Expressive rendering of animated hair

Noura Faraj

Institut National de Recherche en Informatique et en Automatique

Expressive rendering of animated hair

MoSIG Project

presented in public June 24, 2009

by

Noura FARAJ

Supervisors:	Florence BERTAILS Joelle THOLLOT
Jury members:	James CROWLEY Marie-Christine FAUVET Gilles SERASSET
Expert:	Franck HÉTROY

Acknowledgments

I would like to express my gratitude to all those who gave me the possibility to complete this master project. I want to thank my supervisors Florence BERTAILS and Joëlle THOLLOT for their suggestions and encouragement.

It was a real pleasure to work among ARTIS team. I want to thank my colleagues for all their help, support, interest and valuable hints. Especially, I am obliged to Pierre Bernard . I also want to thank

Contents

List of Figures

Introduction

Context

Hair simulation is one of the crucial elements of a character realism in video games as well as animated movies. It is also one of the most challenging because of its complex nature. A simulation model needs to be able to handle hair fibers or wisp interaction while keeping the desired rendering style. During the past few years intensive work has been done in this field. Most of the authors have tried to render and animate hair as realistically as possible. Impressive results have been obtained and computation times have been reduced. Nevertheless this level of realism is not always desired by the animator. Most animated characters are represented with a hair model only composed of a few hair wisps or clumps in other words the individual hair fibers are not even accounted for. Only little work has been done to animate and render non-photorealistic hair for cel-characters[1].

The goal of this work is to design an expressive rendering technique for a realistic animation of hair. This project is a part of an ANR research program for a joint industrial project with two production studios: Neomis Animation and BeeLight, two other INRIA project-teams: Bipop and Evasion and a CNRS lab (Institut Jean Le Rond d'Alembert de l'Université Pierre et Marie Curie). The aim of this project is to provide hair rendering and animating tools for movie making. According to the discussions we had with artists from Neomis studio, it appears that an animator will expect realism of hair motion combined with an expressive rendering technique that is dedicated to animated movies.

To the best of our knowledge, it would be the first expressive rendering technique combined with a realistic animation of hair. The main idea is to use as an input a physically based hair simulation done using Super-Helices [BAC+06] and render using Diffusion curves primitive [OBW+08]. We will use 3D as well as 2D information to deal with view point changes, curve intersections, delimitation of the hair model...

Outline

Chapter 1 is an overview of the previous work in rendering of animated hair. We will see the extensive work that has been done for photorealistic rendering and the few expressive rendering methods proposed. Chapter 2 presents the motivation of this work and the simulation choices. Our solution for expressive rendering of a realistic hair animation will be developped in chapter 3. Limitations and perspectives will be discussed in a last chapter.

[1] cartoon-like characters, the name comes from celluloid: clear sheets of acetate, called cels, painted on in traditional 2D animation

1

Previous work

In this section, we will go through the different solutions proposed for realistic as well as expressive hair simulation. The way hair is rendered depends on what one wishes to represent. Hence, Section 1.1 presents how hair can be represented: as individual hair fibers, hair clumps or a volume. Then, Section 1.2 reviews the techniques used so far to animate hair. An overview of the extensive work done to realistically render hair models is proposed in Section 1.3. Finally, the few existing expressive rendering techniques will be presented in Section 1.4.

1.1 Hair representation

Hair can be viewed in different manners and the model choices to simulate and render hair strongly depend on what one wishes to represent. In this section, we will see how artist have perceived and represented hair then we give a brief overview of the corresponding representation choices for the simulation.

1.1.1 Aimed artistic style

Drawing three dimensional and shiny looking hair models is a challenge for artists. Sometimes, hair is modeled as groups of strands, or wisps, and it can be also perceived as one large volume.

Individual hair fibers representation

Individual hair fibers are drawn by the artist to either represent realistic-looking hair models or to add information about the hair model. For instance in Figure 1.1, representing hair strands in the left drawing makes the hair model look realistic and three-dimensional whereas on the right one drawn strands help stylize the hair model.

Hair represented as wisps or clumps

These disjoint hair strands have a natural tendency to be grouped in a hair wisp or clump. As a result, a hairdo can be represented using only these groups. This means that only the shape of the wisps is represented, it is not filled with hair strands. Surfaces of hair are used to represent hair wisps and their union gives the global hair model. This level of detail is usually used to draw Anime[2] and manga hair models (Figure 1.2).

[2] Japanese style of motion-picture animation

Figure 1.1: Hair fibers representation. (Left) Realistic drawing of hair [Sou]. (right) Hand-drawn animated character.

Figure 1.2: Examples of hair clump representation. (left) Anime hair model (middle) Tarzan hair style using hair clump (right) Dragon Ball Z hair wisps.

Hair represented as a cohesive mass

A hair model can be represented by simply drawing its global shape. Only the outline is drawn and shading or highlights are added by the artist to give volumic information. Cartoon hair is often represented as long and flowing hair that looks like a cohesive mass.

Figure 1.3: Hair represented as a continuum.

In order to simulate those different kinds of hair representations the appropriate geometric model has to be used. The next section presents the possible representation choices for rendering and animation hair models.

1.1.2 Geometric representation for hair simulation

Hair is typically seen as a set of individual strands, or one-dimensional curves in three-dimensional space. A full head of hair usually counts 100 000 to 200 000 strands therefore the representation choice depend on the level of detail desired. As we will see in the following sections, hair behavior can be simulated at a given level (individual hair fiber, wisp or as a continuum) and rendered at a different one depending of the aimed rendering result.

Hair representation for animation

One-dimensional curves: the behavior of each hair fiber and their interactions are simulated individually. Because hair strands have a tendency to be grouped only a few guide hair strands can be used for this kind of simulation (for details see Section 1.2.2). This kind of representation is usually used to physically animate hair models.

Two-dimensional surfaces: one of the traditional technique to animate cel-looking hair models is to use key-frame interpolation (see Section 1.2.1). The hair wisps are represented by two-dimensional surfaces that are animated by interpolation of their user-defined positions.

Three-dimensional surfaces: a full hair model can be considered as a global geometry which motion will direct the hair primitives (Section 1.2.3). Otherwise, the animation can be controlled as a continuous medium.

Hair representation for rendering

One-dimensional curve: each hair strand is explicitly represented to render photorealistic hair models. Hence, a hair fiber is modeled using a cylinder (see section 1.3). This is a tedious task because each hair strand is thiner than a pixel. Even if hair is modeled at another level of detail dfor instance wisps, clumps or volumic representation for the simulation, the final goal is to render and represent individual hair fibers.

Two-dimensional surfaces: to render patch-like hair styles, models of hair strips through two-dimensional surfaces (defined in image space) can be used. A single surface can also represent the shape of the global hair model. Any kind of rendering technique can be used on either representation. This kind of representation in mainly used to render expressive hair styles (Section 1.4).

Three-dimensional surfaces: "clumpy" hair models are usually represented using three-dimensional surfaces to render disjoints hair wisps. Those surfaces are often extruded or defined along a guide curve.

1.2 Animation

This section presents how hair can be animated using key-frame interpolation, physically-based simulation of individual hair strands and finally using hybrid techniques.

1.2.1 Key-frame interpolation

Key-frame interpolation, also called *in-betweening*, is an animation technique widely used to create the motion of virtual characters. This animation technique is derived from earliest hand-drawn animated movies. The idea behind it is pretty simple: An artist draws a few sets of specific key images and another artist manually interpolates to get the intermediate images. The interpolation is the process of generating intermediate frames between two images to give the appearance that the first image evolves smoothly into the second image. Key-frame animation techniques make this process automatic. Intermediate frames are generated by the interpolation of graphic parameters.

This animation method is used by Côté *et al.* [CJDO04] to animate patch-like hair wisps. For each key-frame, the user manually draws a few strokes from which a set of control points are defined to generate the hair wisp. The system interpolates the position of wisps control points to obtain the *in-between* frames. The animation is therefore entirely hand made. As a result the animator in completely in charge of the realism of the hair motion. Moreover, the key-frame positions are defined by drawing on an image which means that no view point changes are possible to generate an animation. The motion is then restricted to the viewing plan.

1.2.2 Physically-based animation of individual hair strands

Individual hair strands can be animated with a physical simulation meaning that the user only has to define the properties of the hair model such as the number of hair strands, curliness, damping, length... Then the system computes the hair motion and interactions for a given animation (response to given forces). Three main techniques have been proposed to physically animate individual hair strands: mass-springs systems, rigid multi-body serial chains and Cosserat based models. A thorough comparison of those methods is proposed by Ward *et al.* [WBK+07], this section only presents a brief overview of each technique.

Mass-Springs Systems

Rosenblum *et al.* proposed one of the first approaches to model features of individual hair strands by considering their structure and the control of their motion [RER91]. A mass-springs system, defined by a set of particles linked by springs, is used to control the position and motion of a single hair strand.

Mass-spring models are easy to implement and allow to easily deal with hair collision or constraints. Their main limitation is that spring stiffness has to be increased to respect the inextensibility property of a hair strand leading to a numerical instability and a small simulation time step. During the past few years, significant improvement occurred using constraint mass-springs systems [PCP01, BKCN03]. The inextensibility condition being difficult to achieve mass-springs system are better suited for the simulation of extensible wavy hair strands.

Recently, a mass-springs system solving the hair torsion issue has been proposed Selle by *et al.* [SLF08]. The goal is to represent each hair fiber in the simulation so no hair has to be added during the rendering step (Fig 1.4). Therefore, each hair strand is represented by an evolutive mass-springs system. Their model allows hair torsion. Furthermore a semi-implicit discretization of standard springs, making them linear in multiple spatial directions and unconditionally stable, is proposed. Complex interactions can be simulated. However, the inextensibility constraint is difficult to achieve because of the inherent properties of mass-springs systems.

Figure 1.4: Full hair models simulations [SLF08]. (Left) Ten thousand long straight hair (Middle) Ten thousand medium length straight hairs (Right) Five thousand long curly hairs.

Rigid multi-body Serial Chain

Hadap and Magnenat-Thalmann use an articulated chain of rigid to model the motion of an individual hair strand [HMT01]. This formulation, through reduced coordinates, allows the parametric definition of hair curvature and torsion. A hair strand is modeled by a set of rigid segments linked by spherical joints with three degrees of freedom as represented Figure 1.5. This model allows to simulate hair torsion as well as curvature while keeping the length constant but external constraints becomes tedious to add while keeping the system stable. Chang *et al.* [CJY02] use this method to simulate guide hair strands directing the hair model motion. Compared to the previous technique, it enhances realism allowing hair torsion while keeping a constant length.

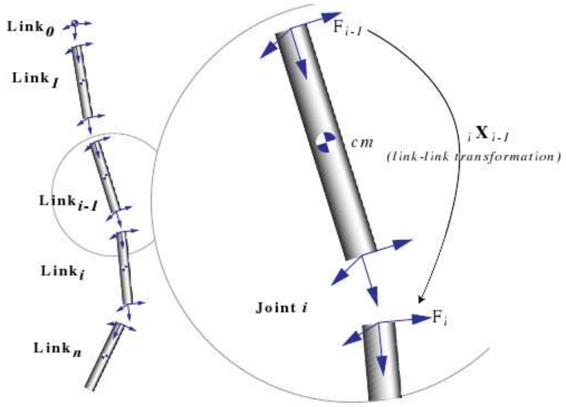

Figure 1.5: Hair strand as rigid multibody serial chain [HMT01].

Anjyo *et al.* [AUK92] use one dimensional projective equations which are a simplification of articulated chain of rigids by associating directions in space. This method is stable and easy to implement but hair torsion as well as the influence of the tip of the hair on the root is not modeled. This technique is well suited for the simulation of straight non extensible hair.

Cosserat-based models

The Cosserat model [CF09] propose a mathematical description of rods using 3D geometric curves and adapted reference frames. Pai was the first to introduce it to the computer graphic community [Pai02] to interactively simulate strands under Kirchoff hypothesis - physical requirement of inextensibility and unshearability. Bertails *et al.* [BAC+06] show how Kirchhoff's equations can be used to accurately describe hair motion. Those equations account for the non linear behavior of hair when bending and twisting. The authors introduce a deformable model to solve those equations. Each hair strand is represented by a Super-Helix - a piecewise helix - that will be animated using the principles of Lagrangian mechanics. This method offers stable simulations of realistic hair with a reasonable time step for various type of hair (curly, wavy, straight...).

Figure 1.6 illustrates results for a full head of hair of different kind of hair models. This technique models bending, torsion, intrinsically preserves a constant hair length and takes into account soft constraints such as external forces. However, reduced coordinates being used, handling hard constraint can be tricky.

Those three techniques have been used to simulate individual hair strands animation. Only one of them, [BAC+06], ensures all the hair properties as it is summed up in Table 1.7. Until now, these realistic simulations of hair motion have only been used to animate realistic looking hair models.

Figure 1.6: Dynamic simulations of hair using Super-Helices [BAC+06].(Left) A Super-Helix, (Middle) to (Right) wavy, curly, straight hair simulations.

	Mass-springs	Articulated chains of rigids	Dynamic Super-helices	Mass-springs [SLF08]
Bending	yes	yes	yes	yes
Torsion	no	yes	yes	yes
Perfect inextensibility	no	yes	yes	no
Curliness	no	no	yes	yes
Constraints	easy	tricky	tricky	easy

Figure 1.7: Comparative table of models for dynamic animation of individual hair strands proposed by [WBK+07] and adapted by [SLF08].

1.2.3 Hybrid methods

Hair simulation, like all physical simulation, is difficult to direct while preserving plausibility of the motion. That is why a mixture of key-frame interpolation and physically-based animation have been proposed. The existing hybrid animation methods either use key-frame interpolation and somehow adds a layer of dynamics or propose a tool to retrieve a physically-based animation sequence using sketches.

Animation and dynamics

Montoya-Vozmediano *et al.* [MVH02] uses a mixture of NURBS[3] surfaces for the overall motion and mass-springs to add realism to the animation. A key-frame animation is used to drive the movement combined with different layers of dynamic, composed of angular as well as linear springs. The combination is done using a relaxation process with a factor defined by the user that allows to set different values for the roots and the tips. This technique was used to simulate the motion of realistic-looking individual hair fibers inside the global surface. Noble and Tang [NT04] extended this method by filling the NURBS volume with clumps of hair geometry. As shown in Figure 1.8, key hair curves are defined along the surface. The dynamic is added by putting particles on each control vertex of the key hair that will direct the hair clump defined around it.

This approach gives good results by enhancing motion realism but is limited to a certain kind of motion and hair (long flowing hair). It does not allow all kind of hairdo: curly, short or sparse.

[3] Non-Uniform Rational Basis Splines

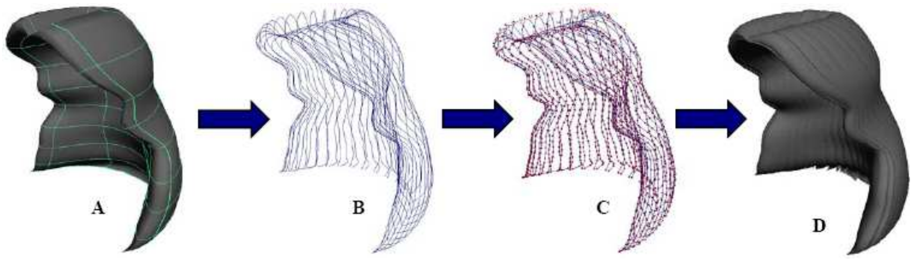

Figure 1.8: Example of hybrid animation [NT04]: A - NURBS surface, B - Key curves, C - Dynamic layer., D - Extruded hair clumps.

Sketch and motion database

Sugisaki *et al.* [SYAM05] propose an hybrid animation technique that uses rough sketchs of a few key-frames of the animation as indicator of hair motion and a predefined database of physical simulations. The dataset is made by the user, who needs to be an animator. For each motion sequence, he draws the hair wisps from which 3D hair wisps are extracted. Each one of them is composed of a multi-body rigid chain (Section 1.2.2) centerline to model the wisp dynamic. Then, he defines the forces that will generate the motion.

This animation technique can be decomposed into two steps: A matching process to find the desired motion sequence and key-frames interpolated through the animation using a transition function. The sketches should be close enough to the database information to enable the matching process of the 3D model and the drawings.

This method allows the user to create different animations using only a few key-frames that gives him control on a physically-based simulation. Nevertheless, the animator needs to create a database of motion for each animation sequence which can be time consuming.

In this section, we have seen how hair can be animated using key-frame interpolation of user defined positions, physically-based simulation of individual hair strands and finally hybrid technique that try to offer a control over a physically-based simulation. The main problem of key-frame interpolation and hybrid technique is that specific skills are required for the user to obtain a coherent motion. When we watch a cel character moving, we can say if the hair motion is not corelated to the action. Thus, realism of hair simulation adds meaning to the motion and is often desired. Next sections present the rendering models that have been proposed to obtain the aimed hair representation.

1.3 Realistic rendering

Realistic hair rendering methods rely on the real hair strands structure and interactions with light and each other. Local as well as global properties need to be respected to obtain a realistic hair style. The rendering algorithm has to define the way each hair fiber is illuminated - local property -, the light scattering among the hair volume and the self-shadowing of hair - global properties. This section presents how light is scattered by an individual hair fiber and then what are the proposed model to account for this phenomena.

1.3.1 Light scattering

Scattering is a general physical process where a form of radiation, such as light, is forced to deviate from a straight trajectory by one or more localized non-uniformities in the medium through which it passes. Reflections that undergo scattering are often called diffuse reflections and unscattered reflections are called specular (mirror-like) reflections.

Hair fiber properties

Hair strands are not completely opaque which means that both light reflection and transmission have to be taken into account while rendering. Hence, a realistic hair rendering method needs to model the scattering of light by individual fibers of hair. The shiny appearance of hair in mainly due to the light bounced off or transmitted multiple times through strands.

That is why physical behavior and structure of hair have been closely studied because the rendering model strongly depends on it. The composition and structure of hair is known (cortex, cuticle and medulla) as well as its chemistry. We also know that it behaves as a transparent medium. For an explicit representation of hair, each hair fiber is locally assumed to be a cylinder (Figure 1.9).

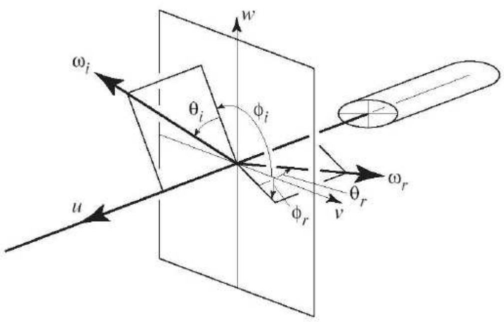

Figure 1.9: Light scattering notations [WBK+07].

Before presenting how light is scattered on a surface let us define how light scatters in a most common case: light scattering at a surface that is defined by a bidirectional reflectance distribution function (BRDF), $f_r(\omega_i, \omega_r)$ where ω_i is the direction of illumination and ω_r the direction in which light is scattered. It is defined as the ratio of surface radiance L, intensity per unit projected area, exiting the surface in direction ω_r to surface irradiance$E$, flux per unit area, falling on the surface from a differential solid angle in the direction ω_i:

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

The radiance due to an incoming 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.

Defining light scattering function, noted f_s, from hair fibers is quite similar, with D the hair diameter seen from the illumination direction:

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

where f_s gives the ratio of curve radiance, intensity per unit of 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. Marschner *et al.* [MJC+03] introduce curve radiance, contribution of a thin fiber to an image independent of its width, and curve irradiance, radiant power intercepted per unit length (increases with D). computing the light scattering function is tedious therefore simplified model, dedicated to hair and fur rendering, are proposed to account for light scattering.

Light scattering models

The earliest model for hair scattering was proposed to render fur by Kajiya and Kay [KK89]. Their model is still the most used nowadays. It is composed of a diffuse and a specular component:

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

The diffuse component is defined by integrating reflected radiance across the width of an opaque, diffuse cylinder. It is based on the hypothesis that reflection from a non ideal fiber should be a lobe concentrated near a cone representing an ideal specular reflection from the surface is used. Nevertheless, changes in θ and ϕ do not affect the peak value nor the width of the specular lobes meaning that the specular reflexion intensity do not vary with the incident angle. An other work focusing on rendering fur, proposed by Goldman [Gol97], extends their method by adding an azimuthal dependency into the fiber scattering model. To do so, both terms are multiplied by a factor dependent of this azimuthal angle (around the fiber axis). Depending on its value, this factor affects either backward (surface) scattering or forward (transmitted) scattering. Another way of introducing an azimuthal dependency is to use cosine lobes for surface reflection and transmission as presented by Tae-Yong Kim [Kim02]. Their model is Kajiya's one multiplied by a factor balancing backward and forward scattering.

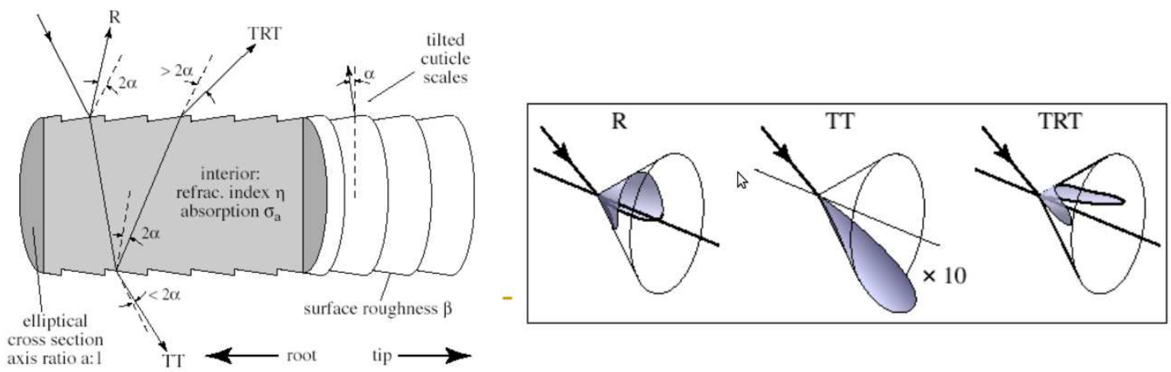

Figure 1.10: Light scattering measurement of hair fibers [MJC+03]. Two specular peaks observed: surface reflection, light passes through the interior of the fiber.

Finally, the most complete physically-based hair scattering model has been proposed by Marschner *et al.* [MJC+03]. Their model separates the highlight into surface reflection, transmission and internal reflection (Figure 1.10). Furthermore, their measurements, discussed in Section 1.3.1, show a multiple specular highlight and variation in scattering with rotation around the fiber axis. They model the sources of these effects using a transparent elliptical cylinder with an absorbing interior and a surface covered with tilted scales (Figure 1.10) as a model of a hair fiber. Using this method a scattering function for a circular cylinder is proposed as well as a practical shading model for hair that accounts for the two measured specular peaks not predicted by Kajiya and Kay's model.

Figure 1.11: Light scattering comparison [MJC+03]. (left) Kajiya model, (middle) Marschner model and (right) real hair.

Figure 1.11 shows a comparison between the two main methods used to model hair light scattering. We can see that Marschner's model gives a result that is visually closer to the real hair picture than Kajiya's model.

1.3.2 Self shadowing

In the previous sections, we have seen that the hair structure and the behavior of hair have been extensively studied and that satisfying approaches proposed to model light scattering of a single hair fiber. But to get a realistic-looking hair model, those models are not sufficient. The global properties of a hair model have to be studied. For instance, the way hair fibers cast shadows on each other needs to be accounted for when rendering. Figure 1.12 illustrates the importance of self-shadowing on hair appearance.

The difficulty of computing shadows for a hair volume is due to its complex nature. As we have seen in Section 1.3.1, light is neither fully blocked nor fully transmitted. Furthermore, the number of hair fibers amplify the complexity of the problem. Two main techniques are usually used to cast shadow into volumetric objects: ray casting through volumetric densities and Shadow Maps.

Ray casting method

An implicit representation of hair, a volumetric model, can be chosen for rendering. Ray tracing is usually the most accurate method of calculating shadows through a volumetric representation. Volume density can be directly ray traced [YXWY00]. A two pass shadowing scheme can be

Figure 1.12: Importance of self-shadowing on hair appearance [WBK+07]. (left) Shadows computed using Deep Shadow Maps [LV00] compared to (right) No shadows. Images courtesy of Pixar Animation Studios.

used for volume density [KK89]. The first pass fills volume density with shadow information and the second pass renders the volume density.

Shadow Map Approach

When an explicit hair model is used, each individual hair is represented by a three-dimensional curved cylinder, thus the approach is different. The earliest technique using Shadow Maps was proposed by LeBlanc *et al.* [LTT91]. Shadows are created by testing whether a pixel is visible from the light source using a z-buffer or depth image from the light source's view. The main problem of the explicit representation of hair is aliasing due to the small diameter of a hair strand (0.1 mm). A pixel contains several fraction of different hair fibers (Figure 1.13) which causes abrupt changes in color if hair are undersampled. The method uses pixel blending combined with Z-buffer and shadow buffer information from the scene to get a final anti-aliased image with soft shadows.

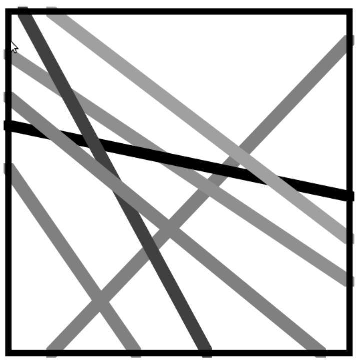

Figure 1.13: Hair strand pixel coverage for an explicit representation [LTT91].

The Shadow Map does not account for the fact that hair is not completely opaque therefore using Deep Shadow Map [LV00], that stores a piecewise linear approximation of the transmittance function at each pixel, improves the result. Opacity Maps [KN01] are also an improvement of Shadow Maps. The hair volume is uniformly sliced perpendicularly to the light direction and slices are rendered in the order of the distance from the light sources as shown Figure 1.14. The frame-buffer, saved after each slice, is rendered. Then, hair strands are projected onto opacity maps. The main problems of Opacity Map are that it assumes that shadow is linear between two maps (Two neighboring opacity maps alpha values are interpolated). It is fast but gives incorrect self shadow.

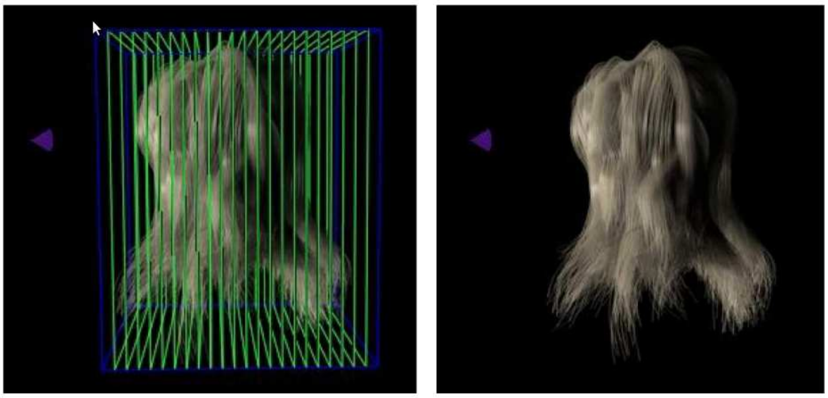

Figure 1.14: Opacity Shadow Maps [KN01].

In this section, we have seen the different methods proposed to obtain a realistic-looking hair rendering. To do so, hair local properties (light scattering) and global properties (self-shadowing) have been studied and mimicked by numerous methods. Impressive results have been obtained and computing time reduced. The main new challenges of realistic hair rendering are to account for inter-reflections inside the hair volume and to propose simulating models for both scattering and self-shadow at interactive rate.

1.4 Expressive rendering

Such level of realism is not always desired, this section presents the expressive rendering techniques. Only little work has been done to expressively render a hair simulation. The target rendering style is to make cartoon-like hair look like hand drawn. Nevertheless, the approach is different from photorealistic hair rendering. The rendering style depends on the chosen representation (Section 1.1): an expressive rendering style usually does not represent each hair fiber, only a small number of wisps is drawn (Section 1.1.1).

Most of the expressive hair simulation methods use a toon shading to give a Non-Photorealistic Rendering (NPR) style to the hair model. The shading of cartoon hair is nearly always shaded with areas of flat color composed of highlights, lowlights and an emphasized outline. Hair rendering methods try to match one or more of those features. We will define the toon shading technique, then presented the alternative methods proposed to expressively render hair model. So far only a few techniques propose expressive rendering methods dedicated to hair.

1.4.1 Toon shading for hair rendering

Cel-shaded animation (also called cel-shading or toon shading) is a type of non-photorealistic rendering technique designed to make computer graphics appear to be hand-drawn. This kind of shading is often used to mimic the style of a comic book or a cartoon. Flat region of constant colors are rendered using a small number of discrete shades. Shadows and highlights appear more like blocks of color rather than mixed in a smooth way.

Typically, the outline and contour lines are emphasized by a thick black line. Several techniques exist to achieve this feature. The most common one involves the inversion of the backface culling. The back-facing triangles are either drawn with thick lines or multiple times with slight changes in translation. The outline thickness is constant, thus, it seems to grow when the object size reduces (when the object gets further from the camera). An alternative technique is to render first a slightly bigger copy of the object (or only its silhouette). As a result, the outline changes with the distance to the camera. The silhouette can also be extracted using an edge detector on a reference image.

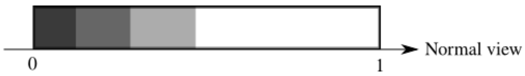

Figure 1.15: 1D toon texture example.

Facing triangles are then rendered with no lighting giving a bright look to the object that has no shadow. A texture look-up is done on a toon texture similar to Figure 1.15, to find out where each pixel center falls on the texture. For each object point, the texture coordinate depend on the angle between the normal vector and the view direction. Finally, the shade is applied to the resulting image.

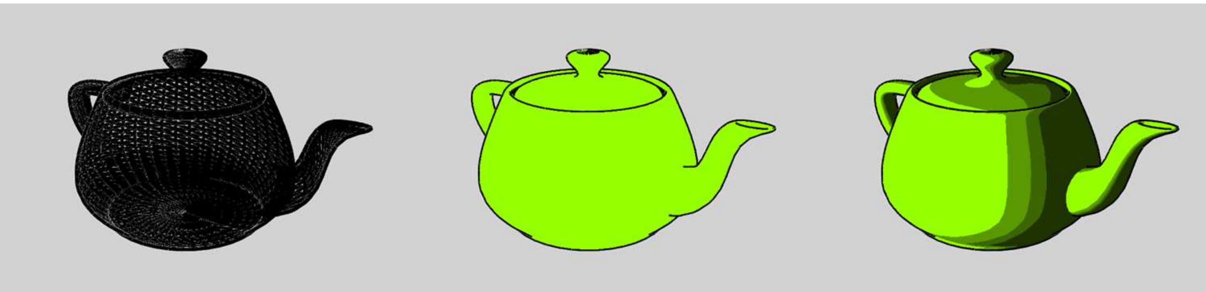

Figure 1.16: The Utah teapot rendered using cel-shading from wikipedia. (left) Back faces are drawn with thick lines. (center) Object rendered with no lighting. (right) Toon shading

Most of the expressive rendering techniques use a simple toon shader to give a hand-drawn looking style to the hairdo. Noble and Tang [NT04] implementation of toon shading uses the basic flat shading with two-tone to highlight the hair clumps and the hairs' natural tendency to stick together (Figure 1.17). Sugisaki *et al.* [SYAM05] create the thickness of their hair wisp to implement a toon shading [AG99].

Shin *et al.* [SHM06] propose a modified toon diffuse shading. First, hair wisps are represented using billboards then sorted according to their depth and relative position to their neighbor. The resulting configuration is rendered into a texture on which a Sobel edge detection filter in used to extract the silhouette. The intensity of the wisps silhouette is faded near the root because the re-arrangement of hair wisps could cause "jumping" effects near the root. The billboard particles are then rendered using the Painter algorithm: from the furthest to the closest according to the eye position. This technique combined with a step-function similar to

Figure 1.17: Toon shading used for hair rendering. (left) Two-tone toon shading on a ponytail [NT04]. (right) Toon shading on hair clumps [SYAM05].

Figure 1.15 can cause unwanted shading effects therefore a texture as represented in Figure 1.18 is used to generate the diffuse lighting.

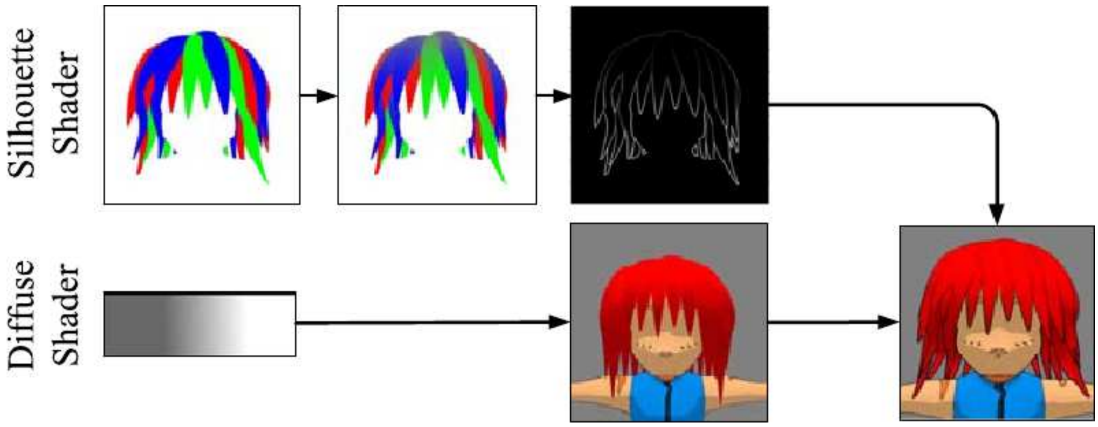

Figure 1.18: Toon shading pipeline [SHM06].

Traditional toon shading uses 1D texture to describe how tone varies. Barla *et al.* [BTM06] propose a toon shader that support view-dependent effects by adding a second dimension to the texture. Therefore an information about the “tone detail” can be stored and used.

Toon shading offers flat or limited shading but only a few alternatives are proposed to render hair expressively. The outline of the hair model can be emphasized or a specular highlight defined to add a cel-look to the hair model.

1.4.2 Hair model outline emphasis

Mao *et al.* [MIAI05] propose a top down approach to model (using a sketch based interface) and render an expressive hair model. This work is based on their previous technique but offers better user control and improvements over the rendering style [MKIA04]. Their rendering technique is based on the assumption that only the silhouette of cartoon-like characters hair model is drawn and then a few strokes represented inside it to give volumetric information.

Hair wisps are represented using cluster polygon which is a polygon strip. Silhouette and shadow lines extraction are done on a color reference image. The outline pixels are found using a neighborhood comparison of colors. Shadow lines are drawn using Deussen and Strothotte’s technique [DS00] that uses depth discontinuities to determine what parts of the primitives are to be drawn. Figure 1.19 illustrates the rendering results.

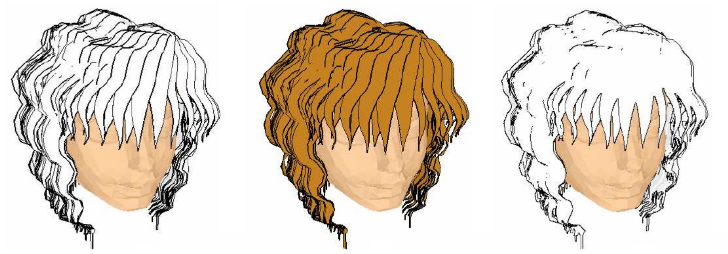

Figure 1.19: Outline emphasis [MIAI05]. (left) Render all outlines (right) Render only silhouette and shadow lines.

This method helps to emphasize of the outline and shadow lines but the color inside the hair model is uniform. As opposed to a toon shader, that also accentuates the outlines, no shading is proposed. As a result the hair model seems flat.

1.4.3 Specular highlight

A specular highlight is a bright spot of light that appears on shiny objects when illuminated. Specular highlights are important in 3D computer graphics, as they provide a strong visual information about the object shape and its location with respect to light sources. Two methods have been proposed so far to add specular highlights to expressive hair models: specular view-independent highlight and *feathering* style highlights.

Stylized hair renderer

In addition to the diffuse toon shading presented in Section 1.4.1, Shin *et al.* [SHM06] propose a method to add a view-independent stylized highlight. Their specular term is defined using the tangent and normal vector, noted respectively T and N, of hair wisp billboard:

$$specular = K_s.lightColor.(max(L.W, 0))^{shininess}$$

where L is the direction towards the light source, and W a weighted factor defining the specular highlight position on the hair model:

$$W = N.w + T.(1 - w)$$

Using this technique highlight can be moved from the tip to the root by changing the weight value w from 0 to 1. The specular model is composed of particles that follow the hair model shape. The user defines a highlight threshold that defines how those particles are merged to obtain a triangle strip that is shaped thank to a user defined texture.

This specular highlighting model can not be used for most characters. Furthermore, numerous user inputs are needed thus getting a nice rendering result might be time consuming and tedious.

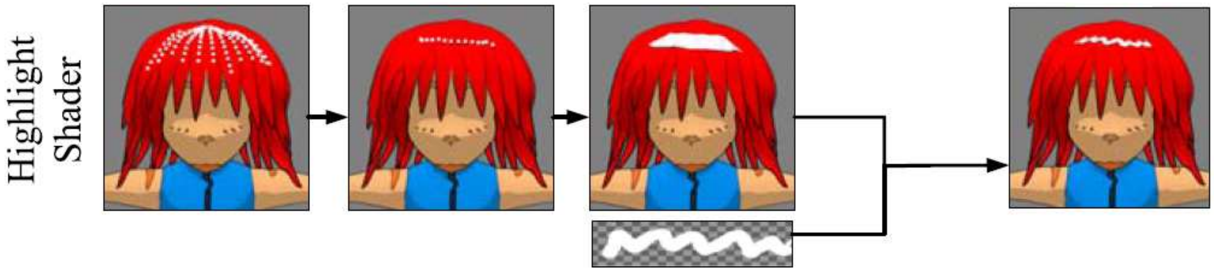

Figure 1.20: The different pipeline steps for generating specular highlight [SHM06]: (a) all particles are marked with a special specular highlight threshold (b) Potential particles are merged (c) and define a triangle strip, used for rendering a highlight texture.

Feathering highlights

Côté *et al.* [CJDO04] propose a method that uses *feathering* inking technique to simulate lighting. *Feathering* is done by drawing strokes in the direction of the hair strands while emphasizing the highlights with ink stains (Figure 1.21).

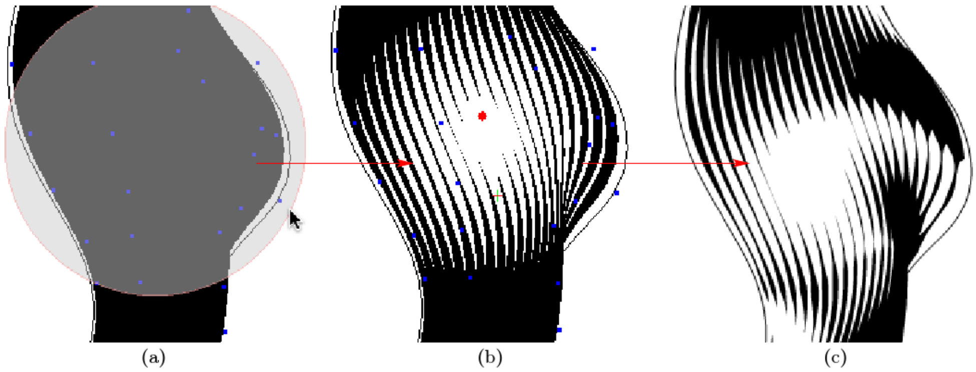

Figure 1.21: Highlight spots generation [CJDO04]. (a) Circular region manually placed by the user. (b) Automatic positioning of a pseudo-light. (c) User defined depth perturbations to give a 3D look to the highlight.

Wisps are represented by surfaces delimited by the animator. The system generates a bicubic patch defined by control points extracted from strokes, converted to cubic curves, drawn by the user and linearly interpolated over the surface. Their shape can be adjusted locally by the mean of control points or globally using FFD (Free Form Deformation). The user places the highlight in 2D and fixes it width (circular region with a user defined radius placed over the wisps). Then the system computes the pseudo-light 3D position such that the shading fits the overall given specular region. To give a 3D look to the highlight, depth of control points can be interactively adjusted. The highlight is rendered using a series of polygons uniformly distributed over the patch and whose size depends on the pseudo-lights' position.

This method intuitive for artist because only a few set of parameter that can be interactively manipulated. To preserve coherence through a key-frame animation either multiple or moving light source are used. It offers a nice cartoon rendering but limited animation and shapes for hair models but also for highlights. As a matter of fact, only circular hightlight are proposed

Figure 1.22: *Feathering* highlights results [CJDO04].

whereas toon characters are often represented with long and thin highlight region along hair wisps. Furthermore, aliasing can appear depending on the polygons size.

It is difficult and tedious to get a stylized highlight animation for 3D models when using conventional cartoon-shading algorithms. Anjyo *et al.* [AH03] propose an highlight shader that depicts cartoon-style highlights for 3D objects in cel animation that could be used to enhance cartoon look of the chosen rendering technique.

1.5 Conclusion

As it has been shown in this section, intensive research has been conducted concerning the simulation of realistic hair. Nevertheless, only few methods have been proposed for the animation and rendering of non-photorealistic ones. Most of the authors propose limited key-frame animations and toon shading for which temporal and spatial coherence is tedious, if not impossible, to maintain. Furthermore the hairstyle is often restricted to simple patch-like shapes which strongly limits the artist's work. In this work, we propose a method that combines physically-based hair animation and expressive rendering. The next chapter will explain the different steps of the algorithm, the representation we chose and how we mix 3D and 2D information to get an expressive rendering result.

2

Overview

In this chapter we will see what motivated this work, in Section 2.1, what is the aimed rendering style and the simulation choices made to obtain it in Section 2.2. The challenges raised by expressive rendering of a realistic hair animation are presented Section 2.3.

2.1 Motivation: Physically based animation of expressive hair

The study of the previous works emphasized a lack of expressive rendering techniques dedicated to animated hair. Our work is an attempt to fill this void by using a physically based simulation of an expressive hair model. Cel-character hair motion needs to be plausible, fluid and to follow head movements. As seen in Section 1.2.1, manually creating a hair animation that offers the appropriate answer to the character motion is a tedious task. Furthermore, our discussions with the production studio members of the ANR project emphasized a real demand from the animators for a physically-based animation of cartoon-like hair. It is important to distinguish realistic animation from realistic rendering. We can have a physically-based animation of a cel hair model. A few guide hairs need to be animated and we can associate to them any kind of geometry depending on what we want to represent. So we believe that it would be an interesting tool for animators to be able to generate a realistic looking animation of an expressive hairstyle. To our knowledge, it would be the first expressive rendering technique combined with a realistic animation of hair.

The main idea is to transcribe the motion of the 3D guide hair strands into a time and view consistent animation of 2D vector primitives generating an expressive hairdo. Any kind of geometry could be represented and animated using the motion of those guide hair strands.

2.2 Simulation choices

This section presents the representation, animation and rendering choices that we have made in this work. Among the animation models represented in the previous chapter, we will use the dynamic Super-Helices to simulate the behavior of a few guide hair strands and Diffusion curves as a vector graphic rendering primitive.

2.2.1 Hair representation

The rendering pipeline strongly depends on the aimed hair look and the chosen technique to represent it. This section presents the aimed style and the chosen primitives to represent it.

Aimed style

Section 1.1 presented different artistic hair model representations. Cartoon-like characters usually have a hairdo composed of a few wisps and individual hair fibers are rarely represented. Highlights and shades over the wisps give the volumic information. We want to represent this kind of hair style composed of delimited hair wisps.

Figure 2.1: Examples of manga hairdo close to our aimed style.

As presented in the previous section, only one NPR technique proposes a diffuse shading [SHM06]. Other techniques use classical toon shading that give flat and limited results. We want to represent patch-like hair styles with smooth-shaded colors.

Chosen primitive

To get a cel-looking hairstyle, we choose to represent the hair model using patch that will be animated. Thus, we need to find a well suited representation for those patches. We choose to use 2D vector-primitives to represent the patches delimitation because vector-based primitives support a variety of operations including geometry-based editing, key-frame animation, ready stylization, and are resolution-independence. We can easily animate them using their control points. Furthermore, we want to render stylized hair with smooth-shaded colors. We then choose to use a 2D vector-based primitive that creates smooth-shaded images: Diffusion curves [OBW+08]. A Diffusion curve partitions the space through which it is drawn, defining different colors on each side. These colors may vary smoothly along the curve in real time which makes this primitive well suited for animation. Moreover, Figure 2.2 shows smooth-shaded hair models obtained with Diffusion curves that reinforce our primitive choice. Further details about Diffusion curves are given in the next section.

2.2.2 Rendering primitive

To represent and stylize the hair model, we use Diffusion curves [OBW+08] vector primitive. Those primitives generate images with diffused color using an intuitive mean. By only drawing edges, which contain the features informations, and setting a few color and blur values the user

Figure 2.2: Characters designed with Diffusion curves with the aimed hair style.

can create, edit and eventually animate a vector based image that he can easily stylize. The Diffusion curves represent edges defined as Bézier curves. The control is proposed for both shape, color and blur via control points. Control points colors are linearly interpolated on each side using two curves of color (Figure 2.3 (2)). These colors are diffused, using a Poisson equation, and then reblured. The curve color gradient, expressed as a gradient field **w**, which is zero everywhere except on the curve, creates a sharp transition between right and left colors.

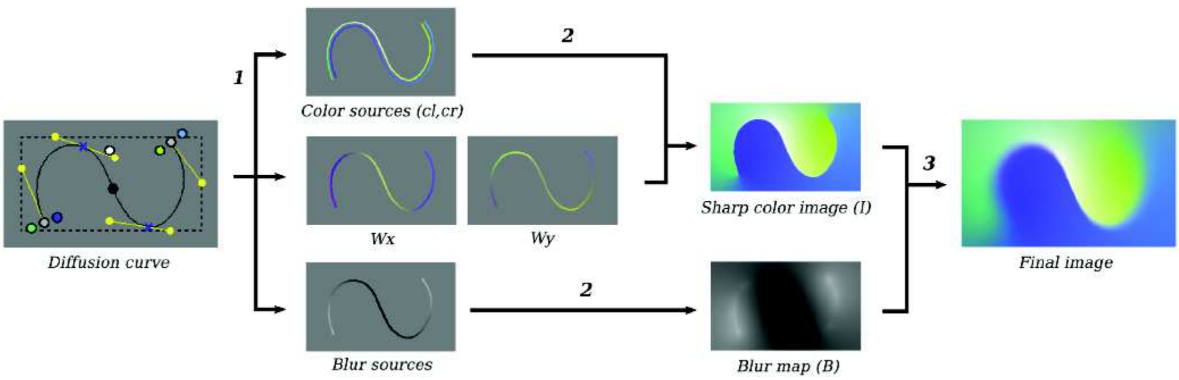

Figure 2.3: Diffusion curve [OBW+08] rendering. (1) rasterization of the color and blur sources, along with the gradient field $\mathbf{w} = (\mathbf{w}_x; \mathbf{w}_y)$, (2) diffusion of colors and blur, and (3) reblurring of the color image.

Three images are used: C containing colored pixel on each side of the curve, W_x gradient component and W_y gradient component. To diffuse colors a Poisson equation is solved:

$$\Delta I = div\mathbf{w}$$
$$I(x, y) = C(x, y) \text{ if pixel } (x, y) \text{ stores a color value}$$

where Δ and div are the Laplace and divergence operators.

Blur values are interpolated and diffused in the same way. The original blur values are only defined on the curve and are set to zero otherwise. This diffusion gives a blur map that is used to reblur the color image. Solving a Poisson equation leads to a global solution, therefore any color can influence any pixel of the final image.

Those features make the Diffusion curves well suited for our problem because we can easily define a shadow color on one side, a wisp color on the other and smooth shaded result are provided. The animation of the Diffusion curves can be done using the Bézier curves control points. Figure 2.2 shows characters designed using diffusion curves. We can see that the hair model are represented using a small number of curves, a shadow color and a wisp color (respectively white and purple for the right most character).

2.2.3 Animation

Section 1.2 has presented the different existing hair models. And as we have seen, it is tedious to obtain the desired animation using key-frame animation, and hybrid methods proposed so far are limited to a specific kind of hair style and motion. Furthermore, there is a real demand from animators for realistic simulations of expressive hair. For those reasons, we choose to use as an input a simulation of a sparse set of hair strands using Super-helices. As presented in the previous chapter, it is the only method that allow to simulate accurately the behavior of different hair types and styles. The method simulates the overall motion using a few guide hair strands. As a result, we have as an output of the simulation the 3D position of each guide hair strand for each frame.

The projected sampled positions of the guide hair strands are used to generate Diffusion curves. Their motion is guided by the physically-based simulation. A straightforward application does not solve the issues of hair rendering as it is developed in the next section.

2.3 Challenges

As presented in the previous sections, we use 2D curves, defined in the image-space, to represent hair wisps. This means that hair patches are represented by their outlines and no surface is defined. This fact raises a number of problems presented in this section and overcome in the next chapter.

2.3.1 Hair visibility

Because no surface is explicitly defined, dealing with hair occlusion is challenging. Which hair patches have to be drawn on top of the other has no straight forward solution. Accounting for hair visibility is a main part of this work. The proposed solution is presented Section 3.3.

2.3.2 Delimitation of the hair model and head rendering

Because we are using Diffusion curves, any color can influence any pixel of the final image. The color diffusion needs to be "stopped" in order to get a closed hair model. In the same way, hair color must not be diffused inside the character body and the other way around. A method needs to be found to solve those issues as shown in Section 3.4.

2.3.3 Temporal and spatial coherence

As pointed out previously, we are using 2D curves to render a 3D animation and the camera can be moved around the 3D scene. Hence, temporal and spatial coherence needs to be maintained.

Furthermore, we do a per frame processing by applying our rendering technique on the projected animated guide hair strands thus temporal coherence is also challenging.

In this work we will propose a technique that combines a physically-based hair simulation - dynamic Super-Helices - and an expressive rendering technique obtained using a vector primitive - Diffusion curves. We will express the hair motion by the means of 2D curves and more precisely edges that convey the relevant information about the hair model (depth, color, intersections...). This method allows to obtain a realistic motion of an NPR primitive and try to overcome the challenges of hair simulation.

3

Contribution

This chapter presents a rendering pipeline dedicated to animated hair. As we presented in Chapter 2 the main challenges raised are hair occlusion (by the head and one another) and delimitation of the hair model while keeping spatial and temporal coherence.

3.1 Rendering pipeline

As an input, we have an animation of guide hair strands simulated by Super-Helices. For each frame of the simulation the following information are stored: head motion through head frame position and orientation and sampled hair strands positions. If needed, any other information can be extracted from the simulation.

Our system defines a 3D scene composed of the head mesh, used for the simulation, and the 3D hair strand sampled points. The camera can be move in the scene to render different view-points.

For each view-point change or animation frame the rendering pipeline detailled in Figure 3.1 in done. The 3D scene is rendered in a reference image used to test head occlusion. Those projected points are used to extrude the wisp patch. Then the patch's inside and outside colors are defined. Finally, the background and the body are rendered.

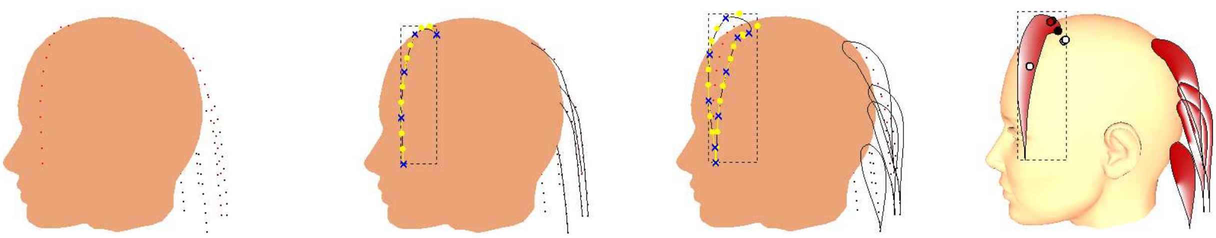

Figure 3.1: Rendering pipeline for a few wisps. From left to right: Reference image with the projected sampled points, the central Bézier spline, the extruded patch defined by the closed curve, then the final display.

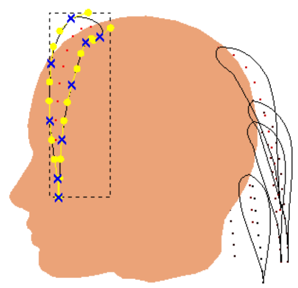

Figure 3.2: An example of patch control points.

3.2 Hair wisp generation

The rendering pipeline strongly depends on the hair geometric representation. As presented section 2.2.2, we chose to use patches to represent hair wisps and Diffusion curves as a vector primitive to define them. As it is usually done for hair simulation, the hair representation for rendering does not have to match the one for animation. Therefore, the patches motion is defined by the motion of the 3D guide hair strands. Our system automatically generates the hair wisps using the animation data.

3.2.1 Patch extrusion

A Diffusion curve is used as the outline of a hair wisp which control positions are defined as followed. For each strand, the 3D sampled points are projected in the camera reference frame. The resulting 2D points set, noted P_B, is used as a control points list to generate a geometric curve defined by a cubic Bézier spline B. Moreover, the depth information, i.e. the corresponding 3D point distance to the view point, is stored at each control point and linearly interpolated along the curve. Therefore, even if we work in the image space a third dimension is added. The curve parametric coordinates are noted $B(t) = (x, y, z)$ with $t \in [0, 1]$ where $B(0)$ is the root of the guide hair strand and $B(1)$ its tip.

The patch-like hair wisp is extruded along the resulting Bézier spline. To compute the Diffusion curve control points positions and information, we define the patch width as an offset, $d(t)$ with $t \in [0, 1]$, along B. Note that the offset value is obtained by given a finite set of control width values D that are linearly interpolated along the curve. Therefore, the shape of the patch can be controlled through D, for instance a pointy wisp can be obtained by setting a width control point to zero at the tip then $d(1) = 0$. If no width control points are defined a default value is used.

The extrusion is done with respect to the normal direction, noted N, of the central Bézier spline. For any point of B we can compute two extruded points, p_l and p_r each one located on a half space of the curve, with the following equations:

$$p_l = B(t) + N(t).d(t)$$
$$p_r = B(t) - N(t).d(t)$$

with $d(t)$ the width scalar value and $N(t) = (x, y, 0)$. The points p_l and p_r are expressed using three-dimensional coordinates that hold the depth information of $B(t)$. Therefore, a set of sorted control points P is computed to define the corresponding wisp Diffusion curve. Those points are order such that a closed curve is obtained with the root of the extruded hair strand as a starting, and thus finishing, point.

3.2.2 Color and shadow

Each guide hair strand is now represented as a patch-like hair wisp defined by a closed curve. Therefore, half space of each curve define an *inside* and an *outside* of the wisp. Diffusion curves allow shape color transitions in between both side therefore only the color set on the inside influences the wisp color. Nevertheless, the outside color of two overlapping patches colors changes the resulting inside diffusion. Using this fact, hair wisps cast shadow on one another by setting a shadow color as the outside color of each hair wisp.

At this point, patches of uniform inside color are obtained. To get a diffuse shading of each hair wisps a shadow or highlight color point is placed on the same side of each Diffusion curve. this technique gives a volumic look to the hair wisp. Moreover, the constant position of the color point makes hair model look like it is illuminated by a fixed light source. The size of the highlight can be controlled by adding color control points in the wisp. For instance, a wisp color point at a symmetric position regarding the central curve reduces the width of the highlight.

In our examples, we used the same color for the entire hair model but each hair wisp could have its own color to stylize the character.

3.3 Hair visibility

Hair wisps are defined in the image space and are always facing the camera. This representation is view-dependent therefore the rendering result has to be recomputed at each camera motion. For each view-point some parts of hair strands are hidden by each other of by the head the system needs to account for these occlusions.

3.3.1 Head occlusion

The viewer cannot see hair if it is not in the camera field of view or completely occluded by the head or other objects in the scene. Therefore, hidden hair strand are not represented to reduce computing times and render accurately the hair model. To do so, a visibility test is done for each sampled point of the 3D hair strands. In order to do that, the 3D scene is rendered at the current camera position into a reference image giving a different color for each individual strand (Figure 3.3).

A sample point is visible if the pixel at its projected position holds the color assigned to its originating strand. For each hair strand, visibility is test from the root to the tip. Only with the

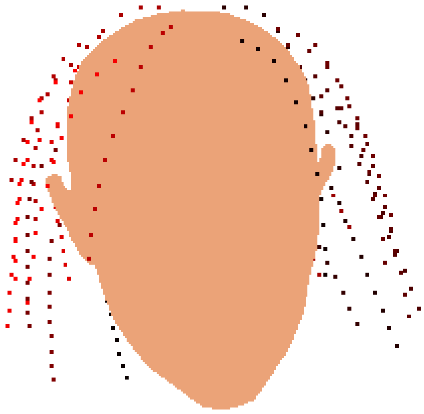

Figure 3.3: Reference image used for hair occlusion, the head is rendered with no lighting to keep its color uniform and each hair strand is assigned a given color.

top part of the hair strand visible are rendered. To simplify the algorithm and reduce computing time, we only perform this test on the upper part of the hair strand.

The visible curves are used to generate patches, as describe in Section 3.2, that intersect in the image plan and occlude each other.

3.3.2 Hair mutual occlusion

A wisp surface is defined by its outline in image space therefore depth information is only know on the curve. Thus, in order to render resulting patches while respecting their 3D depth arrangement, we need to deal with Diffusion curves intersections. They need to be detected and processed before rendering the patches.

Intersection detection and storage

On one hand, detecting curve intersection points can be done using a comparison of neighborhood pixel colors on a reference image furthermore intersecting curves can easily be identified because each one is assigned a given color before rendering. But on the other hand, finding the parametric coordinate of the intersection is tedious. To overcome this problem, the hair model is represented as an arrangement of curves in the image space.

Computational Geometry Algorithms Library (CGAL) [cga] proposes a package to construct and display such an arrangement. This package is used to locate intersections and manage point localization. Furthermore, the construction history is stored such that it is possible to identify the curve that generate the detected intersections.

Diffusion curves carry information so we store its associated sorted list of intersection points, noted I, as an additional attribute list. Intersections attributes are sorted according to their parametric coordinates along the curve. As explained in Section 3.4, the depth information is known for all the curve points because the control points depth (i.e. distance to the camera)

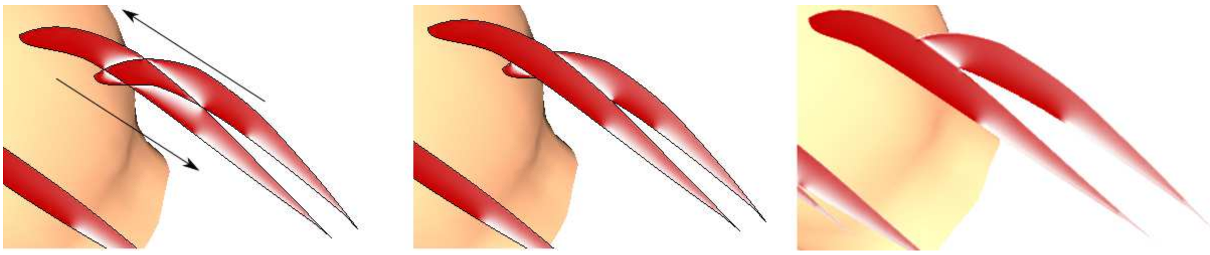

Figure 3.4: Intersections between patches. (left) The arrows represent the research direction. (middle) The parts in between the intersection points are hidden. (right) The root is inside the intersecting curve then the top part of the patch is hidden.

is linearly interpolated along the curve. Therefore, at any given intersection a visibility test is done and the result is stored in both intersecting point attributes along with the intersecting curve identifier *id*. I is then an array of $(id, visibility, t)$.

Dealing with intersection

Dealing with intersection is accounting for hair visibility. To do so, we need to process intersection points starting from the begin of the curve to its end, both are the root because the curve is closed. Patches are represented by closed curves then intersections are paired meaning that each intersecting curves will at least intersect twice.

For each intersection point where the current curve is under, we store its information and look through the sorted set I for an other one with the same *id*. We need to know if the starting point of the curve is inside the intersected hair wisp. If it is, then the top part of the curve is occluded otherwise, (right) Figure 3.4, the part in between the point is hidden, (middle) Figure 3.4.

We need to repeat this process for each pair of intersection points so all the hidden part of the current spline will be accounted for.

3.4 Delimitation of the diffusion

Because we are using Diffusion curves, any color can influence any pixel of the final image. The color diffusion needs to be "stopped" in order to get a closed hair model. In the same way, hair color must not be diffused inside the character body and the other way around. A method needs to be found to solve those issues.

3.4.1 Hair delimitation

Each hair wisp has an outside that casts shadow on the other wisps. But the parts of the wisp that are on the outline of the model should not diffuse shadow color on the background.

The arrangement of a finite collection of geometric objects is the decomposition of the space into connected cells induced by them. The resulting subdivisions into connected regions is called a planar map. As defined in Section 3.2, each hair wisp is represented as a closed curve. Therefore, the union of the planar map connected components represents the global shape (silhouette) of the hair model (see Figure 3.5). Those closed regions are then computed and rendered into a reference image. By extracting its background pixels, a mask delimiting the hair model is obtained and render on top of the diffusion resulting image.

An alternative solution is to delimit the color diffusion by defining a closed Diffusion curve on the silhouette of the hair model. Its inside color would be the one of the hair model and its outside the one of the background. The silhouette can be extracted using an edge detector or joining the induced edges of the only unbounded face of the planar map. Because color are defined on both sides of a Diffusion curve this technique might cause problems if the wisp color is not uniform and only works if all the wisps of the hair model have the same color.

3.4.2 Body rendering

The character body also needs to be rendered without affecting the color diffusion in the hair model. Using a similar technique to the one used for hair delimitation, a mask is computed to render the head model. The rendered image of the 3D scene to create a mask of the body. We compare each of its pixel to the planar map image if the pixel color in the image is different from the back ground color and is not in the delimited closed region of the hair model.

Figure 3.5: Body rendering and hair delimitation using a combination of the reference image and the planar map. (left) Reference image with the projected sampled points, the extruded patch is the closed curve. (middle) Planar map. (right) Rendered wisps.

Body rendering and hair delimitation are done as a last step of the algorithm because the obtained masks are applied on the diffusion result.

In this section, we have presented our rendering pipeline dedicated to animated hair.

Discussion

Limitations

Our model defines the hair wisps in the image space therefore a number of problems are encountered. The wisps patches are defined as a simple extrusion along a central curve thus self intersections can occur for hair with a hight curvature. A closed loop appears inside the wisp and create undesirable color diffusion. Furthermore, curly and wavy hair cannot be represented.

The depth information is only held by the curve then we cannot use classical algorithms, such as the Painter's algorithm, to render the hair model. Computationally intensive operations, that have not yet been fully tested, are needed to account for hair mutual occlusion. This method is then limited to a sparse number of hair strands.

The threshold used for the visibility test can create popping of hair wisps through the animation. More work is needed to ensure temporal coherence.

Future work

An other representation for hair wisp have been explored. The Bézier spline used to define the patches is in fact a Diffusion curve. Its control points are the projection of the guide hair 3D sampled point. Those curves could be used as a border of a wisp with a shadow color on one side and the hair model on the other side. The wisp is defined implicitly representing only one side meaning that the hair color will be diffused until it encounters another hair wisp then no hair wisp width is defined (Figure 1).

This representation raises similar challenges than the proposed approach: hair delimitation, head rendering and intersection. Using patches information can help dealing with those issues with an analogous pipeline. The patches planar map can be used to delimit the hair model and render the character's body using the proposed technique. Then, intersection attributes are also

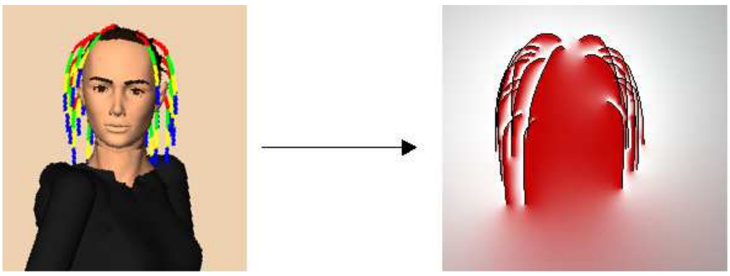

Figure 1: An example of hair rendered as a border of a wisp.

stored along the curve but they are far less numerous because an intersection of two projected curves generates at least two or four patches intersection points.

We could also use the fact that Diffusion curves are vector primitives, therefore suited for key-frame animation through their control points, to give the user control over the animation.

Conclusion

We propose to represent each hair wisp by a patch defined using a closed Diffusion curve that stores the patches' depth information, intersection points list, highlight and shadow colors attributes. The hair model patches are extruded in the image space using linearly interpolated distances to the projected 3D hair strands. The use of Diffusion curves offers smooth-shading and self-shadowing of the hair wisps through the definition of color attribute points. In other words, highlight colors are defined on the *inside* of the curve and shadow colors on the *outside*. Then the color diffusion creates the smooth-shading and shadowing. Hair patches occluded by the head are not rendered to reduce computing time and simplify hair-hair occlusion calculation. No surface is defined for a hair patch therefore hair visibility is determined using the Diffusion curves depth information and the sorted intersection points. The last step of the pipeline is to delimit the hair model. Any color can influence any pixel of the final image thefore we add the background and the body colors after the color diffusion such that they do not affect the hair model. A mask is computed using the arrangement of the closed curves and then applied on the diffusion result to give the final rendering. This work proposes an expressive rendering pipeline dedicated to animated hair. Our solution offers possible answers to the main challenges presented Section 2.3.

Bibliography

[AG99] Anthony A. Apodaca and Larry Gritz. *Advanced RenderMan: Creating CGI for Motion Picture*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[AH03] Ken-ichi Anjyo and Katsuaki Hiramitsu. Stylized highlights for cartoon rendering and animation. *IEEE Comput. Graph. Appl.*, 23(4):54–61, 2003.

[AUK92] Ken.ichi Anjyo, Yoshiaki Usami, and Tsuneya Kurihara. A simple method for extracting the natural beauty of hair. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 111–120, New York, NY, USA, 1992. ACM.

[BAC+06] Florence Bertails, Basile Audoly, Marie-Paule Cani, Bernard Querleux, Frédéric Leroy, and Jean-Luc Lévêque. Super-helices for predicting the dynamics of natural hair. In *ACM Transactions on Graphics (Proceedings of the SIGGRAPH conference)*, August 2006. accepted to Siggraph'06.

[BKCN03] Florence Bertails, Tae-Yong Kim, Marie-Paule Cani, and Ulrich Neumann. Adaptive wisp tree - a multiresolution control structure for simulating dynamic clustering in hair motion. In *ACM-SIGGRAPH/EG Symposium on Computer Animation (SCA)*, July 2003.

[BTM06] Pascal Barla, Joëlle Thollot, and Lee Markosian. X-toon: An extended toon shader. In *International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*. ACM, 2006.

[CF09] E. Cosserat and F.Cosserat. *Thèorie des corps déformables*. Hermann, 1909.

[cga] CGAL, Computational Geometry Algorithms Library. http://www.cgal.org.

[CJDO04] Martin Côté, Pierre-Marc Jodoin, Charles Donohue, and Victor Ostromoukhov. Non-Photorealistic Rendering of Hair for Animated Cartoons. In *Proceedings of GRAPHICON'04*, 2004.

[CJY02] Johnny T. Chang, Jingyi Jin, and Yizhou Yu. A practical model for hair mutual interactions. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 73–80, New York, NY, USA, 2002. ACM.

[DS00] Oliver Deussen and Thomas Strothotte. Computer-generated pen-and-ink illustration of trees. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 13–18, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[Gol97] Dan B. Goldman. Fake fur rendering. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 127–134, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[HMT01] Sunil Hadap and Nadia Magnenat-Thalmann. Modeling dynamic hair as a continuum. *Comput. Graph. Forum*, 20(3), 2001.

[Kim02] Tae-Yong Kim. *Modeling, Rendering, and Animating Human Hair.* PhD thesis, University of Southern California, 2002.

[KK89] J. T. Kajiya and T. L. Kay. Rendering fur with three dimensional textures. *SIGGRAPH Comput. Graph.*, 23(3):271–280, 1989.

[KN01] Tae-Yong Kim and Ulrich Neumann. Opacity shadow maps. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, pages 177–182, London, UK, 2001. Springer-Verlag.

[LTT91] A. M. LeBlanc, R. Turner, and D. Thalmann. Rendering hair using pixel blending and shadow buffers. 2(3):92–97, July–September 1991.

[LV00] Tom Lokovic and Eric Veach. Deep shadow maps. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 385–392, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[MIAI05] X. Mao, S. Isobe, K. Anjyo, and A. Imamiya. Sketchy hairstyles. In *CGI '05: Proceedings of the Computer Graphics International 2005*, pages 142–147, Washington, DC, USA, 2005. IEEE Computer Society.

[MJC+03] Stephen R. Marschner, Henrik Wann Jensen, Mike Cammarano, Steve Worley, and Pat Hanrahan. Light scattering from human hair fibers. *ACM Trans. Graph.*, 22(3):780–791, 2003.

[MKIA04] Xiaoyang Mao, Hiroki Kato, Atsumi Imamiya, and Ken Anjyo. Sketch interface based expressive hairstyle modelling and rendering. *Computer Graphics International Conference*, 0:608–611, 2004.

[MVH02] Ramon Montoya-Vozmediano and Mark Hammel. Stylized flowing hair controlled with nurbs surfaces. In *SIGGRAPH '02: ACM SIGGRAPH 2002 conference abstracts and applications*, pages 254–254, New York, NY, USA, 2002. ACM.

[NT04] Paul Noble and Wen Tang. Modelling and animating cartoon hair with nurbs surfaces. In *CGI '04: Proceedings of the Computer Graphics International*, pages 60–67, Washington, DC, USA, 2004. IEEE Computer Society.

[OBW+08] Alexandrina Orzan, Adrien Bousseau, Holger Winnemöller, Pascal Barla, Joëlle Thollot, and David Salesin. Diffusion curves: A vector representation for smooth-shaded images. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008)*, volume 27, 2008.

[Pai02] D. Pai. Strands: Interactive simulation of thin solids using cosserat models. *Computer Graphics Forum*, 21(3):347–352, 2002.

[PCP01] Eric Plante, Marie-Paule Cani, and Pierre Poulin. A layered wisp model for simulating interactions inside long hair. In *Proceedings of the Eurographic workshop on Computer animation and simulation*, pages 139–148, New York, NY, USA, 2001. Springer-Verlag New York, Inc.

[RER91] Edwin Tripp III Robert E. Rosenblum, Wayne E. Carlson. Simulating the structure and dynamics of human hair: Modelling, rendering and animation. *The Journal of Visualization and Computer Animation*, 2(4):141–148, Mar-Apr 1991.

[SHM06] Jung Shin, Michael Haller, and R. Mukundan. A stylized cartoon hair renderer. In *Advances in Computer Entertainment Technology*, page 64, 2006.

[SLF08] Andrew Selle, Michael Lentine, and Ronald Fedkiw. A mass spring model for hair simulation. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, pages 1–11, New York, NY, USA, 2008. ACM.

[Sou] Helen South. Long hair. http://drawsketch.about.com/library/weekly/aa032303c.htm.

[SYAM05] Eiji Sugisaki, Yizhou Yu, Ken-Ichi Anjyo, and Shigeo Morishima. Simulation-Based Cartoon Hair Animation. In *Proceedings of the 13th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG'05)*, pages 117–122, 2005.

[WBK+07] Kelly Ward, Florence Bertails, Tae-Yong Kim, Stephen R. Marschner, Marie-Paule Cani, and Ming Lin. A survey on hair modeling: Styling, simulation, and rendering. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 13(2):213–34, Mar-Apr 2007. To appear.

[YXWY00] Xue Dong Yang, Zhang Xu, Tao Wang, and Jun Yang. The cluster hair model. *Graph. Models*, 62(2):85–103, 2000.

Abstract

The goal of this work is to design an expressive rendering technique for a realistic animation of hair. This project is a part of an ANR research program for a joint industrial project with two production studios: Neomis Animation and BeeLight, two other INRIA project-teams: Bipop and Evasion and a CNRS lab (Institut Jean Le Rond d'Alembert de l'Université Pierre et Marie Curie). The aim of this project is to provide hair rendering and animating tools for movie making.

Keywords: Hair animation, expressive rendering, physically based simulation.