [9], [23], allowing users to select a desired level of control in shape modeling. Higher level clusters provide efficient means for rapid global shape editing, while lower level cluster manipulation allows direct control of a detailed hair geometry – down to every hair strand. Kim and Neumann [9] further show that their multi-resolution method can generate complex hairstyles such as curly clusters with a copy-and-paste tool that transfers detailed local geometry of a cluster to other clusters (see Figure 4).

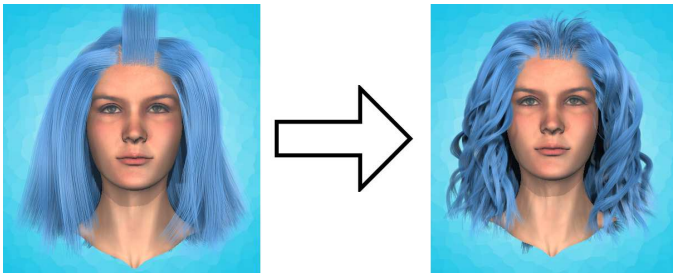

Fig. 4. Multiresolution hairstyling [9]

2) Physically-based Hairstyling: Some hairstyling techniques are strongly linked to physically-based animation of hair. These approaches rely on the specification of a few key parameters in methods ranging from cantilever beams that control individual strands to fluid flow methods that control the entire volume of hair. These methods customarily reduce the amount of direct user control over the resulting hairstyle.

a) The cantilever beam: In the field of material strengths, a cantilever beam is defined as a straight beam embedded in a fixed support at one end only. Anjyo *et al.* [3] consider that it is a similar case to a human hair strand, where the strand is anchored at the pore, and the other end is free. Considering gravity is the main source of bending, the method simulates the simplified statics of a cantilever beam to get the pose of one hair strand at rest. However, due to the use of a linear model, extra-forces need to be applied to the strand in order to get a proper final shape.

b) Fluid Flow: Hadap and Magnenat-Thalmann [24] modeled static hairstyles as streamlines of fluid flow based on the idea that static hair shapes resemble snapshots of fluid flow around obstacles. The user creates a hairstyle by placing streams, vortices and sources around the hair volume. For example, a vortex is used to create a curl in the hair at a desired location (see Figure 5).

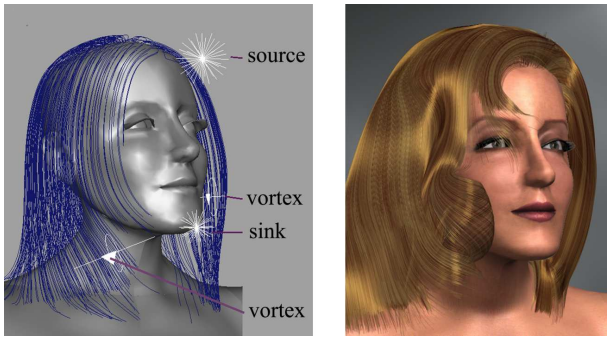

Fig. 5. Modeling hair using a fluid flow [24]

Hadap and Magnenat-Thalmann later extended this work to simulate dynamic hair, as explained in Section III-C.1.a.

c) Styling Vector and Motion Fields: Yu [8] observed that both vector fields and hair possess a clear orientation at specific points while both are also volumetric data; this led him to the use of static 3D vector fields to model hairstyles, see Figure 6 (left). Given a global field generated by superimposing procedurally defined vector field primitives, hair strands are extracted by tracing the field lines of the vector field. A hair strand begins at a designated location on the scalp and then grows by a certain step size along the direction of the accumulated vector of the vector field until a desired length is reached. Similarly particles can be used in motion fields to shape strands [25]. A particle is given a fixed life-time and traced through a motion field. The history of the particle comprises the whole hair strand; changing the life-time of the particle then changes the length of the hair.

Choe *et al.* [13] also use a vector field to compute global hair position while accounting for hair elasticity. Their algorithm calculates hair joint angles that best account for both the influence of the vector field and the natural trend of the strand for retrieving its rest position. Another important feature of the approach is the ability for the user to define *hair constraints*. A hair constraint causes a constraint vector field to be generated over a portion of 3D space that later modifies the original vector field proportionally to a weight parameter. Hair deformation is computed by using the previous algorithm applied on the modified vector field. In practice, the user can specify three types of constraints: point constraints, trajectory constraints and direction constraints. Hair constraints turn out to be very useful for creating complex hairstyles involving

ponytails, bunches or braids, as illustrated in Figure 6 (right).

Fig. 6. A styling vector field [8] (left) and constraint-based hairstyling [13] (right)

3) Generation of Hairstyles from Images: Recent hairstyle generation approaches have proposed an alternative way of generating hairstyles based on the automatic reconstruction of hair from images.

a) Hair Generation From Photographs: Kong *et al.* were the first who used real hair pictures to automatically create hairstyles [26]. Their method is merely geometric and consists of building a 3D hair volume from various viewpoints of the subject's hair. Hair strands are then generated inside this volume using a heuristic that does not ensure faithfulness in hair directionality. This approach is then best suited for simple hairstyles.

Grabli *et al.* introduced an approach exploiting hair illumination in order to capture hair local orientation from images [27]. Their system works by studying the reflectance of the subject's hair under various controlled lighting conditions. Fixing the viewpoint allows them to work with perfectly registered images. By considering a single viewpoint and using a single filter to determine the orientation of hair strands, the method reconstructs hair only partially. Paris *et al.* extended this approach [28] to a more accurate one, by considering various viewpoints as well as several oriented filters; their strategy mainly consists of testing several filters on a given 2D location and choosing the one that gives the most reliable results for that location. This method captures local orientations of the visible part of hair, and thus produces visually faithful results with respect to original hairstyles (see Figure 7). Wei *et al.* [29] subsequently improved the flexibility of the method by exploiting the geometry constraints inherent to multiple viewpoints, which proves sufficient to retrieve a hair model with no need for controlled lighting conditions nor a complex setup.

Fig. 7. Hair capture from photographs [28]

b) Hair Generation From Sketches: Mao *et al.* [30] developed a sketch-based system dedicated to modeling cartoon hairstyles. Given a 3D head model, the user interactively draws the boundary region on the scalp where hair should be placed. The user then draws a silhouette of the target hairstyle around the front view of the head. The system generates a silhouette surface representing the boundary of the hairstyle. Curves representing clusters of hair are generated between the silhouette surface and the scalp. These curves become the spine for polygon strips that represent large portions of hair, similar to the strips used by [16], [17].

This sketch-based system quickly creates a cartoon hairstyle with minimal input from its user. The strips, or cluster polygons, used to represent the hair, however, are not appropriate for modeling more intricate hairstyles such as those observable in the real world.

4) Evaluation: Each of the global hair shaping methods described in this section is appropriate for styling hair under different circumstances. Table I shows a comparison of several global shaping methods in hair shape flexibility, user control, and time for manual setup or input. The larger the range of hair shapes that can be modeled by an algorithm, the broader its applicability in practice is. The level of user control is important in order to facilitate placing exact details where desired in the hair. Moreover, while some styling methods can capture a hair shape quickly through automatic processing, others require time-consuming manual setup or input by its user.

As Table I indicates, geometry-based hairstyling techniques, such as through generalized cylinders or parametric surfaces, customarily give the user a large degree of control over the hair; however, the manual positioning of hair can be a tedious, time-consuming task due to the large intricate volume of hair. The time for a user to create a hairstyle using the multiresolution generalized cylinder approach presented by Kim and Neumman [9] ranged between several minutes to several hours depending on the complexity of the hair shape. While parametric surfaces typically provide fast methods for hairstyle creation, the results tend to be limited to flat, straight hairstyles due to the 2D surface representation. Alternatively, wisp or generalized cylinders can model many straight or curly hairstyle shapes.

Controlling the volume of the hair through physically-based techniques, such as through fluid flow or vector fields, typically requires less tedious input by the user; however, finer details of many complex hairstyles are often difficult to capture through such interaction. Many of the parameters can be non-intuitive to hairstyling and the user typically has less specific control over the hairstyle creation in comparison to the geometry-based approaches.

The generation of hairstyles from images has been shown to be a highly automatic process even with a relatively simple setup by Wei *et al.* [29]. The final hairstyles created from images can be quite impressive, but these methods are limited in that they result from hairstyles that have to exist in the real world, making the range of styles modeled generally less flexible than geometric or physically-based methods. Hairstyles generated from sketches can allow for more creativity in

the resulting hair shapes, though specific finer details, such as with braided hair, can be impossible to achieve without cumbersome user involvement.

	Hair Shapes	User Control	Manual Time
Gen. Cylinders	flexible	high	slow
Surfaces	limited to straight	high	fast
Physical Volumes	limited, details hard	cumbersome	medium
Photos	limited, must exist	none	fast
Sketches	limited, details hard	medium	fast

TABLE I
ANALYSIS OF GLOBAL SHAPING METHODS *Evaluation of geometry-based generalized cylinders and surfaces, physically-based volumes and image-based using photographs and sketches in the areas of user control, flexibility of resulting hair shapes, and the time of manual input or setup.*

There are recent techniques that build on the strengths of different methods. For example, the work by Choe *et al.* [13] model hair in the form of wisps where the user edits the prototype strand that controls the wisp shape, but vector fields and hair constraints are also utilized to achieve intricate hair shapes such as braids, buns, and ponytails. While exact timings for manual input is not provided, the amount of user input is still considered high and the most time-consuming aspect of the whole virtual hairstyling process.

D. Managing Finer Hair Properties

After hair has been given a global shape, it is often desirable to alter some of the finer, more localized properties of the hair to either create a more realistic appearance (e.g. curls or volume) or to capture additional features of hair such as the effects of water or styling products. In practice, most of these techniques to control finer details have been used in conjunction with geometric or physically-based approaches for defining a global hair shape (Sections II-C.1 and II-C.2).

1) Details of Curls and Waves: Local details such as curls, waves or noise might need to be added to achieve a natural appearance for hair once a global shape has been defined. Yu [8] generates different kinds of hair curliness by using a class of trigonometric offset functions. Various hairstyles can thus be created by controlling different geometric parameters such as the magnitude, the frequency or the phase of the offset function. In order to prevent hair from looking too uniform, offset parameters are combined with random terms that vary from one hair cluster to another (see Figure 8, left). Similarly, a more natural look can be generated for hair shaped through fluid flow by incorporating a breakaway behavior to individual hair strands that allow the strand to breakaway from the fluid flow based on a probability function [24].

Choe *et al.* [13] model a hairstyle with several wisps, and the global shape of each wisp is determined by the shape of a *master strand*. Within a wisp, the degree of similarity among the strands is controlled by a length distribution, a deviation radius function and a fuzziness value. The geometry of the master strand is decomposed into an outline component and a details component. The details component is built from a *prototype* strand using a Markov chain process where the degree of similarity between the master strand and the prototype strand can be controlled through a Gibbs distribution. Resulting hairstyles are thus globally consistent while containing fine variations that greatly contribute to their realism, as shown by Figure 8 (right).

Fig. 8. Waves and curls procedurally generated by Yu [8] (left) and Choe *et al.* [13] (right)

These methods for localized shape variation help to alleviate the synthetic look of the virtual hair, however since most of them incorporate some form of random generation the user has less control over the finer details. This semi-automatic process helps accelerate the creation of hairstyles as these minute details could take many man-hours if performed manually. On the other hand, the random generation can also cause unwanted artifacts if strands are perturbed in a way that causes unnatural collisions. Moreover, these methods do not account for the physical hair properties for computing the hair geometry, although it is well-known that such features, described in Section II-A, have a great influence on the hair shape [4], [5].

In order to automatically generate the fine geometry, including waves or curls, of natural hair, Bertails *et al.* [6] recently introduced a new hairstyling method using a mechanically accurate model for static elastic rods (the Kirchhoff model). The method, based upon a potential energy minimization, accounts for the natural curliness of hair, as well as for the ellipticity of hair fibers' cross-section (see Figure 9). Though not appropriate for creating complex hairstyles, this method is promising for a more accurate hairstyle generation process, accounting for individual hair fiber properties. It could thus be useful for cosmetics prototyping. Very recently, this approach was extended to hair dynamics (see Section III-B.4)

Fig. 9. A real ringlet (left), and a synthetic one (right) automatically generated by the physically-based method of Bertails *et al.* [6]

2) Producing Hair Volume: Whereas most geometric-based hairstyling methods implicitly give volume to hair by using

volumetric primitives (see Section II-C.1), physically-based methods often account for hair self-collisions in order to produce volumetric hairstyles. Approaches that view hair as a continuous medium [25], [24], [8], [13] add volume to the hair through the use of continuum properties that reproduce the effects of collisions between hair strands, such as via vector fields or fluid dynamics. As strands of hair become closer, these techniques either prevent them from intersecting due to the layout of the vector or motion fields, or foster a repulsive motion to move them apart from each other.

Since detecting collisions between strands of hair can be difficult and, in the least, very time consuming, Lee and Ko [31] developed a technique that adds volume to a hairstyle without locating specific intersections among strands. The idea is that hair strands with pores at higher latitudes on the head cover strands with lower pores. Multiple head hull layers are created of different sizes from the original head geometry. A hair strand is checked against a specific hull based on the location of its pore. A hair-head collision detection and response algorithm is then used. This method only works in the case of a quasi-static head that remains vertically oriented.

3) Modeling Styling Products and Water Effects: Styling products, such as hairspray, mousse, and gel, have significant effects on the hair appearance, including hairstyle recovery after the hair has moved, stiffened overall hair motion, large grouping of hair strands due to the adhesiveness of fixative products, and the change in the hair volume.

Lee and Ko [31] developed a method to model the effects of hair gel on a hairstyle. A styling force is used to enable hairstyle recovery as the hair moves due to external force or head movement. As a result, an initial hairstyle can be restored after motion. When gel is applied to the hair, the desire is to retain the deformed hairstyle rather than returning to the initial style. This algorithm preserves the deformed shape by updating the styling force during the simulation. Alternatively, breakable *static links* or *dynamic bonds* can be used to capture hairstyle recovery by applying extra spring forces between nearby sections of hair to mimic the extra clumping of hair created by styling products [32], [33].

Styling products also increase the stiffness of hair motion allowing a curled section of hair with styling products applied to retain a tight curl as the hair moves. Through the use of a *dual-skeleton* model for simulating hair, separate spring forces can be used to control the bending of hair strands versus the stretching of curls [33]. Styling products can then alter the spring stiffness' independently to create desired results.

Water will also drastically change the appearance, shape and motion of hair. As water is absorbed into hair the mass of the hair increases up to 45%, while its elasticity modulus decreases by a factor of 10 – leading to a more deformable and less elastic material [34]. Moreover, as hair gets wet, the volume of the hair decreases because strands of hair in close proximity with each other adhere due to the bonding nature of water. In their static physically-based model, Bertails *et al.* [6] easily incorporated the effect of water on hair by simply modifying the relevant physical parameters that actually change when hair gets wet: the mass and the Young's modulus of each fiber. Ward *et al.* [33] modeled dynamic wet hair with their dual-skeleton system by automatically adjusting the mass of the hair along the skeletons as water is added to the hair. The increased mass resulted in limited overall motion of the hair and elongation of curls. A flexible geometric structure allows the volume of the hair to change dynamically by altering the radius of the strand groups that are used in simulation (see Figure 10).

Fig. 10. Comparison of hair (a) dry and (b) wet [33].

An interactive virtual hairstyling system introduced by Ward *et al.* [7] illustrates how water and styling products can be used to interactively alter the look and behavior of hair dynamically through a 3D interface that allows users to perform common hair salon applications (such as wetting, cutting, blow-drying) for the purpose of intuitively creating a final hairstyle.

III. Hair Simulation

It is difficult to provide a realistic model for dynamic hair because each individual hair strand has a complex mechanical behavior and very little knowledge is available regarding the nature of mutual hair interactions. Animation of a full head of hair raises obvious problems in terms of computational costs. As a consequence, existing hair animation methods propose a tradeoff between realism and efficiency, depending on the intended application. Before analyzing existing methods on hair animation, we briefly describe in Section III-A some mechanical features of real hair.

Numerous methods for the dynamics of an individual hair strand have been borrowed from existing 1D mechanical models. These models, presented and commented on in Section III-B, have subsequently been used for animating both individual hair strands and groups of hair. While there can be over 100,000 strands of hair on a human head, it was observed that most hair strands tend to move in a similar way as their neighbors. This observation led to a number of approaches extending the single-strand method to simulate the collective behavior of hair. These methods, which range from continuum to hair wisp models, will be presented in Section III-C. Finally, Section III-D presents recent works that have used multi-resolution techniques in order to gain efficiency and to achieve interactive hair simulations.

A. The Mechanics of Hair

A hair strand is an anisotropic deformable object: it can easily bend and sometimes twist but it strongly resists shearing

and stretching. A hair strand also has elastic properties in the sense that it tends to recover its original shape after the stress being applied to it has been removed. The nature of interactions between hair strands is very complex. This is largely due to the surface of individual hair strands, which is not smooth but composed of tilted scales (see Figure 11). This irregular surface causes anisotropic friction inside hair, with an amplitude that strongly depends on the orientation of the scales and of the direction of motion [35]. Moreover, hair is very triboelectric, meaning it can easily release static charges by mere friction. This phenomenon has been measured in the case of combed hair, but it seems that no study has been published regarding this effect in the case of hair-hair friction.

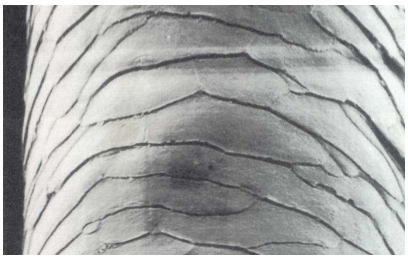

Fig. 11. An electron micrograph of a hair fiber that shows the structure of the outer cuticle surface, which is composed of thin overlapping scales [4].

Also, the geometric hair shape, which is correlated to some structural and physical features of the hair (see Section II-A) affects the motion of hair. For example, a curly moving hair will look more "elastic" than a straight hair, because hair curls can longitudinally stretch during motion, like springs–although hair strands still remain unstretchable. In addition, hair clumps are more likely to appear in curly hair, where contacts exist among hair strands, and thus the probability for them to get into tangles is greater. In fact, as observed in the real world, the more intricate the hair's geometry is, the less degrees of freedom it has during motion. The feature *curliness* will be evaluated on existing approaches in Sections III-B and III-C. To our knowledge, there are no quantitative nor qualitative results published on hair grouping from the mechanics literature.

Unlike some well-known physical materials such as fluids–which have been studied for centuries and modeled by accurate equations–hair remains an unsolved problem for which there is currently no standard physically-based model. Hence, one of the challenges lies in finding an appropriate representation of hair for dynamic simulation.

B. Dynamics of Individual Hair Strands

Within the 20 past years, three families of computational models have been proposed and used for simulating the dynamics of one individual hair strand: mass-spring systems, projective dynamics, and rigid multi-body serial chains. Very recently, some existing work on static Kirchhoff rods [36], [6] has been extended to hair dynamics, leading to a new model called *dynamic Super-Helices*.

Each one of these four models is described and evaluated in terms of realism and ability to be included inside a full hair.

1) Mass-Spring Systems: One of the first attempts to animate individual hair strands was presented by Rosenblum *et al.* [2] in 1991. A single hair strand is modeled as a set of particles connected with stiff springs and hinges. Each particle has three degrees of freedom, namely one translation and two angular rotations. Hair bending rigidity is ensured by angular springs at each joint. This method is simple and easy to implement. However, torsional rigidity and non-stretching of the strand are not accounted for. Limiting the stretching of the strand requires the use of strong spring forces, which leads to stiff equations that often cause numerical instability, unless very small time steps are used.

Many advances in mass-spring formulation were recently made, especially in the context of cloth simulation. Baraff and Witkin [37] showed that implicit integration methods prove very useful for the simulation of a stiff system as they ensure that the system will remain stable even with large time steps. Implicit integration was later used in the case of hair simulation [38], [39]. Other approaches [12], [40] used a constrained mass-spring model, well-suited for animating extensible wisps such as wavy or curly wisps.

2) One Dimensional Projective Equations: In 1992, Anjyo *et al.* proposed a simple method based on one-dimensional projective differential equations for simulating the dynamics of individual hair strands. Initially, the statics of a *cantilever beam* is simulated to get an initial plausible configuration of each hair strand. Then, each hair strand is considered as a chain of rigid sticks s_i. Hair motion is simulated as follows:

- Each stick s_i is assimilated as a direction, and thus can be parameterized by its polar angles ϕ (azimuth) and θ (zenith) (see Figure 12).
- The external force $\mathbf{F}$ applied to the stick is projected onto both planes P_ϕ and P_θ, respectively, defined by ϕ and θ (the longitudinal projection of $\mathbf{F}$ on s_i is neglected since it should have no effect on the rigid stick).
- Fundamental principles of dynamics are applied to each parameter ϕ and θ which leads to two differential equations that are solved at each time step.

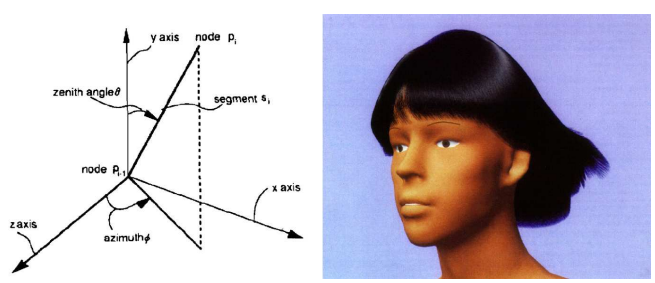

Fig. 12. Left: the polar coordinate system for a hair segment. Right: simulating individual hair strands using one dimensional projective equations for dynamics [3].

This method is attractive for many reasons. It is easy to implement, efficient (tens of thousands of hair strands can efficiently be simulated this way). Moreover, hair is prevented from stretching while hair bending is properly recovered.However, as torsional hair stiffness cannot be accounted for, this method cannot properly handle fully 3D hair motions.

Furthermore, as motion is processed from top to bottom, it is difficult to handle external punctual forces properly. Issues related to the handling of external forces are discussed in Section III-B.5.

3) Rigid multi-body serial chain: In order to compute the motion of individual hair strands, forward kinematics have been used as a more general alternative to one-dimensional projective equations [41], [32]. Such techniques are well-known in the field of robotics, and efficient multi-body dynamics algorithms have been proposed for decades [42].

Each hair strand can be represented as a serial, rigid, multi-body open chain using the reduced or spatial coordinates formulation [42], in order to keep only the bending and twisting degrees of freedom of the chain: stretching DOFs are removed (see Figure 13). Apart from the gravitational influence, forces accounting for the bending and the torsional rigidity of the hair strand are applied on each link. Forward dynamics are processed using the Articulated-Body Method described in [42], with a linear time complexity. Hadap and Magnenat-Thalmann [41] and Chang *et al.* [32] used this technique to animate several sparse individual hair strands within an elaborate, global continuous model for hair (see Section III-C.1). Results for these methods have typically been limited to straight hair as possible issues related to curly hair simulation are not explained.

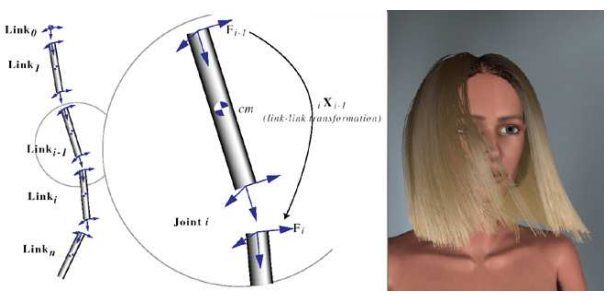

Fig. 13. (left) Hair strand as a rigid multi-body serial chain [41] (right) Simulation of hair blowing in the wind using fluid flow.

4) Dynamic Super-Helices: The Kirchhoff's theory for elastic rods has just been exploited by Bertails *et al.* for accurately predicting the motion of individual hair fibers [43]. The resulting mechanical model for one individual hair strand, called a *Super-Helix*, corresponds to a spatial discretization of the original continuous Kirchhoff model, where the curvatures and the twist of the rod are assumed to remain constant over each predefined piece of the rod. As a result, the shape of the hair strand is a piecewise helix, with a finite number of degrees of freedom. This model is then animated using the principles of Lagrangian mechanics. The super-Helix model naturally accounts for the typical nonlinear behavior of hair, as well as for its bending and twisting deformation modes. It also intrinsequely incorporates the constraint of inextensibility. Finally, unlike all previous models, hair natural curliness is properly handled through this model, making it possible to accurately simulate the dynamics of curls.

5) Handling external forces: A good dynamics model for one individual hair strand is expected to yield realistic motion of one isolated hair strand, but it should also be able to properly handle any external forces, such as gravity, wind, contacts or collisions.

In methods simulating chains of rigid links (Section III-B.2), motion is processed from top to bottom in one single pass. This means that a collision detected at stick s_k only affects the following sticks s_j, where $j > k$ without propagating the effect backward to the sticks located near the roots, which can lead to unrealistic shapes for hair. Refining Anjyos's method, Lee and Ko [31] simply fix the problem by adding an extra force that enables hair to get a proper shape when colliding with an object other than the head.

In the case of serial rigid multi-body chains, external forces can properly be accounted for when using a multi-pass forward dynamics algorithm. However, because of the high bending and torsional stiffness that are required to maintain the curved shape at rest, the simulation may lack stability if external forces are integrated within an explicit integration scheme. The major drawback of the articulated bodies model is that, unlike the mass-springs model, it is difficult to formulate the conventional dynamics–based on reduced coordinates–using an implicit integration scheme [44]. As proposed by Baraff [45], a solution may be the use of Lagrange multipliers instead of a reduced-coordinates formulation of the problem, in order to integrate hard constraints implicitly. Probably because of its non triviality, this method has never been implemented in the case of hair dynamics. Choe *et al.* recently proposed another solution based on a hybrid model, which takes advantage of both mass-springs models and rigid multi-body serial chains [39]. This model allows for an implicit (and thus stable) integration of the dynamics, including robust constraints between the hair and the body. But as mass-springs, it does not fully avoid stretching of the hair strand.

Finally, the Super-Helix model properly handles soft constraints such as external forces. However, as the model is parameterized by reduced coordinates, accounting for hard constraints may be tricky.

6) Evaluation: The following table indicates, for each hair dynamic model given above, which are the required properties that it ensures.

	Mass-springs	**Projective dynamics**	**Rigid multi-body serial chain**	**Dynamic Super-Helices**
Bending	yes	yes	yes	yes
Torsion	no	no	yes	yes
Non-stretching	no	yes	yes	yes
Curliness	no	no	no	yes
Constraints	easy	tricky	tricky for hard	tricky for hard

TABLE II

ANALYSIS OF DYNAMIC MODELS FOR INDIVIDUAL HAIR STRANDS

Criteria: bending rigidity, torsional rigidity, non-stretching, curliness handling, and handling external constraints (soft and hard) properly.

C. Simulating the Dynamics of a Full Hairstyle

Handling a collection of hair strands leads to additional challenges in the field of computer graphics: the realism of

the collective dynamic behavior and the efficiency of the simulation.

As mentioned in Section III-A, hair interactions are very complex, and little knowledge is known about the actual phenomena of interactions, and their order of magnitude. In addition, the enormous number of contacts and collisions that occur permanently *or* temporarily inside hair raises obvious problems in terms of computational treatment. Consequently, two challenging issues have to be handled when computing hair contacts and collisions: detection and response.

While early hair animation methods generally neglected hair-hair interactions for the sake of efficiency and simplicity, more recent approaches for animating hair make assumptions on hair consistency during motion to simplify the problem of collisions. Hair is essentially either globally considered as a continuous medium (Section III-C.1), or as a set of disjoint groups of hair strands (Section III-C.2.b). Specific hair-hair interaction models are proposed in both cases.

1) Hair as a Continuous Medium: Due to the high number of strands composing a human head of hair, simulating each strand individually is computationally overwhelming. Furthermore, strands of hair in close proximity with each other tend to move similarly. This observation led researchers to view hair as an anisotropic continuous medium.

a) Animating Hair using Fluid Dynamics: Considering hair as a continuum led Hadap and Magnenat-Thalmann [41] to model the complex interactions of hair using fluid dynamics. The interactions of single hair strands are dealt with in a global manner through the continuum.

Individual strand dynamics is computed to capture geometry and stiffness of each hair strand (see Section III-B.3). Interaction dynamics, including hair-hair, hair-body, and hair-air interactions, are modeled using fluid dynamics. Individual hair strands are kinematically linked to fluid particles in their vicinity. In this model, the density of the hair medium is defined as the mass of hair per unit of volume occupied. The pressure and viscosity represent all of the forces due to interactions to hair strands.

Using this setup, it is possible to model hair-body interactions by creating boundary fluid particles around solid objects (see Figure 13 which shows hair blowing in the wind). A fluid particle, or *Smooth Particle Hydrodynamics* (SPH), then exerts a force on the neighboring fluid particles based on its normal direction. The viscous pressure of the fluid, which is dependent on the hair density, accounts for the frictional interactions between hair strands.

Utilizing fluid dynamics to model hair captures the complex interactions of hair strands. However, since this method makes the assumption of a continuum for hair, it does not capture dynamic clustering effects that can be observed in long, thick real hair. Moreover, computations required for this method are quite expensive; using parallelization, it took several minutes per frame to simulate a hair model composed of 10,000 individual hair strands.

b) Loosely Connected Particles: Bando *et al.* [10] have modeled hair using a set of SPH particles that interact in an adaptive way. Each particle represents a certain amount of hair material which has a local orientation (the orientation of a particle being the mean orientation of every hair strand covered by the particle), refer to Figure 14.

Fig. 14. (left) Particles defining hair, line segments indicate direction (right) Animation of hair with head shaking [10].

Initially, connected chains are settled between neighboring particles being aligned with local hair orientation: two neighboring particles having similar directions and being aligned with this direction are linked. This initial configuration is kept during the motion because it represents the spatial consistency of interactions between particles. During motion, each particle can interact with other particles belonging to its current neighborhood. The method proposes to handle these interactions by settling breakable links between close particles; as soon as the two particles are not close enough, these links vanish. Thus, this method facilitates transversal separation and grouping while maintaining a constant length for hair. At each time step, searching the neighborhood of each particle is done efficiently by using a grid of voxels.

c) Interpolation between Guide Hair Strands: Chang *et al.* [32] created a system to capture the complex interactions that occur among hair strands. In this work, a sparse hair model of *guide strands*, which were first introduced in [46], [47], is simulated. A dense hair model is created by interpolating the position of the remaining strands from the sparse set of guide strands. Using multiple guide hair strands for the interpolation of a strand alleviates local clustering of strands.

The sparse set of guide strands is also used to detect and handle mutual hair interactions. Since detecting collisions only among the guide strands is inefficient, an auxiliary triangle strip is built between two guide strands by connecting corresponding vertices of the guide strands (see Figure 15). A collision among hair strands is detected by checking for intersections between two hair segments and between a hair vertex and a triangular face. Dampened spring forces are then used to push a pair of elements away from each other when a collision occurs. Figure 15 shows the sparse and dense hair models, respectively.

The use of guide strands can lead to missed collisions when the interpolated strands collide with an object with which the guide strands do not.

d) Free Form Deformation: To achieve hair simulation of complex hairstyles in real-time, Volino *et al.* [48] proposed to use a global volumetric free form deformation (FFD) scheme instead of considering an accurate mechanical model related to the structure of individual hair strands. A mechanical model is defined for a lattice surrounding the head. The lattice is then deformed as a particle system and hair strands follow the deformation by interpolation. Collisions between the hair and the body are handled by approximating the body as a set of

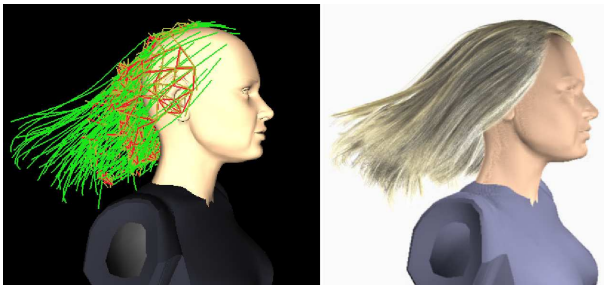

Fig. 15. (left) Sparse hair model with static links and (right) Rendered image of interpolated dense hair model [32].

metaballs.

This method is well-suited for animating various complex hairstyles, when the head motion has a low magnitude. For high deformations, hair discontinuities observed in real hair (e.g., see Figure 18) would not be reproduced because only continuous deformations of hair are considered through the lattice deformation.

2) Hair as Disjoint Groups: In order to reduce the complexity of hair, an alternative approach consists of grouping nearby hair strands and simulating these disjoint groups as independent, interacting entities. This representation of hair was especially used to save computation time in comparison with the simulation of individual strands, and even reach interactive frame rates. It also captures realistic features of hair as it accounts for local discontinuities observed inside long hair during fast motion; these local discontinuities cannot be captured using the continuum paradigm.

a) Real-time Simulation of Hair Strips: As discussed in Section II-C.1.a, the complexity of hair simulation has been simplified by modeling groups of strands using a thin flat patch, referred to as a *strip* (see Figure 16) [16], [19], [49], [50], [17], [51], [52]. A simple dynamics model for simulating strips is presented in [49] that is adapted from the projective angular dynamics method introduced by Anjyo *et al.* [3] (see Section III-B.2); dynamics is applied to the control point mesh of the NURBS surface.

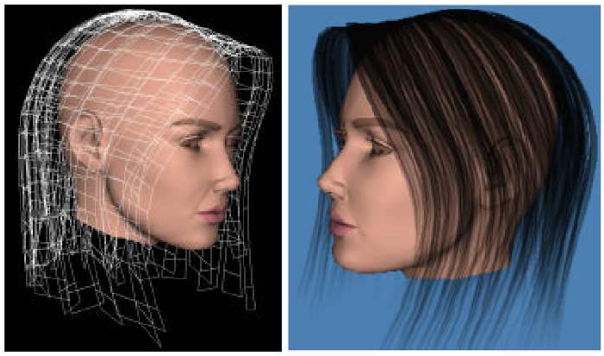

Fig. 16. Hair strips as an approximate hair model [16].

Using strips to model hair results in significantly faster simulation because fewer control points are required to model a strip in comparison to modeling individual strands. In [49], collision avoidance between hair strips and external objects, such as the head or the body, is achieved by using ellipsoids to approximate the boundaries of these objects. When a control point of the strip is inside the ellipsoid a reaction constraint method is used to move it back to the boundary. Furthermore, collisions between hair strips are avoided by introducing springs within the strips and between neighboring strips. The springs are used to prevent neighboring strips from moving too far apart or too close together. Moreover, springs are also used to prevent a strip from overstretching or over-compressing. The result is that the hairstyle remains relatively consistent throughout the simulation. Similarly, Guang and Zhiyong [50] presents a strip-based hair structure for modeling short hair where the strips of texture-mapped hair are simulated using a mass-spring model and 3D morphing.

By using a single strip to represent tens or hundreds of hair strands, hair simulation, including hair-hair collision avoidance, can be achieved in real-time. This process, however, is limited in the types of hairstyles and hair motions it can represent; the flat shape of the strips is most suited to simulating simple, straight hair.

b) Simulation of Wisps: One of the first methods to take advantage of grouping hair was presented by Watanabe and Suenaga in [53]. They animate a set of trigonal prism-based wisps. During motion, the shape of a wisp is approximated by *parabolic trajectories* of fictive particles initially located near the root of each wisp. At each time step, the trajectories of the particles are estimated using initial velocities and accelerations, such as gravitational acceleration. This method amounts to simulating only approximate kinematics without considering the inertia of the system, which appears to be limited to slow hair motion. Moreover, interactions between different wisps are not taken into account.

A similar process of grouping neighboring strands together into wisps was used by [47], [46]. In these works, a wisp of strands is modeled by simulating the motion of a single *typical* strand and then generating other strands by adding random displacements to the origin of the typical strand. The number of overall strands that need to be simulated is reduced significantly. Again, in this work, interactions among strands, or between wisps, is not considered.

To account for complex interactions being observed in real hair during fast motion, Plante *et al.* [12], [54] have represented hair using a fixed set of deformable, volumetric wisps. Each wisp is structured into three hierarchical layers: a *skeleton curve* that defines its large-scale motion and deformation, a deformable volumetric envelope that coats the skeleton and accounts for the deformation of the wisp sections around it, and a given number of hair strands that are distributed inside the wisp envelope and used only at the rendering stage of the process (see Figure 17).

As the skeleton approximates the average curve of a wisp, it is likely to stretch or compress a bit while the wisp is not completely straight. The mass-spring simulation can thus be well-suited for simulating wavy or curly wisps.

Assuming that the local discontinuities inside hair are caused by collisions between wisps of different orientations, this method provides a model of anisotropic interactions

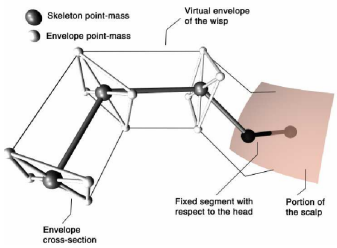

Fig. 17. Elements defining a deformable volumetric wisp [12].

between wisps. Wisps of similar orientations are allowed to penetrate each other, and are submitted to viscous friction, whereas wisps of different orientations actually collide in a very dissipative way.

Fig. 18. The layered wisp model [12] (bottom) captures both continuities and discontinuities observed in real long hair motion (top).

As illustrated in Figure 18, the approach has led to convincing results for fast motions, capturing the discontinuities that can be observed in long, thick hair. Nevertheless, very expensive computations were required for the examples shown, which was mainly due to the high cost for detecting collisions between the deformable wisps. Moreover, the high number of contacts that needed to be computed between each wisp at rest caused some visible artefacts in the rest state.

Choe *et al.* [39] have recently improved the stability of this kind of approaches. Collisions between the wisps and the body are robustly handled by using constrained dynamics. Moreover, to avoid undesired oscillations when computing wisp-wisp interactions, they propose an empiric law for controlling the amplitude of penalty forces. A cohesive force is also used to preserve the initial hairstyle during the simulation.

D. Multi-resolution Methods

Recently, researchers have begun to explore adaptive representations for hair. These methods can be used to alleviate unnatural clumping of hair strands that can be common in other approaches or to accelerate simulation while preserving realistic features in hair motion.

1) Level-of-Detail Representations: To better capture natural clustering of hair, a multi-resolution hair modeling scheme may be used to accelerate both the simulation and rendering of hair while maintaining a high visual quality. Ward *et al.* [55], [38] use three different levels of detail (LODs) for modeling hair – *individual strands*, *clusters* and *strips* represented by subdivision curves, subdivision swept volumes, and subdivision patches, respectively (see Figure 19, left). By creating a hair hierarchy composed of these three discrete LODs along with an efficient collision detection method that uses the family of *swept sphere volumes* (SSVs) [56] as bounding volumes to encapsulate the hair, this method was able to accelerate hair simulation up to two orders of magnitude.

During simulation, the hair hierarchy is traversed to choose the appropriate representation and resolution of a given section of hair. The transitions between the LODs occur automatically using a higher resolution simulation for the sections of hair that are most significant to the application based on the hair's visibility, viewing distance, and motion, relative to the viewer. If an object in the scene occludes a section of hair or if the hair is outside of the field-of-view of the camera then the section is simulated at the coarsest LOD (a strip) and is not rendered. If the hair can be viewed, the distance of the viewer from the hair and its motion determine its current resolution. As the distance from the hair to the viewer decreases, or as the hair moves more drastically, there is more observable detail and a need for a more detailed simulation within the hair, thus the hair is simulated and rendered at higher resolutions. Figure 19 shows LOD representations (left) used for simulating hair blowing in the wind (right).

Recently, Ward *et al.* [7] introduced a simulation-localization technique that provided additional performance improvements by quickly finding areas of high activity. Coupled with LOD representations, this method both simulated and rendered hair fast enough for a user to interact with dynamic hair.

Fig. 19. Left: Level-of-detail representations for hair (a) strip (b) cluster (c) strand. Right : Curly, long hair blowing in the wind using LOD representations [55].

2) Adaptive Clustering: In order to *continuously* adjust the amount of computations according to the local complexity of motion, techinques for adaptive clustering and subdivision of simulated hair have been proposed recently [40], [38]. Bertails *et al.* [40] introduced an adaptive animation control structure, called *Adaptive Wisp Tree* (AWT), that enables the dynamic splitting and merging of hair clusters. The AWT depends on a complete hierachical structure for the hair, which can either be precomputed–for instance using a hierarchical hairstyle [9]–or computed on the fly. The AWT represents at each time step the wisps segments of the hierarchy that are actually simulated (called *active* segments). Considering that hair should always be more refined near the tips than near

the roots, the AWT dynamically splits or merges hair wisps while always preserving a tree-like structure, in which the root coincides with the hair roots and the leaves stand for the hair tips.

At each time step, different wisps segments of the global hierarchy, that is, different LOD, can thus be active, while only the finest levels of details are used at the rendering stage. The splitting process locally refines the hair structure when a given wisp segment is not sufficient for capturing the local motion and deformation. The merging process simplifies the AWT when the motion becomes coherent again. Splitting and merging criteria are linked to the local motion of hair (for example, the magnitude of velocity of the wisps segments) at each time step.

One of the key benefits of the AWT is that it implicitly models mutual hair interactions so that neighboring wisps with similar motions merge, mimicking the static friction in real hair. This avoids subsequent collision processing between these wisps, thus increasing efficiency as well as gaining stability from the reduced number of primitives. In addition, the splitting behavior models wisps deformation without the need of the complex deformable wisp geometry used in [12]. For collision processing, active wisp segments of the AWT are thus represented by cylinders, which greatly simplifies collision detection tests.

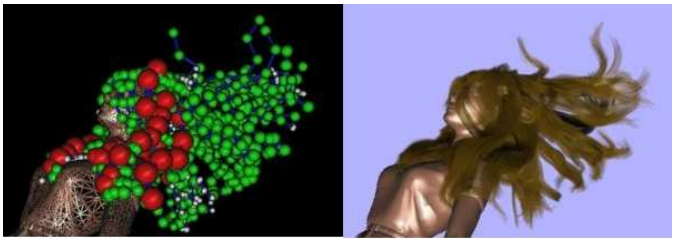

Fig. 20. Illustration of the AWT on long hair (left) and its final rendered version (right) [40].

Ward and Lin [38] proposed a similar, but a more top-down approach, for animating hair. Their continuous multi-resolution structure, called *hair hiearchy* [38], is coupled with the level-of-detail representations [55], instead of wisps [12].

IV. Hair Rendering

Realistic rendering of human hair requires the handling of both local and global hair properties. To render a full hairstyle, it is necessary to choose an appropriate global representation for hair. Implicit and explicit representations are presented and discussed in Section IV-A. Local hair properties define the way individual hair fibers are illuminated. Section IV-B describes the scattering properties of hair and reviews the different models that have been proposed to account for those properties. Global hair properties also include the way hair fibers cast shadows on each other; this issue of self-shadowing, handled in Section IV-C, plays a crucial role in volumetric hair appearance. Rendering hair typically requires time-consuming computations, Section IV-D reviews various rendering acceleration techniques.

A. Representing Hair for Rendering

Choices of hair rendering algorithms largely depend on the underlying representations for modeling hair geometry. For example, explicit models require line or triangle-based renderers, whereas volumetric models need volume renderers, or rendering algorithms that work on implicit geometry.

1) Explicit Representation: With an explicit representation, one has to draw each hair fiber. A hair fiber is naturally represented with a curved cylinder. The early work by Watanabe and Suenaga [53] adopted a trigonal prism representation, where each hair strand is represented as connected prisms with three sides. This method assumes that variation in color along the hair radius can be well approximated by a single color. Others use ribbon-like connected triangle strips to represent hair, where each triangle always faces towards the camera. Ivan Neulander [57] introduced a technique that adaptively tessellates a curved hair geometry into polygons depending on the distance to the camera, curvature of hair geometry, etc. At large distances, a hair strand often resembles many hairs. Kong and Nakajima [58] exploited this property to reduce the number of rendered hairs by adaptively creating more hairs at the boundary.

Difficulties arise with explicit rendering of tesselated hair geometry due to the unique nature of hair – a hair strand is extremely thin in diameter (0.1 mm). In a normal viewing condition, the projected thickness of a hair strand is much smaller than the size of a pixel. This property causes severe undersampling problems for rendering algorithms for polygonal geometry. Any point sample-based renderer determines a pixel's color (or depth) by a limited number of discrete samples. Undersampling creates abrupt changes in color or noisy edges around the hair. Increasing the number of samples alleviates the problem, but only at slow convergence rates [59] and consequently at increased rendering costs.

LeBlanc *et al.* [60] addressed this issue by properly blending each hair's color using a pixel blending buffer technique. In this method, each hair strand is drawn as connected lines and the shaded color is blended into a pixel buffer. When using alpha-blending, one should be careful with the drawing order. Kim and Neumann [9] also use an approximate visibility ordering method to interactively draw hairs with OpenGL's alpha blending.

2) Implicit Representation: Volumetric textures (or *texels*) [61], [62] avoid the aliasing problem with pre-filtered shading functions. The smallest primitive is a volumetric cell that can be easily mip-mapped to be used at multiple scales. The cost of ray traversal is relatively low for short hairs, but can be high for long hairs. Also when hair animates, such volumes should be updated for every frame, making pre-filtering inefficient.

The rendering method of the cluster hair model [20] also exploits implicit geometry. Each cluster is first approximated by a polygonal boundary. When a ray hits the polygonal surface, predefined density functions are used to accumulate density. By approximating the high frequency detail with volume density functions, the method produces antialiased images of hair clusters. However, this method does not allow

changes in the density functions, making hairs appear as if they always stay together.

B. Light Scattering in Hair

The first requirement for any hair rendering system is a model for the scattering of light by individual fibers of hair. This model plays the same role in hair rendering as a surface reflection, or local illumination, model does in conventional surface rendering.

1) Hair Optical Properties: A hair fiber is composed of three structures: the cortex, which is the core of the fiber and provides its physical strength, the cuticle, a coating of protective scales that completely covers the cortex several layers thick (see Figure 11 in Section III-A), and the medulla, a structure of unknown function that sometimes appears near the axis of the fiber.

A hair is composed of amorphous proteins that act as a transparent medium with an index of refraction $\eta = 1.55$ [4], [63]. The cortex and medulla contain pigments that absorb light, often in a wavelength-dependent way; these pigments are the cause of the color of hair.

2) Notation and Radiometry of Fiber Reflection: Our notation for scattering geometry is summarized in Figure 21. We refer to the plane perpendicular to the fiber as the *normal plane*. The direction of illumination is ω_i, and the direction in which scattered light is being computed or measured is ω_r; both direction vectors point away from the center. We express ω_i and ω_r in spherical coordinates. The inclinations with respect to the normal plane are denoted θ_i and θ_r (measured so that 0 degree is perpendicular to the hair). The azimuths around the hair are denoted ϕ_i and ϕ_r, and the relative azimuth $\phi_r - \phi_i$, which is sufficient for circular fibers, is denoted $\Delta\phi$.

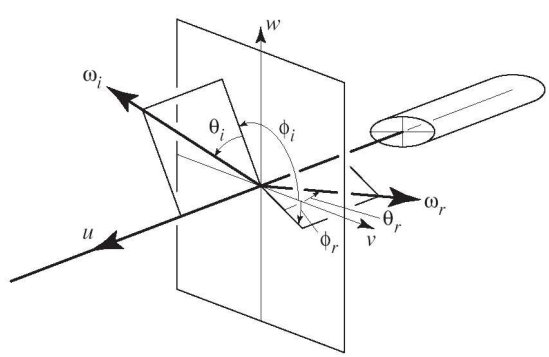

Fig. 21. Notation for scattering geometry

Because fibers are usually treated as one-dimensional entities, light reflection from fibers needs to be described somewhat differently from the more familiar surface reflection. Light scattering at a surface is conventionally described using the bidirectional reflectance distribution function (BRDF), $f_r(\omega_i, \omega_r)$. The BRDF gives the density with respect to the projected solid angle of the scattered flux that results from a narrow incident beam from the direction ω_i. It is defined as the ratio of surface radiance (intensity per unit projected area) exiting the surface in direction ω_r to surface irradiance (flux per unit area) falling on the surface from a differential solid angle in the direction ω_i:

$$f_r(\omega_i, \omega_r) = \frac{dL_r(\omega_r)}{dE_i(\omega_i)}.$$

Under this definition, the scattered radiance due to an incident radiance distribution $L_i(\omega_i)$ is

$$L_r(\omega_r) = \int_{H^2} f_r(\omega_i, \omega_r) L_i(\omega_i) \cos\theta_i d\omega_i$$

where H^2 is the hemisphere of directions above the surface.

Light scattering from fibers is described similarly, but the units for measuring the incident and reflected light are different because the light is being reflected from a one-dimensional curve [64]. If we replace "surface" with "curve" and "area" with "length" in the definition above we obtain a definition of the scattering function f_s for a fiber: "the ratio of *curve radiance* (intensity per unit projected length) exiting the curve in direction ω_r to *curve irradiance* (flux per unit length) falling on the curve from a differential solid angle in the direction ω_i." The curve radiance due to illumination from an incoming radiance distribution L_i is

$$L_r^c(\omega_r) = D \int_{H^2} f_s(\omega_i, \omega_r) L_i(\omega_i) \cos\theta_i d\omega_i$$

where D is the diameter of the hair as seen from the illumination direction.

This transformation motivated Marschner et al. [64] to introduce curve radiance and curve irradiance. Curve radiance is in some sense halfway between the concepts of radiance and intensity, and it describes the contribution of a thin fiber to an image independent of its width. Curve irradiance measures the radiant power intercepted per unit length of fiber and therefore increases with the fiber's width. Thus, given two fibers with identical properties but different widths, both will have the same scattering function but the wider fiber will produce a brighter curve in a rendered image because the wider fiber intercepts more incident light. This definition is consistent with the behavior of real fibers: very fine hairs do appear fainter when viewed in isolation.

Most of the hair scattering literature does not discuss radiometry, but the above definitions formalize the common practice, except that the diameter of the hair is normally omitted since it is just a constant factor. The factor of $\cos\theta_i$ is often included in the model, as was common in early presentations of surface shading models.

3) Reflection and Refraction in Cylinders: For specular reflection, a hair can be modeled, to a first approximation, as a transparent (if lightly pigmented) or purely reflecting (if highly pigmented) dielectric cylinder. The light-scattering properties of cylinders have been extensively studied in order to inversely determine the properties of optical fibers by examining their scattering [65], [66], [67].

As first presented in graphics by Kajiya and Kay [61] (their scattering model is presented in Section IV-B.5), if we consider a bundle of parallel rays that illuminates a smooth cylinder, each ray will reflect across the local surface normal at the point where it strikes the surface. These surface normals are

all perpendicular to the fiber axis–they lie in the normal plane. Because the direction of each reflected ray is symmetric to the incident direction across the local normal, all the reflected rays will make the same angle with the normal plane. This means that the reflected distribution from a parallel beam due to specular reflection from the surface lies in a cone at the same inclination as the incident beam.

For hairs that are not darkly pigmented, the component of light that is refracted and enters the interior of the hair is also important. As a consequence of Bravais's Law [68], a corrolary of Snell's Law, light transmitted through a smooth cylinder will emit on the same cone as the surface reflection, no matter what sequence of refractions and internal reflections it may have taken.

4) Measurements of Hair Scattering: In cosmetics literature, some measurements of incidence-plane scattering from fibers have been published. Stamm et al. [63] made measurements of reflection from an array of parallel fibers. They observed several remarkable departures from the expected reflection into the specular cone: there are two specular peaks, one on either side of the specular direction, and there is a sharp true specular peak that emerges at grazing angles. The authors explained the presence of the two peaks using an incidence-plane analysis of light reflecting from the tilted scales that cover the fiber, with the surface reflection and the first-order internal reflection explaining the two specular peaks.

A later paper by Bustard and Smith [69] reported measurements of single fibers, including measuring the four combinations of incident and scattered linear polarization states. They found that one of the specular peaks was mainly depolarized while the other preserved the polarization. This discovery provided additional evidence for the explanation of one lobe from surface reflection and one from internal reflection.

Bustard and Smith also discussed preliminary results of an azimuthal measurement, performed with illumination and viewing perpendicular to the fiber. They reported bright peaks in the azimuthal distribution, speculated that they were due to caustic formation, but they did not report any data.

Marschner et al. [64] reported measurements of single fibers in more general geometries. In addition to incidence plane measurements, they presented normal plane measurements that show in detail the peaks that Bustard and Smith discussed and how they evolve as a strand of hair is rotated around its axis. The authors referred to these peaks as "glints" and showed a simulation of scattering from an elliptical cylinder that predicts the evolution of the glints; this clearly confirmed that the glints are caused by caustic formation in internal reflection paths. They also reported some higher-dimensional measurements that show the evolution of the peaks with the angle of incidence, which showed the full scattered distribution for a particular angle of incidence.

5) Models for Hair Scattering: The earliest and most widely used model for hair scattering is Kajiya and Kay's model, which was developed for rendering fur [61]. This model includes a diffuse component and a specular component:

$$S(\theta_i, \phi_i, \theta_r, \phi_r) = k_d + k_s \frac{\cos^p(\theta_r + \theta_i)}{\cos(\theta_i)}.$$

Fig. 22. Comparison between Kajiya's model (left), Marschner's model (middle) and real hair (right).

Kajiya and Kay derived the diffuse component by integrating reflected radiance across the width of an opaque, diffuse cylinder. Their specular component is simply motivated by the argument from the preceding section that the ideal specular reflection from the surface will be confined to a cone and therefore the reflection from a non-ideal fiber should be a lobe concentrated near that cone. Note that neither the peak value nor the width of the specular lobe changes with θ or ϕ.

Banks [70] later re-explained the same model based on more minimal geometric arguments. For diffuse reflection, a differential piece of fiber is illuminated by a beam with a cross section proportional to $\cos\theta_i$ and the diffusely reflected power emits uniformly to all directions.[1] For specular reflection, Fermat's principle requires that the projection of the incident and reflected rays onto the fiber be the same.

In another paper on rendering fur, Goldman [71], among a number of other refinements to the aggregate shading model, proposed a refinement to introduce azimuthal dependence into the fiber scattering model. He multiplied both terms of the model by a factor f_{dir} that can be expressed in the current notation as:

$$f_{dir} = 1 + a\cos\Delta\phi.$$

Setting $a > 0$ serves to bias the model toward backward scattering, while setting $a < 0$ biases the model towards forward scattering.[2]

Tae-Yong Kim [72] proposed another model for azimuthal dependence, which accounts for surface reflection and transmission using two cosine lobes. The surface reflection lobe derives from the assumption of mirror reflection with constant reflectance (that is, ignoring the Fresnel factor), and the transmission lobe is designed empirically to give a forward-focused lobe. The model is built on Kajiya-Kay in the same way Goldman's is, defining:

$$g(\phi) = \begin{cases} \cos\phi & -\frac{\pi}{2} < \phi < \frac{\pi}{2} \\ 0 & \text{otherwise} \end{cases}$$

This model is Kajiya and Kay's model multiplied by:

$$f_{dir} = a\, g(\Delta\phi/2) + g(k(\Delta\phi - \pi))$$

where a is used to balance forward and backward scattering and k is a parameter to control how focused the forward

[1]Banks does not discuss why uniform curve radiance is the appropriate sense in which the scattered light should be uniform.

[2]In Goldman's original notation $a = (\rho_{reflect} - \rho_{transmit})/(\rho_{reflect} + \rho_{transmit})$. A factor of $\frac{1}{2}(\rho_{reflect} + \rho_{transmit})$ can be absorbed into the diffuse and specular coefficients.

scattering is. The first term is for backward (surface) scattering and the second term is for forward (transmitted) scattering.

Marschner *et al.* [64] proposed the most complete physically-based hair scattering model to date. Their model makes two improvements to Kajiya and Kay's model: it predicts the azimuthal variation in scattered light based on the ray optics of a cylinder, and it accounts for the longitudinal separation of the highlight into surface-reflection, transmission, and internal-reflection components that emerge at different angles. The azimuthal component of the model is based on a ray analysis that accounts for focusing and dispersion of light, absorption in the interior, and Fresnel reflection at each interaction. The longitudinal component models the shifts of the first three orders of reflection empirically using lobes that are displaced from the specular cone by specific angles.

6) Light Scattering on Wet Hair: The way light scatters on hair is changed when hair becomes wet. Jensen *et al.* [73] noted that when objects become wet they typically appear darker and shinier; hair behaves the same way. Bruderlin [74] and Ward *et al.* [33] altered previous light scattering models to capture the effects of wet fur and wet hair, respectively.

As hair becomes wet, a thin film of water is formed around the fibers, forming a smooth, mirror-like surface on the hair. In contrast to the naturally rough, tiled surface of dry hair, this smoother surface creates a shinier appearance of the hair due to increased specular reflections. Furthermore, light rays are subject to total internal reflection inside the film of water around the hair strands, contributing to the darker appearance wet hair has over dry hair. Moreover, water is absorbed into the hair fiber, increasing the opacity value of each strand leading to more aggressive self-shadowing (see Section IV-C).

Bruderlin [74] and Ward *et al.* [33] modeled wet strands by increasing the amount of specular reflection. Furthermore, by increasing the opacity value of the hair, the fibers attain a darker and shinier look, resembling the appearance of wet hair (see Figure 10).

C. Hair Self-Shadowing and Multiple Scattering

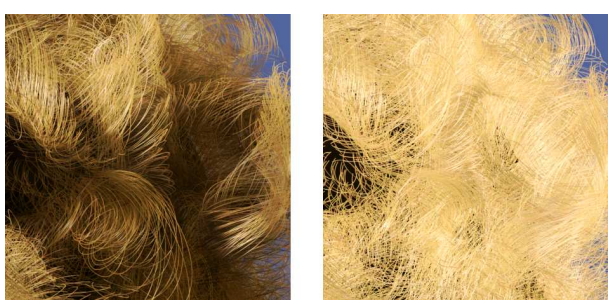

Fig. 23. Importance of self-shadowing on hair appearance. (left) Shadows computed using Deep Shadow Maps [75] compared to (right) No shadows. Images courtesy of Pixar Animation Studios.

Hair fibers cast shadows onto each other, as well as receiving and casting shadows from and to other objects in the scene. Self-shadowing creates crucial visual patterns that distinguish one hairstyle from another, see Figure 23. Unlike solid objects, a dense volume of hair exhibits complex light propagation patterns. Each hair fiber transmits and scatters rather than fully blocks the incoming lights. The strong forward scattering properties as well as the complex underlying geometry make the shadow computation difficult.

One can ray trace hair geometry to compute shadow, whether hair is represented by implicit models [61] or explicit models [64]. For complex geometry, the cost of ray traversal can be expensive and many authors turn to caching schemes for efficiency. Two main techniques are generally used to cast self-shadows into volumetric objects: ray casting through volumetric densities and shadow maps.

1) Ray-casting through a Volumetric Representation: With implicit hair representations, one can directly ray trace volume density [20], or use two-pass shadowing schemes for volume density [61]; the first pass fills volume density with shadow information and the second pass renders the volume density.

2) Shadow Maps: LeBlanc [60] introduced the use of the shadow map, a depth image of hair rendered from the light's point of view. In this technique, hair and other objects are rendered from the light's point of view and the depth values are stored. Each point to be shadowed is projected onto the light's camera and the point's depth is checked against the depth in the shadow map. Kong and Nakijima [58] extended the principle of shadow caching to the visible volume buffer, where shadow information is stored in a 3D grid.

In complex hair volumes, depths can vary radically over small changes in image space. The discrete nature of depth sampling limits shadow buffers in handling hair. Moreover, lights tend to gradually attenuate through hair fibers due to forward scattering. The binary decision in depth testing inherently precludes such light transmission phenomena. Thus, shadow buffers are unsuitable for volumetric hair.

The transmittance $\tau(p)$ of a light to a point p can be:

$$\tau(p) = \exp(-\Omega), \text{ where } \Omega = \int_0^l \sigma_t(l')dl'.$$

l is the length of a path from the light to p, σ_t is the extinction (or density) function along the path. Ω is the opacity thickness (or accumulated extinction function).

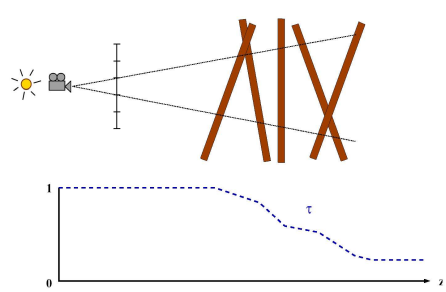

Fig. 24. Top: a beam of light starting at the shadow camera origin (*i.e.*, the light source) and passing through a single pixel of the deep shadow map. Bottom: the corresponding transmittance (or visibility) function τ, stored as a piecewise linear function.

In the deep shadow maps technique [75], each pixel stores a piecewise linear approximation of the transmittance function instead of a single depth, yielding more precise shadow

computations than shadow maps, see Figure 24 for an illustration. The transmittance function accounts for two important properties of hair.

Fractional Visibility: In the context of hair rendering, the transmittance function can be regarded as a fractional visibility function from the light's point of view. If more hair fibers are seen along the path from the light, the light gets more attenuated (occluded), resulting in less illumination (shadow). As noted earlier, visibility can change drastically over the pixel's extent. To handle this partial visibility problem, one should accurately compute the transmission function by correctly integrating and filtering all the contributions from the underlying geometry.

Translucency: A hair fiber absorbs, scatters and transmits the incoming light. Assuming that the hair fiber transmits the incoming light only in a forward direction, the translucency is also handled by the transmittance function.

Noting that the transmittance function typically varies radically over image space, but gradually along the light direction, one can accurately approximate the transmittance function with a compact representation. Deep shadow maps [75] use a compressed piecewise linear function for each pixel, along with special handling for discontinuities in transmittance (see Figure 23).

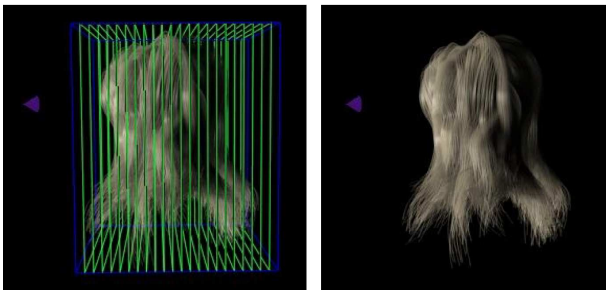

Fig. 25. Opacity Shadow Maps. Hair volume is uniformly sliced perpendicular to the light direction into a set of planar maps storing alpha values (top). The resulting shadowed hair (bottom).

Opacity shadow maps [76] further assume that such transmittance functions always vary smoothly, and can thus be approximated with a set of fixed image caches perpendicular to the lighting direction (see Figure 25). By approximating the transmittance function with discrete planar maps, opacity maps can be efficiently generated with graphics hardware (see Section IV-D.3). Linear interpolation from such maps facilitates fast approximation to hair self-shadows.

For light-colored hair, recent work has shown that shadowing and attenuation alone are insufficient to produce the correct appearance. For fully realistic results, light that reflects from hair to hair, or *multiple scattering*, must be accounted for. Photon mapping methods [77] can reduce per-frame rendering times from days, required for path tracing methods, to hours, but simulating multiple scattering in hair truly efficiently is still an open problem.

D. Rendering Acceleration Techniques

Accurately rendering complex hairstyles can take several minutes for one frame. Many applications, such as games or virtual reality, require real-time rendering of hair. These demands have initiated recent work to accelerate precise rendering algorithms by simplifying the geometric representation of hair, by developing fast volumetric rendering, or by utilizing recent advances in graphics hardware.

1) Approximating Hair Geometry: Section IV-B explained the structure of hair and showed that hair fibers are actually quite complex. Simplifying this geometry, using fewer vertices and rendering fewer strands, is one strategy for accelerating hair rendering. Removing large portions of hair strands can be distracting and unrealistic, therefore surfaces and strips have been used for approximating large numbers of hair strands [16], [49], [50], [78].

These two-dimensional representations resemble hair by texture mapping the surfaces with hair images and using alpha mapping to give the illusion of individual hair strands. Curly wisps can be generated by projecting the hair patch onto a cylindrical surface [78].

Level of detail (LOD) representations used by Ward *et al.* [55], [38] (see Section III-D.1) for accelerating the dynamic simulation of hair, also accelerates hair rendering. Using a coarse LOD to model hair that cannot be seen well by the viewer requires rendering fewer vertices with little loss in visual fidelity. As a result, the time required to calculate light scattering and shadowing effects is diminished by an order of magnitude.

2) Interactive Volumetric Rendering: Bando *et al.* [10] modeled hair as a set of connected particles, where particles represent hair volume density. Their rendering method was inspired by fast cloud rendering techniques [79] where each particle is rendered by splatting a textured billboard, both for self-shadowing computation and final rendering. This method runs interactively, but it does not cast very accurate shadows inside hair (see Figure 14).

Bertails *et al.* [80] use a light-oriented voxel grid to store hair density values, which enables them to efficiently compute accumulative transmittance inside the hair volume. Transmittance values are then filtered and combined with diffuse and specular components to calculate the final color of each hair segment. Though very simple, this method yields convincing interactive results for animated hair (see Figure 26). Moreover, it can easily be parallelized to increase performance.

Fig. 26. Interactive hair self-shadowing processed by accumulating transmittance values through a light-oriented voxel grid [80]. (left) Animated hair without self-shadows; (right) Animated hair with self-shadows.

3) Graphics Hardware: Many impressive advances have been made recently in programmable graphics hardware. Graphics processor units (GPUs) now allow programming of more and more complex operations through dedicated languages, such as Cg. For example, various shaders can directly be implemented on the hardware, which greatly improves performance. Currently, the major drawback of advanced GPU programming is that new features are neither easy to implement nor portable across different graphics cards.

Heidrich and Seidel [81] efficiently render anisotropic surfaces by using OpenGL texture mapping. Anisotropic reflections of individual hair fibers have also been implemented with this method for straightforward efficiency.

As for hair self-shadowing, some approaches have recently focused on the acceleration of the opacity shadow maps algorithm (presented in Section IV-C.2), by using the recent capabilities of GPUs. Koster *et al.* [78] exploited graphics hardware by storing all the opacity maps in a 3D texture, to have the hair self-shadow computation done purely in graphics hardware. Using textured strips to simplify hair geometry (as seen in Section IV-D.1), they achieve real-time performance. Mertens *et al.* [82] explored efficient hair density clustering schemes suited for graphics hardware, achieving interactive rates for high quality shadow generation in dynamically changing hair geometry. Finally, a real-time demonstration showing long hair moving in the sea was presented by NVidia in 2004 [83] to illustrate the new capabilities of their latest graphics cards (see Figure 27).

Fig. 27. Real-time rendering of long, moving hair using recent graphics hardware [83]. Image Courtesy of NVIDIA Corporation, 2004

V. New Challenges

As the need for hair modeling continues to grow in a wide spectrum of applications, the main focus for future research may be put either on physically-based realism (for cosmetic prototyping), visual realism with a high user control (for features films), or computations acceleration (for virtual environments and videogames). Some of these goals have been partially achieved, but many important issues still remain, especially in the field of hair animation.

A. Hairstyling

One of the most difficult challenges to virtual hairstyling remains to be creating intricate styles with a high level of user control in a short amount of time. There is typically a tradeoff between the amount of user control and the amount of manual input time. An interesting future direction in hairstyling could be to combine different shaping techniques in a manner that keeps a high degree of user control while still accelerating the time for user input. Moreover, haptic techniques for 3D user input have shown to be quite effective for mimicking real-world human interactions and have only recently been explored for hairstyling [7]. Attaining input through haptic gloves rather than through traditional mouse and keyboard operations is a possibility that could allow a user to interact with hair in a manner similar to real-world human-hair interactions. Creating a braid, for example, could potentially be performed in just minutes with haptic feedback, similar to real-world hairstyling.

In addition to user input, interactive virtual hairstyling techniques can also benefit from accelerations in rendering and simulation. While most styling techniques are targeted towards static hair, faster hair animation and rendering techniques would enable more realistic human-hair interaction. Styling of dynamic hair would be beneficial for cosmetic training and other interactive hairstyling functions. These high-performance applications demand the ability to interact accurately with hair via common activities, such as combing or brushing hair, in real time. But as explained in next Section, hair dynamic behavior as well as hair interactions are currently far from being satisfactorily simulated, especially in terms of accuracy.

B. Animation

Unlike some other mechanical systems, such as fluids, hair has not been deeply studied by physicists, and thus no macroscopic model describing the accurate dynamics of hair (individual and collective behavior) is currently available. Some recent work accounting for relevant structural and mechanical properties of hair starts to explore and to develop new mechanical models for simulating more closely the complex, nonlinear behavior of hair [43].

While hair animation methods still lack physically-based grounds, many advances have been made in terms of performance through the use of hair strips (Section III-C.2.a), FFD (Section III-C.1.d), and multi-resolution techniques (Section III-D), but each of these methods have various limitations to overcome. Hair strips can be used for real-time animation of hair, though hairstyles and hair motions are limited to simple examples due to the flat surface representation of the hair. Multi-resolution techniques have been able to model some important features of hair behaviors, including dynamic grouping and separation of hair strands, and have successfully accelerated hair simulation while preserving visual fidelity to a certain extent. However, highly complex hairstyles with motion constraints are still not simulated in real-time with these multi-resolution methods. FFD methods have been used to attain real-time animation of various hairstyles; nevertheless such approaches are limited mainly to small deformations of

hair. It would be interesting to explore the synthesis of one or more of these techniques by drawing on their strengths; for example, the use of an FFD approach that would allow for the hair volume to split into smaller groups for finer detail.

C. Rendering

Whereas very little physical data is available for hair mechanical properties, especially the way a collection of hair fibers behave together during motion, the microscopic structure of hair is well-known (Section IV-B.1). Measurements of hair scattering have recently led researchers to propose an accurate physically-based model for a single hair fiber, accounting for multiple highlights observable in real hair (Section IV-B.5). So far, this model is only valid for a single hair fiber. Other complex phenomena such as inter-reflection inside the hair volume should also be considered for capturing the typical hair lighting effects. Another important aspect of hair is self-shadowing. Many existing approaches already yield convincing results. The most challenging issue perhaps lies in simulating accurate models for both the scattering of individual hair fibers and the computations of self-shadows at *interactive* rates.

VI. Conclusion

We presented a literature review on hair styling, simulation, and rendering. For hairstyling, the more flexible methods rely mostly on manual design from the user. Intuitive user interfaces and pseudo-physical algorithms contribute to simplifying the user's task, while recent approaches capturing hair geometry from photographs automatically generate existing hairstyles. Various methods for animating hair have also been described, such as through a continuous medium or disjoint groups of hairs. Existing hair simulation techniques typically require a tradeoff among visual quality, flexibility in representing styles and hair motion, and computational performance. We also showed how multi-resolution techniques can be used to automatically balance this tradeoff. Finally, we discussed the main issues in hair rendering. We explained the effects of light scattering on hair fibers, explored techniques for representing explicit and implicit hair geometry, and examined different shadowing methods for rendering hair.

Hair modeling remains an active area of research. Depending on the specific field–styling, animation or rendering–different levels of realism and efficiency have been made. While hair rendering is probably the most advanced field, styling and above all animation still raise numerous unsolved issues. Researchers have begun to explore techniques that will enable more authentic user experiences with hair.

References

[1] N. Magnenat-Thalmann and S. Hadap, "State of the art in hair simulation," in *International Workshop on Human Modeling and Animation*, ser. Korea Computer Graphics Society, June 2000, pp. 3–9.

[2] R. Rosenblum, W. Carlson, and E. Tripp, "Simulating the structure and dynamics of human hair: Modeling, rendering, and animation," *The Journal of Visualization and Computer Animation*, vol. 2, no. 4, pp. 141–148, 1991.

[3] K. Anjyo, Y. Usami, and T. Kurihara, "A simple method for extracting the natural beauty of hair," in *Proceedings of ACM SIGGRAPH 1992*, ser. Computer Graphics Proceedings, Annual Conference Series, August 1992, pp. 111–120.

[4] C. R. Robbins, *Chemical and Physical Behavior of Human Hair*, 3rd ed. Springer-Verlag, New York, 1994.

[5] L'Oréal, "Hair science," 2005, http://www.hair-science.com.

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

[7] K. Ward, N. Galoppo, and M. Lin, "Interactive virtual hair salon," in *PRESENCE: Teleoperators & Virtual Environments (to appear)*, 2007.

[8] Y. Yu, "Modeling realistic virtual hairstyles," in *Proceedings of Pacific Graphics'01*, Oct. 2001, pp. 295–304.

[9] T.-Y. Kim and U. Neumann, "Interactive multiresolution hair modeling and editing," *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 620–629, July 2002, proceedings of ACM SIGGRAPH 2002.

[10] Y. Bando, B.-Y. Chen, and T. Nishita, "Animating hair with loosely connected particles," *Computer Graphics Forum*, vol. 22, no. 3, pp. 411–418, 2003, proceedings of Eurographics'03.

[11] D. Patrick and S. Bangay, "A lightwave 3d plug-in for modeling long hair on virtual humans," in *Proceedings of the 2nd international conference on Computer graphics, virtual Reality, visualisation and interaction in Africa*. ACM Press, 2003, pp. 161–187.

[12] E. Plante, M.-P. Cani, and P. Poulin, "A layered wisp model for simulating interactions inside long hair."

[13] B. Choe and H.-S. Ko, "A statiscal wisp model and pseudophysical approcahes for interactive hairstyle generation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 2, March 2005.

[14] L. Chen, S. Saeyor, H. Dohi, and M. Ishizuka, "A system of 3d hairstyle synthesis based on the wisp model," *The Visual Computer*, vol. 15, no. 4, pp. 159–170, 1999.

[15] B. Hernandez and I. Rudomin, "Hair paint," in *Computer Graphics International (CGI)*, June 2004, pp. 578–581.

[16] C. Koh and Z. Huang, "Real-time animation of human hair modeled in strips," in *Computer Animation and Simulation'00*, Sept. 2000, pp. 101–112.

[17] W. Liang and Z. Huang, "An enhanced framework for real-time hair animation," in *Pacific Graphics Conference on Computer Graphics and Applications*, October 2003.

[18] P. Noble and W. Tang, "Modelling and animating cartoon hair with nurbs surfaces," in *Computer Graphics International (CGI)*, June 2004, pp. 60–67.

[19] T.-Y. Kim and U. Neumann, "A thin shell volume for modeling human hair," in *Computer Animation 2000*, ser. IEEE Computer Society, 2000, pp. 121–128.

[20] X. D. Yang, Z. Xu, T. Wang, and J. Yang, "The cluster hair model," *Graphics Models and Image Processing*, vol. 62, no. 2, pp. 85–103, Mar. 2000.

[21] Z. Xu and X. D. Yang, "V-hairstudio: an interactive tool for hair design," *IEEE Computer Graphics & Applications*, vol. 21, no. 3, pp. 36–42, May / June 2001.

[22] D. Patrick, S. Bangay, and A. Lobb, "Modelling and rendering techniques for african hairstyles," in *Proceedings of the 3rd international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*. ACM Press, 2004, pp. 115–124.

[23] T. Wang and X. D. Yang, "Hair design based on the hierarchical cluster hair model," *Geometric modeling: techniques, applications, systems and tools*, pp. 330–359, 2004.

[24] S. Hadap and N. Magnenat-Thalmann, "Interactive hair styler based on fluid flow," in *Computer Animation and Simulation '00*, Aug. 2000, pp. 87–100.

[25] J. Stam, "Multi-scale stochastic modelling of complex natural phenomena," Ph.D. dissertation, University of Toronto, 1995.

[26] W. Kong, H. Takahashi, and M. Nakajima, "Generation of 3d hair model from multiple pictures," in *Proceedings of Multimedia Modeling*, 1997, pp. 183–196.

[27] S. Grabli, F. Sillion, S. R. Marschner, and J. E. Lengyel, "Image-based hair capture by inverse lighting," in *Proc. Graphics Interface*, May 2002, pp. 51–58.

[28] S. Paris, H. Briceño, and F. Sillion, "Capture of hair geometry from multiple images," *ACM Transactions on Graphics (Proceedings of the SIGGRAPH conference)*, 2004.

[29] Y. Wei, E. Ofek, L. Quan, and H.-Y. Shum, "Modeling hair from multiple views," in *Proceedings of ACM SIGGRAPH'05*, 2005.

[30] X. Mao, S. Isobe, K. Anjyo, and A. Imamiya, "Sketchy hairstyles," in *Proceedings of Computer Graphics International*, 2005.

[31] D.-W. Lee and H.-S. Ko, "Natural hairstyle modeling and animation," *Graphical Models*, vol. 63, no. 2, pp. 67–85, March 2001.

[32] J. T. Chang, J. Jin, and Y. Yu, "A practical model for hair mutual interactions," in *ACM SIGGRAPH Symposium on Computer Animation*, July 2002, pp. 73–80.

[33] K. Ward, N. Galoppo, and M. C. Lin, "Modeling hair influenced by water and styling products," in *International Conference on Computer Animation and Social Agents (CASA)*, May 2004, pp. 207–214.

[34] C. Bouillon and J. Wilkinson, *The Science of Hair Care, second edition*. Taylor & Francis, 2005.

[35] C. Zviak, *The Science of Hair Care*. Marcel Dekker, 1986.

[36] D. Pai, "Strands: Interactive simulation of thin solids using cosserat models," *Computer Graphics Forum*, vol. 21, no. 3, pp. 347–352, 2002, proceedings of Eurographics'02.

[37] D. Baraff and A. Witkin, "Large steps in cloth simulation," *Proc. of ACM SIGGRAPH*, pp. 43–54, 1998.

[38] K. Ward and M. C. Lin, "Adaptive grouping and subdivision for simulating hair dynamics," in *Pacific Graphics Conference on Computer Graphics and Applications*, October 2003, pp. 234–243.

[39] B. Choe, M. Choi, and H.-S. Ko, "Simulating complex hair with robust collision handling," in *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. New York, NY, USA: ACM Press, 2005, pp. 153–160.

[40] F. Bertails, T.-Y. Kim, M.-P. Cani, and U. Neumann, "Adaptive wisp tree - a multiresolution control structure for simulating dynamic clustering in hair motion," in *ACM SIGGRAPH Symposium on Computer Animation*, July 2003, pp. 207–213.

[41] S. Hadap and N. Magnenat-Thalmann, "Modeling dynamic hair as a continuum," *Computer Graphics Forum*, vol. 20, no. 3, pp. 329–338, 2001, proceedings of Eurographics'01.

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

[43] F. Bertails, B. Audoly, M.-P. Cani, B. Querleux, F. Leroy, and J.-L. Lévêque, "Super-helices for predicting the dynamics of natural hair," in *ACM Transactions on Graphics (Proceedings of the SIGGRAPH conference)*, August 2006.

[44] S. Hadap, "Hair simulation," Ph.D. dissertation, University of Geneva, 2003.

[45] D. Baraff, "Linear-time dynamics using lagrange multipliers," in *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press, 1996, pp. 137–146.

[46] A. Daldegan, N. M. Thalmann, T. Kurihara, and D. Thalmann, "An integrated system for modeling, animating and rendering hair," *Computer Graphics Forum*, vol. 12, no. 3, pp. 211–221, 1993.

[47] T. Kurihara, K. Anjyo, and D. Thalmann, "Hair animation with collision detection," in *Proceedings of Computer Animation'93*. Springer, 1993, pp. 128–138.

[48] P. Volino and N. Magnenat-Thalmann, "Animating complex hairstyles in real-time," in *ACM Symposium on Virtual Reality Software and Technology*, 2004.

[49] C. Koh and Z. Huang, "A simple physics model to animate human hair modeled in 2D strips in real time," in *Computer Animation and Simulation '01*, Sept. 2001, pp. 127–138.

[50] Y. Guang and H. Zhiyong, "A method of human short hair modeling and real time animation," in *Pacific Graphics*, Sept. 2002.

[51] E. Sugisaki, Y. Yu, K. Anjyo, and S. Morishima, "Simulation-based cartoon hair animation," in *Proceedings of the 13th Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2005.

[52] H. D. Taskiran and U. Gudukbay, "Physically-based simulation of hair strips in real-time," in *Proceedings of the 13th Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2005.

[53] Y. Watanabe and Y. Suenaga, "A trigonal prism-based method for hair image generation," *IEEE Computer Graphics and Applications*, vol. 12, no. 1, pp. 47–53, Jan 1992.

[54] E. Plante, M.-P. Cani, and P. Poulin, "Capturing the complexity of hair motion," *Graphical Models (Academic press)*, vol. 64, no. 1, pp. 40–58, january 2002.

[55] K. Ward, M. C. Lin, J. Lee, S. Fisher, and D. Macri, "Modeling hair using level-of-detail representations," in *International Conference on Computer Animation and Social Agents*, May 2003, pp. 41–47.

[56] E. Larsen, S. Gottschalk, M. Lin, and D. Manocha, "Distance queries with rectangular swept sphere volumes," *Proc. of IEEE Int. Conference on Robotics and Automation*, 2000.

[57] I. Neulander and M. van de Panne, "Rendering generalized cylinders with paintstrokes," in *Graphics Interface*, 1998.

[58] W. Kong and M. Nakajima, "Visible volume buffer for efficient hair expression and shadow generation," in *Computer Animation*. IEEE, 1999, pp. 58–65.

[59] D. P. Mitchell, "Consequences of stratified sampling in graphics," *ACM SIGGRAPH*, 1996.

[60] A. M. LeBlanc, R. Turner, and D. Thalmann, "Rendering hair using pixel blending and shadow buffers," *The Journal of Visualization and Computer Animation*, vol. 2, no. 3, pp. 92–97, – 1991.

[61] J. Kajiya and T. Kay, "Rendering fur with three dimensional textures," in *Proceedings of ACM SIGGRAPH 89*, ser. Computer Graphics Proceedings, Annual Conference Series, 1989, pp. 271–280.

[62] F. Neyret, "Modeling animating and rendering complex scenes using volumetric textures," *IEEE Transaction on Visualization and Computer Graphics*, vol. 4(1), Jan-Mar 1998.

[63] R. F. Stamm, M. L. Garcia, and J. J. Fuchs, "The optical properties of human hair i. fundamental considerations and goniophotometer curves," *J. Soc. Cosmet. Chem.*, no. 28, pp. 571–600, 1977.

[64] S. Marschner, H. W. Jensen, M. Cammarano, S. Worley, and P. Hanrahan, "Light scattering from human hair fibers," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 780–791, July 2003, proceedings of ACM SIGGRAPH 2003.

[65] C. L. Adler, J. A. Lock, and B. R. Stone, "Rainbow scattering by a cylinder with a nearly elliptical cross section," *Applied Optics*, vol. 37, no. 9, pp. 1540–1550, 1998.

[66] D. Marcuse, "Light scattering from elliptical fibers," *Applied Optics*, vol. 13, pp. 1903–1905, 1974.

[67] C. M. Mount, D. B. Thiessen, and P. L. Marston, "Scattering observations for tilted transparent fibers," *Applied Optics*, vol. 37, no. 9, pp. 1534–1539, 1998.

[68] R. A. R. Tricker, *Introduction to Meteorological Optics*. Mills & Boon, London, 1970.

[69] H. Bustard and R. Smith, "Investigation into the scattering of light by human hair," *Applied Optics*, vol. 24, no. 30, pp. 3485–3491, 1991.

[70] D. C. Banks, "Illumination in diverse codimensions," *Proc. of ACM SIGGRAPH*, 1994.

[71] D. Goldman, "Fake fur rendering," in *Proceedings of ACM SIGGRAPH'97*, ser. Computer Graphics Proceedings, Annual Conference Series, 1997, pp. 127–134.

[72] T.-Y. Kim, "Modeling, rendering, and animating human hair," Ph.D. dissertation, University of Southern California, 2002.

[73] H. W. Jensen, J. Legakis, and J. Dorsey, "Rendering of wet material," *Rendering Techniques*, pp. 273–282, 1999.

[74] A. Bruderlin, "A method to generate wet and broken-up animal fur," in *Computer Graphics and Applications, 1999. Proceedings. Seventh Pacific Conference*, October 1999, pp. 242–249.

[75] T. Lokovic and E. Veach, "Deep shadow maps," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 385–392.

[76] T.-Y. Kim and U. Neumann, "Opacity shadow maps," in *Rendering Techniques 2001*, ser. Springer, July 2001, pp. 177–182.

[77] J. T. Moon and S. R. Marschner, "Simulating multiple scattering in hair using a photon mapping approach," *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 25, no. 3, 2006.

[78] M. Koster, J. Haber, and H.-P. Seidel, "Real-time rendering of human hair using programmable graphics hardware," in *Computer Graphics International (CGI)*, June 2004, pp. 248–256.

[79] Y. Dobashi, K. Kaneda, H. Yamashita, T. Okita, and T. Nishita, "A simple efficient method for realistic animation of clouds," in *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 19–28.

[80] F. Bertails, C. Ménier, and M.-P. Cani, "A practical self-shadowing algorithm for interactive hair animation," in *Graphics Interface*, May 2005, graphics Interface'05.

[81] W. Heidrich and H.-P. Seidel, "Efficient rendering of anisotropic surfaces using computer graphics hardware," *Proc. of Image and Multidimensional Digital Signal Processing Workshop (IMDSP)*, 1998.

[82] T. Mertens, J. Kautz, P. Bekaert, and F. V. Reeth, "A self-shadow algorithm for dynamic hair using density clustering," in *Proceedings of Eurographics Symposium on Rendering*, 2004.

[83] C. Zeller, R. Fernando, M. Wloka, and M. Harris, "Programming graphics hardware," in *Eurographics - Tutorials*, September 2004.

Kelly Ward Kelly Ward is currently a software engineer at Walt Disney Feature Animation, where she works on look development and hair modeling tools for feature films. She received her M.S. and Ph.D. degrees in Computer Science from the University of North Carolina, Chapel Hill in 2002 and 2005, respectively. She received a B.S. with honors in Computer Science and Physics from Trinity College in Hartford, CT in 2000, where she was named the President's Fellow in Physics in 1999-2000. Her research interests include hair modeling, physically-based simulation, and computer animation. She has given several presentations and invited lectures on her hair modeling research at international venues.

Florence Bertails Florence Bertails is currently working at INRIA Rhtne-Alpes, France, as a post-doctoral researcher. She graduated in 2002 from the Telecommunication Engineering School of Institut National Polytechnique de Grenoble (INPG) and received a MSc in Image, Vision and Robotics. She has just completed a Ph.D from the INPG where she worked on hair simulation and physically-based modeling, in collaboration with physicists and hair scientists from L'Orial Research. She presented her Ph.D work at international conferences such as the ACM-EG Symposium of Computer Animation and Eurographics, and received the best student paper award at Graphics Interface 2005 for interactive rendering of animated hair. Her latest work on hair animation will be published at ACM SIGGRAPH 2006.

Tae-Yong Kim Tae-Yong Kim is currently a research scientist at Rhythm and Hues Studios. His responsiblities at R&H include research and development of animation tools for movie productions. His work was used in many film productions such as the the Chronicles of Narnia, Superman Returns, X-Men 2 and several other movies.

He holds a Ph.D in computer science from the University of Southern California where he researched human hair modeling and rendering. His Ph.D work was published in SIGGRAPH 2002 and other conferences. In addition, he holds an M.S. in computer science from U.S.C. and a B.S. in computer engineering from the Seoul National University.

He has taught in recent SIGGRAPH courses and has reviewed many academic papers for such venues as SIGGRAPH, Symposium on Computer Animation, Eurographics and others.

Stephen R. Marschner Stephen R. Marschner is an Assistant Professor of Computer Science at Cornell University. He received his Sc.B. in Mathematics and Computer Science from Brown University in 1993 and his Ph.D. in Computer Science from Cornell in 1998, then held research positions at Hewlett-Packard Laboratories, Microsoft Research, and Stanford University before joining the Cornell faculty in 2002. He is the recipient of a 2003 Technical Achievement Award from the Academy of Motion Picture Arts and Sciences, an NSF CAREER award in 2004, and an Alfred P. Sloan Research Fellowship in 2006. Marschner's research interests include simulating the optics and mechanics of complex everyday materials for computer graphics.

Marie-Paule Cani Marie-Paule Cani is a Professor of Computer Science at the Institut National Polytechnique de Grenoble (INPG), France. A graduate from the Ecole Normale Supirieure, she received a PhD from the University of Paris Sud in 1990 and an "habilitation" degree from INPG in 1995. She was awarded membership of the Institut Universitaire de France in 1999. She is the head of the INRIA research group EVASION which she created in 2003. Her main research interests cover physically-based animation, interactive modelling techniques and the design of layered models incorporating alternative representations and LODs. Recent applications include pattern-based texturing, the animation of natural phenomena such as lava-flows, ocean, vegetation and human hair, real-time virtual surgery and interactive sculpting or sketching techniques. Marie-Paule Cani co-chaired IEEE Shape Modeling International (SMI) in 2005 and was paper co-chair of EUROGRAPHICS 2004 and of the ACM-EG Symposium on Computer Animation (SCA) in 2006. She served in the editorial board of Graphical Models (GMOD) from 2001 to 2005 and joined the editorial board of IEEE TVCG in 2006.

Ming C. Lin Ming Lin received her B.S., M.S., Ph.D. degrees in Electrical Engineering and Computer Science all from the University of California, Berkeley. She is currently a full professor in the Computer Science Department at the University of North Carolina (UNC), Chapel Hill. She received the NSF Young Faculty Career Award in 1995, Honda Research Initiation Award in 1997, UNC/IBM Junior Faculty Development Award in 1999, UNC Hettleman Award for Scholarly Achievements in 2002, and five best paper awards at international conferences.

Her research interests include haptics, physically-based modeling, robotics, and geometric computing and has authored over 140 refereed publications in these areas. She has served as the conference organizer and program chair of more than a dozen of technical conferences, as well as the steering committee member of ACM SIGGRAPH / Eurographics Symposium on Computer Animation and World Haptics Conference. She is also an associated editor and a guest editor of several journals and magazines, including IEEE Transactions on Visualization and Computer Graphics, International Journal on Computational Geometry and Applications, IEEE Computer Graphics and Applications, and ACM Computing Reviews. She has given numerous lectures and invited presentations at many international conferences.