An Integrated System for Modeling, Animating and Rendering Hair

Agnes Daldegan

Nadia Magnenat Thalmann

MIRALab, University of Geneva, 24, rue du Général-Dufour
CH-1211 Geneva, Switzerland

Tsuneya Kurihara

Central Research Laboratory, Hitachi, Ltd
Kokubunji, Tokyo 185, Japan

Daniel Thalmann

Computer Graphics Lab, Swiss Federal Institute of Technology
CH 1015 Lausanne, Switzerland

Abstract

There are basically four problems to solve in order to produce realistic animated synthetic actors with hair: hair modeling and creation, hair motion, collision detection and hair rendering. This paper describes a complete methodology to solve these basic four problems. We present how hair styles may be designed with our HairStyler module. Then we survey the animation model and emphasize a method of collision processing. Finally, we explain how hair may be rendered using an extension of a standard ray-tracing program. We also show applications of our synthetic actors with various hair styles and different styles of mustaches and beards.

Keywords: hair modeling, animation and rendering, sculpting software, dynamics, shadow buffer

Introduction

In the field of human simulation, hair presents one of the most challenging problems and therefore has been one of the most unsatisfactory aspects of rendered human images to date. The difficulties of processing hair result from the large number and detailed geometries of the individual hairs, the complex interaction of light and shadow among the hairs, and the small scale of one hair's width in comparison to the rendered image. In fact, there are basically four problems to solve in order to produce realistic animated synthetic actors with hair: hair modeling and creation, hair motion, collision detection and hair rendering. Each of the four problems has been addressed by several researchers, but no one has proposed a complete methodology or integrated system to solve the four problems. This paper describes such an approach based on the cooperation of the University of Geneva, the Swiss Federal Institute of Technology in Lausanne and the Hitachi Central Research Laboratory in Tokyo. The first section reviews the four problems and the various solutions proposed by researchers. The second section describes how we model and create hair and hairstyles. The third section explains a method to animate hair with collision detection. The fourth section surveys the rendering method and the last section presents a discussion and future developments

1. A survey of problems and solutions

The modeling problem is one of specification of hundreds of thousands of individual elements: their geometry, their distribution, their shape, their direction.

Hair animation should be based on physics models. However, the animation of individual strands is too time-consuming. Rosenblum et al.[1] propose hair animation method using a mass spring model. Anjyo et al.[2] have proposed methods using one-dimensional projective differential equations and pseudo-force fields. Both methods neglect the effect of collision between hairs for simplicity. Collisions of hair with other objects, such as the head, are simplified using a sphere or ellipsoid. The problem of self-collision of hair is very difficult. However, the collision between hair and other objects is required to generate natural movement of long hair.

The rendering of hair therefore constitutes a considerable anti-aliasing problem in which many individual hairs, reflecting light and casting shadows on each other, contribute to the shading of each pixel.

Several researchers have published methods for rendering human hair, or the more limited problem of rendering fur. The problem itself falls into the category of rendering naturalistic phenomena, and has much in common with the problem of rendering grass and trees, which has been addressed with much success.

Perlin and Hoffert [3] employed volume densities, controlled with pseudo-random functions, to generate soft furlike objects. Perhaps the most impressive rendering of fur to date was achieved by Kajiya and Kay [4] for a teddy bear using a generalization of 3D texturing known as texels. Texels are a type of model intermediate between a texture and a geometry.

Csuri et al. [5] were the first to render fur-like volumes. Each hair was modeled as a single triangle laid out on a surface and rendered using a Z-buffer algorithm for hidden surface removal. Better results were obtained by Gavin Miller [6] [7] who rendered furry animals with images made of explicit hairs. Each hair was modeled with triangles to form a pyramid. Oversampling avoided aliasing. Although the number of hairs was relatively small and their thickness was large, these techniques were nonetheless rather computationally intense. Presumably, it would become impractical when scaled up to a full head of finer human hair. Watanabe and Suenaga [8] modeled human hairs as connected segments of triangular prisms and were able to render a full head of straight human hair in a reasonably short time using a hardware Z-buffer renderer with Gouraud shading. Although the hair model had a realistic number of hairs (more than a million primitives), the illumination model was quite simplistic and apparently no attempt was made to deal with aliasing. As a result, the images are somewhat stiff in appearance.

Perhaps the closest thing to a practical technique for rendering properly anti-aliased hair in the literature to date is the Reeves and Blau's [9] rendering of grass with particle systems. Since the original purpose of the technique [10] was to animate particles in motion, the points were displaced along their paths of motion during one frame to simulate motion blur, yielding three-dimensional line segments. Later, the same technique was used to represent static images with thin filaments, such as grass. As a result, the term "particle" often refers, in fact, to a filament which may have a considerable length, but are usually very thin.

2. Hair modeling

The purpose of the hair modeler is to produce a hair segment file for the animation module. To create the hair style, the program offers, in the special module HairStyler, some interactive facilities. First, the three-dimensional curved cylinder is composed of straight cylindrical segments connected by points. The "in-between" points can be moved in the space to be adjusted, modified, deleted, or added. Just one type of individual hair can be assigned to each triangle in the scalp mesh. However the same hair can be assigned to various triangles. In order to adjust the hair orientation on the curved scalp surface, its direction can be modified by rotating the hair around the normal of its triangle. Details on the HairStyler interface may be found in [11].

After defining each hair format and applying it to the respective triangle, the final style can be defined. The same hair style can have different lengths, by making it grow or shrink using a multiplication factor. A hair style generally has between 100,000 and 150,000 hairs, but this density can be regulated either for

individual triangles or the entire scalp. Initially all the hairs placed in the triangle have the same length, orientation and symmetrical position. To make hair styles look more natural, all these parameters can be changed interactively. A random length, orientation and position can be assigned to each triangle hair set.
The HairStyler was conceived based on polygonal mesh objects modeled by the SurfMan sculpting software[12]. In SurfMan software, synthetic objects in general are remodeled or sculpted from primitives such as spheres and cylinders or even from other complex objects like body parts, in order to generate new synthetic objects or human actors [13].

The HairStyler module maintains a database of information that parameterizes hair growth on each polygon. Each polygon has the following parameters:

- an identifier referring to a 3D curve
- a material identifier for hairs on this polygon
- jitter for the base location of the curve
- orientation of the hairs, called a tangent vector in the world coordinates
- random angle variation applied to the tangent vector
- scaling value applied to the curve
- random value applied to the curve
- density (number of hairs per unit area)
- a seed for the random generator

Polygonal information is unique to a surface and, once created for that surface, cannot be used with a different surface. Furthermore, if the original surface is modified or deformed, the hairs may not follow along as expected. The reason for this is that if a triangle's shape is changed, the number of hairs within it will also change in proportion to the surface area, therefore hair drawing order will change and random values for each hair (jitter, scale, angle) will differ. Also if the shape of a triangle is changed, the tangent vector should be corrected. This can be done by calculating a binormal vector equal to the cross vector of the new triangle normal and the old tangent vector, and calculating a new tangent vector as the cross vector of the new normal and binormal.

Using the SurfMan polygonal mesh objects as pattern surfaces for hair placement, different fur objects can be created using HairStyler. A wide range of hair styles can be generated also from one single original set of few three-dimensional curved cylinder hairs, by varying the parameters of the HairStyler options, mentioned above.

3. Hair animation

To generate natural hair animation, physical simulation must be applied. However, precise simulation including collision response is impractical because of the large number of individual hairs. Therefore, we use simple differential equations of one-dimensional angular momenta as described by Anjyo et al. (1992), and a simplified collision detection method using cylindrical representation[14].

3.1 The dynamic model of hair

A strand of hair is modeled as a series of line segments (Figure 1). In this figure, s_i $(1 \le i \le k)$ is a segment, P_i $(0 \le i \le k)$ is a node, and k is the number of segments in each strand. We consider only the angles and the torques between the segments for simplicity.

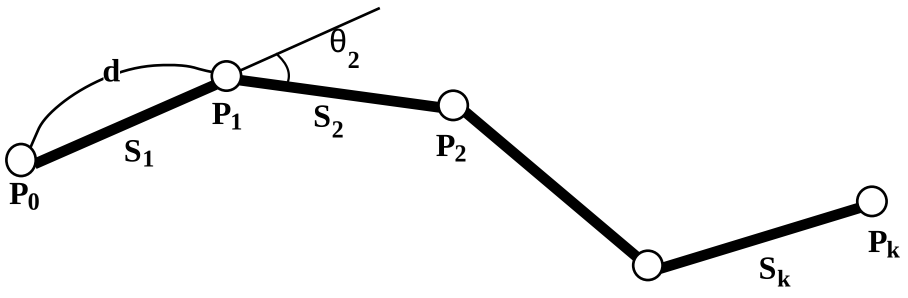

Fig. 1 Dynamic model of a single strand

Taking the polar coordinate system as shown in Figure 2, the variables $\theta_i(t)$ and $\varphi_i(t)$ with time parameter t are governed by the ordinary differential equations:

$$
\begin{aligned}
I_i \frac{d^2\theta_i}{dt^2} + \gamma_i \frac{d\theta_i}{dt} &= M_\theta \\
I_i \frac{d^2\varphi_i}{dt^2} + \gamma_i \frac{d\varphi_i}{dt} &= M_\varphi
\end{aligned}
\tag{1}
$$

where I_i is the moment of inertia of the segment s_i, γ_i is the damping coefficient, M_θ and M_φ are the torque according to θ and φ component respectively.

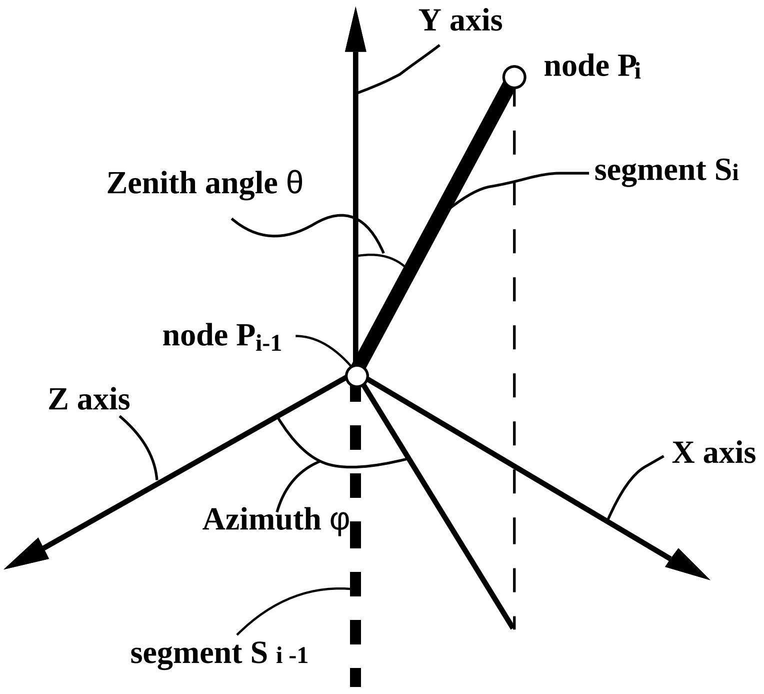

Fig.2. The polar coordinate system for a hair segment

The torque M_θ and M_φ applied to the segment s_i are derived from the hinge effect $M_{\theta spring}$, $M_{\varphi spring}$ between two segments, and external moment $M_{\theta external}$, $M_{\varphi external}$:

$$M_\theta = M_{\theta\, spring} + M_{\theta\, external}$$
$$M_\varphi = M_{\varphi\, spring} + M_{\varphi\, external} \qquad (2)$$

$M_{\theta spring}$ and $M_{\varphi spring}$ are defined as

$$M_{\theta\, spring} = -k_\theta(\theta - \theta_0)$$
$$M_{\varphi\, spring} = -k_\varphi(\varphi - \varphi_0) \qquad (3)$$

where k_θ and k_φ are spring constants, and θ_0 and φ_0 are the initial angles.
External moments are derived from external forces, such as gravity, inertia and wind.

3.2 Collision detection

We use a simplified collision detection method because the number of hairs is very large. This method uses cylindrical representation. First, we create a cylindrical representation of the human body. We divide the human body into several parts including a head and trunk. Each part is described in a cylindrical coordinate system as (r, θ, y), where r is the radius, θ is the azimuth, and y is the height. We create an array of radius on discretized azimuth $i\Delta\theta$ and height $j\Delta y$ $(1 \le i \le m,\ 1 \le j \le n)$ by calculating the intersection between line and object (Figure 3). Along with the radius, we prepare the normal vector on each sample point for calculating collision response.

Fig.3. Cylindrical representation of three dimensional shape

When a cylindrical table is created, collision is easily detected. First, we translate node point **P** into cylindrical coordinates as (r_p, θ_p, y_p). Second, we obtain the corresponding point **Q** of the object with the same azimuth θ_p and height y_p as point **P**. If we assume the object surface smoothness, radius r_q of **Q** can be approximated by linear interpolation using the prepared table. Collision detection is now reduced to comparing the radius part of point **P** and that of point **Q**. If r_q is greater than r_p then point **P** is inside the object. Otherwise point **P** is outside of the object.

When point **P** is inside of the object, point **T** on the surface of the object that is nearest from **P** is calculated and required for collision response. This point **T** is approximated using linear interpolation.

If a hair strand is inside of the body, the reaction constraint method [15] is applied to keep hair outside of the body. Let $\mathbf{F}_{\text{input}}$ be the applied force to node point **P**. Then unconstrained component of $\mathbf{F}_{\text{input}}$ is

$$\text{Funconstrained} = \text{F input} - (\text{F input} \bullet \text{N})\ \text{N}, \qquad (4)$$

where **N** is the normal vector at point **T.** The constrained force to avoid the collision is

$$\mathbf{F}_{\mathbf{constrained}} = -(k\,\mathbf{PT} + c\,\mathbf{V}\bullet\mathbf{N})\,\mathbf{N}, \qquad (5)$$

where **V** is the velocity of point **P**, **T** is the nearest point on the surface from point **P**, *k* is the strength of the constraint and *c* is the damping coefficient.

The output force which is applied to point **P** is a summation of $\mathbf{F}_{\text{unconstrained}}$ and $\mathbf{F}_{\text{constrained}}$.

$$\mathbf{F}_{\text{output}} = \mathbf{F}_{\text{input}} - (\mathbf{F}_{\text{input}}\bullet\mathbf{N})\,\mathbf{N} - (k\,\mathbf{PT} + c\,\mathbf{V}\bullet\mathbf{N})\,\mathbf{N}. \qquad (6)$$

To simulate inelastic collision, we apply reaction constraint only if the input force is not lifting the point **P** away from the surface.

3.3 The wisp model

It is expensive to simulate all hundreds of thousands of strands of hair. When we observe real hairs, many neighboring strands move similarly. Thus, if we simulate the movement of a wisp of strands and not each strand, it reduces the computation time significantly.
The wisp model of strands we have applied is very simple one. We simulate the movement of typical strands and generate other strands by adding random numbers to the origin of the typical strands. Hair styling program can change the density of strands. Thus, we can use low density to define typical strands.

3.4 Animation production process

The process of producing hair animation consists of following steps.

- We translate hair segment data created by the HairStyler module into polar coordinates suited for numerical simulation. These angles are used as initial angles,θ_0 and φ_0 in equation (2). The number of strands is reduced by setting low density. Resulting strands are used as the typical strands in the wisp model.
- We create radius tables in cylindrical representation of the human body. These tables are used for efficient collision detection.
- We specify parameters for numerical simulation, such as moment of inertia, spring constant and external force. We also specify the head movement. Since trial and error is inevitable for generating animation, we start with a hair model with a small number (100) of strands and control parameters interactively. The movement of hair does not change according to the number of the strands because strand-strand interaction is not considered.
- We calculate motion of hair by numerical simulation.
- We generate a full hair model using wisp model from typical strands. We can use the preview program to generate rough shading display using graphics workstations. If the motion is satisfactory, we render using pixel blending and shadow buffer techniques.

The wall clock time of the numerical simulation is typically 20 seconds per frame with 1000 strands where each strand consists of 40 segments on Silicon Graphics Iris Power Series workstation. Therefore, if we use 50 strands, we can interactively control parameters.

3.5 Examples

Figure 4 shows four frames taken from a short film representing a windy scene. In this scene, only the force of wind and gravity are applied. The wind is coming from the front of the face. Note that strands do not penetrate the head or shoulders because of the collision response.

Figure 5 shows five frames taken from a head shaking scene, where the head moves left, right, and finally left again. A little wind is also applied coming from the front of the face.

4. Hair rendering

Rendering an image of hair with our system involves several steps:

- creating a database of hair segments
- creating shadow buffers from all lights
- rendering the hairless objects using all shadow buffers
- composing the hair on the hairless image

In our system, hair rendering is done by raytracing using a modified version of the public domain Rayshade program. An implementation module of the shadow buffer algorithm[16] [17] has been added to Rayshade, based on an earlier version of hair rendering based on pixel blending[18]. It works fine for calculating shadows of normal objects and is also used to figure out the shadows cast by hair.

In the hair style rendering module, the process is step by step. First, the shadow of the scene is calculated for each light source *i*, as well as for the light sources for the hair shadows. The hair shadows are calculated for the object surface and individually for each hair. Finally the hair style is blended into the scene, using all shadow buffers. The result is an image with a three-dimensional realistic hair style rendering where complex shadow interaction and highlight effects can be seen and appreciated. In more detail, the full rendering pipeline may be summarized as follows:

- We take the scene model description and project it onto each light source, creating one scene shadow buffer for each light source.
- We take the hair model and project it onto each light source, creating one hair shadow buffer for each light source. This is done by drawing each hair segment into a Z-buffer based frame buffer and extracting the resulting depth map.
- We compose the depth maps for the scene shadow buffer and the hair shadow buffer, resulting in a single composite shadow buffer for each light source.
- We generate the scene image and its Z-buffer, using the scene model description and the composite shadow buffers as input to the scene renderer, resulting in a fully rendered scene with hair shadows, but no hair.
- We blend the hair segments into the scene image, using the scene's Z-buffer to determine visibility and the composite shadow buffers to determine shadowing, yielding the final image with hair and full shadows. For this blending process, each strand of the hair model is breaking into straight 3D line segments. The intensity H of each of the segment's endpoints is determined using the following hair intensity equation:

  $$H = L_A K_A + \Sigma S_i L_i (K_D \sin(\theta) + K_S \cos^n(\varphi + \theta - \pi)) \quad (7)$$

  where L_i is the light intensity, K_A a is the ambient reflectance coefficient and L_A is the ambient light power incident per unit area. S_i is a shadowing coefficient, obtained by filtering the pixel at the hair strand's depth value against the shadow buffer for the i-th light source.

The raytracer basically has two roles in the hair rendering pipeline. The first is to construct shadow buffers from light sources, and the second is to render hairless objects with full shadowing (coming from the hair and the hairless objects).

There are in fact two ways of creating shadow buffers. The first is by rendering objects with the graphics hardware and reading the contents of the z-buffer. The second is by using the raytracer and rendering the scene from the position of the light, asking it to reduce a z-buffer image of the scene. After the shadow buffers have been created, the hairless objects have been rendered with full shadowing, and a z-buffer from the camera position has been calculated, the hairs may be added to the scene. Now, we need a minimum amount of information:

- hair segment data file
- hair material file
- rendered hairless scene file
- z-buffer of the hairless scene
- a list of buffers to be used for shadow calculations on the hair

The Hair Renderer module offers the possibility of rendering in the same image, various hair styles with their own features. Fig.6 shows an example a synthetic actress with a hairstyle. Fig. 7 to 10 show a variety of mustaches and beards designed using our HairStyler program. These different combinations were generated by changing the scalar bias, the density, the angular bias and the jittering factor.

For rendering fur objects, another important and difficult feature to be achieved in computer graphics, we require a three-dimensional texture. In HairStyler, depending on the density and hair length chosen, the fur rendered object is in fact, a three dimensional texture surface made by concentrated segments of lines, reflecting rays of light in different orientations, and shadows and highlight effects.

5. Discussions and Future Research

The four difficult tasks of synthesizing natural-looking hair for synthetic actors and creatively making their hair styles is achieved by our program. Originally conceived to rendering realistic human hair, the software also offers to the user some facilities for creating all kinds of hair styles for any model or simple objects: teddy bears, fur coats, as well as synthetic actor mustache, beard and furry skin.

Fig.6. Synthetic actress with hairstyle

In computer graphics and animation, research of natural effects for synthetic images are essentially motivated by the realism that algorithms based on specific methods of physics try to attain. These techniques are getting more and more sophisticated, in order to achieve at new qualities of the visual product.

However, the realism of synthetic images should be not evaluated only from the technological point of view. Because, as in any kind of visual communication, the aesthetic aspect of the image has an important role of attracting public attention before transmitting the message. Similarly, in computer graphics, a "hightech" visual domain, the realism reached by sophisticated research in hardware and software is, in fact, a combination of technical and aesthetic factors.

In synthetic human performance animation, this special aspect of technique and aesthetics is also noticed. Research attempting to reach the correspondent realism of physical and biological human features also faces the problem of aesthetics. In synthetic human animation, traditional research is basically oriented towards the following subjects: physical character of the human body and personal physical features; physical-motricity related to skin deformations during the personalized walk and grasping; facial expressions derived from speech and emotion; behavioral movement based on artificial vision. In this research range, all plans of creating a "look" in terms of clothes (fashion clothes and their animation) and hair style (hair style model, rendering and animation) are equally addressed. Hence, the natural aesthetic aspect of the human being considered as scientific research subject requires not only meeting the challenge of synthesis and animation of cloths or hair style, but also allowing the variety and originality of the different physical effects and their appearance observed in nature.

Originally, our program was conceived to reach principally good and realistic effects in hair rendering process. Its first satisfying results had led to more efforts in the elaboration of the hair style, mustache and beard fashion designing and its diversified applications in other surfaces, rather than a synthetic human scalp or face, as clothes and actor's limbs. For this purpose, some basic modifications were made in the HairStyler module to facilitate the artistic design work and the aesthetic purpose of this new version was satisfactorily reached. Although, as it frequently happens, the creative characteristic of the aesthetic experience, in many times overcomes the technical resources of the means itself [19]. This can also happen with computer graphics techniques, which represent a constant challenge for hardware and software researchers to improve their technology and algorithms. However this important remark must be understood as the aesthetic experiment can effectively contribute to software technology evolution and development since it poses new problems which require new approaches.

For the development of our program performance concerning this question of achieving realistic, natural hair, two different aspects for improvement of the technique research can now be adopted: a powerful interface modeler for the specialized user, and a compatibility with realistic skin texture mapping for synthetic actor's natural human appearance.

This first proposition, not developed as such yet, can be a natural way to provide the integration between research in technique and aesthetic with a sophisticated interface tool for the hair maker specialized public. One efficient idea can be the emulation of real tools as brushes and scissors. Those virtual tools can provide as performing constraints the physical aspects of hair gravity, (already implemented by the others important researches as described in the introduction), hair thickness, resistance and specific type, in order to permit the direct and realistic interaction with user's creativity and synthetic actor's hair.

The second one, is a concentrated effort in order to integrate a facial texture mapping research already achieved [20], and hair for a synthetic actor's natural look and animation.
These both propositions are currently under development.

Acknowledgments

The authors would like to thank Dr. Geoff Wyvill for his useful remarks and discussion in hair rendering and hair interface problems. They are also grateful to Guy Moreillon and Pascal Wolino for their work in the integration of the software. They also thank Hans-Martin Werner for revising the English text. Agnes Daldegan is grateful to CNPq-Brazil. The research was sponsored by the "Fonds National pour la Recherche Scientifique."

References

[1] Rosenblum RE, Carlson WE, Tripp III E (1991) Simulating the Structure and Dynamics of Human Hair: Modelling, Rendering and Animation, The Journal of Visualization and Computer Animation,Vol. 2, No. 4 , pp. 141-148.

[2]. Anjyo K, Usami Y, Kurihara T (1992) A Simple Method for Extracting the Natural Beauty of Hair, Computer Graphics, Vol. 26, No. 2, pp. 111-120.

[3]. Perlin K, Hoffert (1989) Hypertexture, Computer Graphics 23(3), 1989, pp. 253-262.

[4]. Kajiya JT, Kay TL (1989) Rendering Fur with Three Dimensional Textures, Computer Graphics Vol. 23, No. 3, pp. 271-280.

[5]. Kajiya JT, Kay TL (1989) Rendering Fur with Three Dimensional Textures, Computer Graphics Vol. 23, No. 3, pp. 271-280.

[6]. Miller GSP (1988a) From Wire-Frame to Furry Animals, Proc. Graphics Interface, pp. 138-146.

[7]. Miller GSP (1988b) The Motion Dynamics of Snakes and Worms, Computer Graphics, 22(4), pp. 169-178.

[8]. Watanabe Y, Suenaga Y (1989) Drawing Human Hair Using Wisp Model, Proc. Computer Graphics International '89, pp. 691-700.

[9]. Reeves WT, Blau R (1985) Approximate and Probabilistic Algorithm for Shading and Rendering Structured Particle Systems, Computer Graphics 19(3), pp. 313-322.

[10]. Reeves WT (1983) Particle Systems - A Technique for Modeling a Class of Fuzzy Objects, Computer Graphics 17(3), pp. 359-376.

[11]. Magnenat Thalmann N, Daldegan A (1993) Creating Virtual Fur and Hair Styles For Synthetic Actors. In: Communicating with Virtual Worlds, Springer-Verlag, Tokyo, pp.358-370.

[12]. LeBlanc A, Kalra P, Magnenat Thalmann N, Thalmann D (1991) Sculpting With the Ball & Mouse Metaphor. Proc. Graphics Interface '91, Calgary, Canada.

[13]. Paouri A, Magnenat Thalmann N, Thalmann D (1991) Creating Realistic Three-Dimensional Human Shape Characters for Computer-Generated Films. Proc. Computer Animation'91, Springer, Tokyo, pp.89-99.

[14]. Kurihara T, Anjyo K, Thalmann D (1993) Hair Animation with Collision Detection. In: Models and Techniques in Computer Animation, Springer-Verlag, Tokyo, pp.128-138.

[15]. Platt JC, Barr AH (1988) Constraint Methods for Flexible Models, Computer Graphics, Vol. 22, No. 4, pp. 279-288.

[16]. Williams L (1978) Casting Curved Shadows on Curved Surfaces"Computer Graphics, Vol.12, No3, pp. 270-274.

[17]. Reeves WT, Salesin DH and Cook R.L (1987) Rendering Antialiased Shadows with Depth Maps, Computer Graphics 21(4), pp. 283-291.

[18]. LeBlanc A, Turner R, Thalmann D (1991b) Rendering Hair using Pixel Blending and Shadow Buffers. The Journal of Visualization and Computer Animation 2, 3, pp. 92-97

[19]. Nadin M (1991) Science and Beauty: Aesthetic Structuring of Knowledge, Leonardo, vol. 24, no. 1 pp 67-72.

[20]. Kalra P, Mangili A, Magnenat Thalmann N, Thalmann D (1992) Simulation of Facial Muscle Actions Based on Rational Free Form Deformations, Proc. Eurographics '92, North Holland, pp.59-69.

Color Plates

Fig. 4. A windy scene,

Fig.5. A head shaking scene,

Fig. 7. Synthetic actor with eyebrows, eyeslashes and beard

Fig.8. Synthetic actor with hairstyle and mustache

Fig. 9. Synthetic actor with the same hairstyle and but a different mustache

Fig. 10. Synthetic Ctor with beard and mustache