A Rotor Platform Assisted System for 3D Hairstyles

Chai-Ying Lee, Wei-Ru Chen, Eugenia Leu*, Ming Ouhyoung

Dept. of Computer Science and Information Engineering,
National Taiwan University, Taiwan
{carollee, wrchen, ming}@cmlab.csie.ntu.edu.tw
*b6506019@csie.ntu.edu.tw

Figure 1: By interactively editing hair, various realistic hairstyles can be created.

ABSTRACT

This paper presents an interactive system for creating natural looking 3D hairstyles, by which users can cut, comb and perm the hair model and generate realistic hair images. The system contains three stages: modeling, styling and rendering. In the first phase, the system produces a physical hair model using a cantilever beam simulation with collision detection. Then the styling phase is a hair editing process, performing styling operations to change the lengths, positions and curvatures of hair strands. Seven operations are developed for styling. A special hardware rotor platform is developed to aid the hair cut process, so that manipulation by two hands that simulate the real styling is made possible. Users can interact with tools and create various convincing hairstyles. Our system together with the rotor platform increases the ease of hairstyling (60% time saving) and can render natural hair images with shadow and back-lighting effects.

Keywords Hairstyle, virtual hair modeling, hair styling, collision detection, 3D characters.

1 INTRODUCTION

The computer graphic community constantly holds a special interest in virtual humans. How to create realistic-looking hair is one of the most challenging topics. Three typical problems need to be solved for hair simulation: hair modeling, hair dynamics, and hair rendering. Many research efforts have been devoted to each of these issues. Hair simulation is difficult for three reasons:

1. Huge number of hair strands.
2. Complex light effects among hairs.
3. The fine volume of each individual hair.

Various models which have been developed aiming at photo-realistic 3D hair. However, CG methods nowadays can only produce limited hairdos, and will take a lot of styling time if traditional tools are used.

We want to propose a sound basis for creating realistic looking hairstyles. Our goal is an interactive system, by which users can cut, comb and perm the hair model and output realistic and convincing hair images. In this paper, we only deal with static hair and leave out hair dynamics and animation.

1.1 PREVIOUS WORK

Previous hair simulation work can be divided into two categories: explicit and implicit hair models. Explicit hair models simulate the physical properties of hair and consider the shape and dynamics of each hair strand or a wisp. The basic concepts of implicit hair models are inspired from some techniques such as fluid flow or volumetric rendering. These approaches model hair shape by mathematical functions and offer no geometric definition of individual hairs.

Thalmann et al presented a survey paper [19], which discuss the advantages and disadvantages of many approaches. In the explicit model category, Anjyo [2] simplified each hair as a linearly connected point set and simulated hair dynamics using the cantilever beam deformation technique. Watanable and Suenaga [20, 21] introduce a wisp-based model, defining a wisp as a group of hairs. Daldegan [6] proposed an integrated system dealing with four problems of the explicit hair model: hair modeling, hair motion, collision detection and hair rendering.

The second category is implicit hair modeling. Some researchers used the volumetric approach [17, 10]. Perlin [17] et al introduced "Hypertexture" and Kajiya [10] extended it to tile onto complex geometry. Kajiya used this approach and rendered a realistic furry teddy bear. But the volumetric approach can usually deal with short hair like fur. Hadap et al [8] proposed a new styling method based on fluid flow, which models hair shapes as streamlines of fluid flow.

2 SYSTEM OVERVIEW

Our system focuses on two goals: the first is the ability to create various convincing hairstyles easily. The second goal is to render photo-real hair images.

The process of making a hairstyle in the physical world provides a reference frame for designing our system pipeline. The whole process can be divided into three stages: hair modeling, hair styling, and hair rendering.

In an integrated system [6], hair styling is part of hair modeling and lacks of flexibility. There are many advantages to let hair styling be an independent phase. Firstly, it makes styling process independent from modeling and rendering. This brings more flexibility to each stage. Secondly, styling methods can be developed freely. There are many possible methods that can be innovated and applied.

Based on the system's purposes, there are some requirements we have to satisfy. As an interactive system, the efficiency of the styling stage must be close to real-time. And a user-friendly user interface (UI) should be taken into consideration, thus bring in the introduction of a rotor-platform based interactive system.

3 HAIR MODELING

The purpose of hair modeling is to produce a simple and ordinary hairstyle. Based on this hair model, we can apply styling operations in the second stage. For a 3D human head model, the modeling procedures need to be carried out just once.

For the convenience of description, let us clear up some terminology here: a hair strand means an individual hair. One strand consists of a point set which represents the 3D path of one hair. We denote the points as hair nodes. In a 3D head model, the polygons with hair attached are hair polygons.

To get a smooth hair curve, we adopt Cardinal splines to represent hair strands in the rendering stage.

3.1 PHYSICAL HAIR MODELS

After loading a 3D head model, we first define where the hair area is. This is done manually with the assistance of OpenGL selection functionality. The hair area is divided into ten regions.

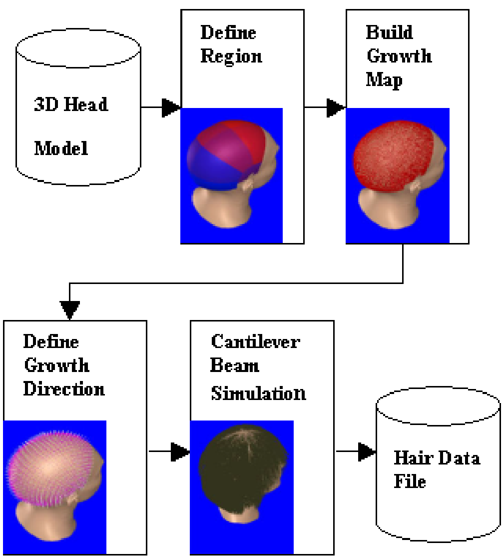


Figure 2: The hair modeling pipeline.

Next, we define the growth map on the hair area. The system defines pores on polygons based on two parameters: the pore density and the wisp size. The result forms the growth map. Then, the hair growth directions are assigned to give an initial orientation to each hair polygon. The modeling procedure pipeline is shown in Fig. 2.

3.2 CANTILEVER BEAM SIMULATION

The natural shape of a hair strand is a downward 3D curve. We adopt Anjyo's method [2] to simulate the bending of the strand under the influence of gravity. Hair strands are treated as cantilever beams, each of which consists of several segments with equal interval lengths. The details of the cantilever beam simulation are described in [2].

3.3 THE WISP MODEL

The hair wisp concept was introduced by Watanabe and Suenaga [21]. It is based on a naturally occurring phenomenon: hairs tend to form clumps due to adhesive/cohesive forces, especially when hair gets wet or becomes oily. But the wisp model not only produces effects such as hair clumping; the model can also reduce the number of hair strands that need to be processed. Without any reduction, styling will be difficult and time consuming. The model requires three parameters as input, and the influence of parameters is shown in Fig. 3.

Parameters:

1: Wisp size: the number of hair strands per wisp.

2: Clump-degree: the distance the hair strand moves from its original position toward key hair / radius of wisp.

3: Clump-rate: the distance from pore to where hair starts to clump / the length of wisp.

Each pore belongs to one wisp, and each wisp has a key hair in the center. Key hairs represent the overall shape of the hairs in the wisp. The hair model will thus not be controlled by editing hair strands directly, but by key hairs.

(a) (b)

Figure 3: The wisp with wisp size = 64 and clump- degree –1 (a) and 1 (b).

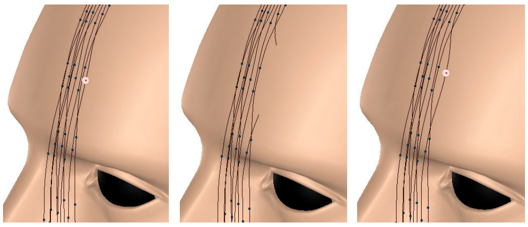
Because a braid is made of three bundles of hairs, we introduce the cluster concept into our model. A cluster actually indicates one of the three bundles of a braid. It may contain many wisps, and we should select one wisp's key hair to be the cluster's key hair. In a certain cluster, we make the key hairs of all wisps follow the direction of the cluster's key hair.

3.4 COLLISION DETECTION

Collision detection and response are important considerations in hair dynamics simulation. Even in a static situation, hair-body collision avoidance should also be supported to prevent hair strands from intersecting into the head model.

We devise an approach using body contour values. In our approach, we project the outmost coordinates of body in five directions: Up, Left, Right, Front, and Back. Then we record these depth values in five contour buffers. To check if collisions occur, hair node coordinates are projected onto planes formed by axes, (*x-z* plane for example,) then we compare the values mentioned above with depths in the *x-z* contour buffer. If the hair node is inside the head model, it will be moved outside by adding an outward vector. Because the body boundary can be totally described accurately in five buffers, the collision position is quite precise. (Fig. 4)

This approach is practical when body remains static because contour buffers need to be recalculated each time the body moves. In our cast, the program only calculates these buffers once, then checks collision detection several times both in the modeling stage and the styling stage.



(a) (b) (c)

Figure 4: (a) Original position; (b) One hair node is moved inside the model; (c) After collision detection, the hair node is moved out.

4 HAIR STYLING

A person's hairstyle is a consequence of many physical factors and his own hair properties. If we want to create a hairstyle realistically, the styling stage should contain artificial operations to change the position of hair nodes. For the purpose, two issues have to be considered: 1. The algorithms are suitable for styling. 2. Styling operations need to be efficient and easy to control.

To achieve our purpose, we assume there is a pseudo force field in the 3D space. When we perform a styling operation, there will be an influence vector added to hair nodes' coordinates.

To take a user-friendly UI into consideration, we find it far easier to use this system with a rotor platform than by a 2D mouse. With the assistance of 3D trackers and a rotor platform, users can make hairdos in the 3D way like real hairdressers.

4.1 STYLING OPERATIONS

The styling operations change the lengths, orientations and curvatures of hair strands. In this stage, we implement seven kinds of methods: scissors, a comb, curler, a hair dryer, a circular brush, fluffing and braiding. To create various hairstyles, users can edit the hair model with arbitrary combinations of these operations. The styling operations change hair nodes by producing a pseudo force field, which gives influence vectors at every point in the 3D space. The pseudo force field is not a really physical force and is defined as a reasonable model to imitate the hairstyling process. For each selected hair node, we calculate its displacement according to the pseudo force field. We can adjust parameters to control the magnitude of the force field.

4.1.1 SCISSORS

The scissors cut the lengths of selected hair strands. Hair nodes crossing the scissors will be removed. New hair nodes will be attached to the hair node point set in the intersection positions of hair strands and scissors.

Figure 5: The scissors.

4.1.2 COMB

The purpose of the comb tool is to sequentially alter positions of hair nodes. The influence vector is a constant vector. While combing hairs, the tool gives hair nodes a constant force field in one direction. κ in Eq. 1 represents the magnitude.

$$\vec{V}_{comb}(\overline{p}) = \kappa \vec{v} \quad (1)$$

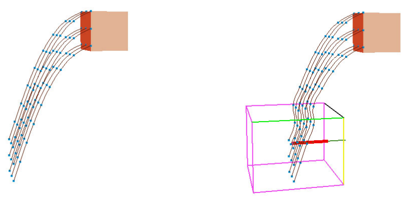


Figure 6: The comb.

4.1.3 CURLERS

A hairdresser uses curlers, curlpapers and curling pins to perm hair in order to make it wavy or curly. The same shapes can be drawn in the CG world by mathematical methods. By the pseudo code below, we present how to "perm" one straight hair strand into a curly shape.

direction 1, direction 2: two orthogonal directions, by which we increase the angle progressively.

radius: radius of circular shapes, larger values create bigger waves.

progressive angle: the angle difference between two continuous hair nodes.

```
angle = 0
For each hair node of the key hair{
        Position(hair node) = Position(hair node) +
        cos(angle) * radius *direction 1 +
        sin(angle)* radius * direction 2
        angle = angle + progressive angle
         Next node}
```

Figure 7: The curlers.

4.1.4 HAIR DRYER

To dry hair, we produce a force field like the wind force from a hair dryer. In Eq. 2, d is the distance from the hair dryer to a hair node, and κ is the magnitude. The function of a hair dryer is similar to the comb: to move hair nodes, but in a different pattern.

$$\vec{V}_{dryer}(\overline{p}) = \kappa \frac{\hat{d}}{4\pi |r^2|} \quad (2)$$

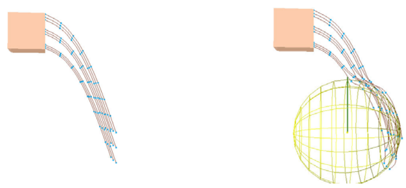


Figure 8: The hair dryer.

4.1.5 CIRCULAR BRUSH

The circular brush is also a tool for creating curly hair. It creates a vortical force field to modify hair. Using cylindrical coordinates, the influence vector of a circular brush is:

$$V_r = 0, \quad V_\theta = \frac{b}{2\pi r}, \quad V_z = 0 \quad (3)$$

Figure 9: The circular brush.

Figure 11: Braiding.

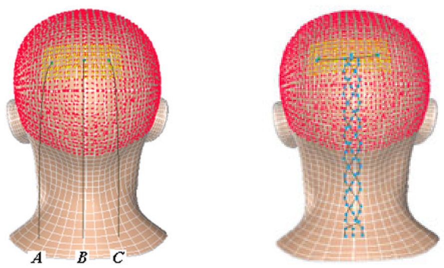
4.1.6 FLUFFING

We implemented a unique tool for disturbing the hair strands. If hairs are too neat and artificial, we can add some noise vectors which can enhance the realism of hairstyles. The noise vectors are bound by maximum and minimum values.

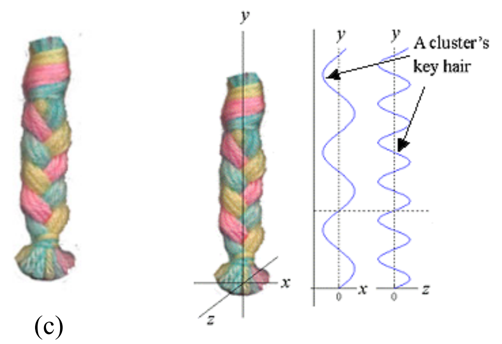
Figure 10: Fluffing.

4.1.7 BRAIDING

To make braids, as Fig. 11 shows, we need to pick 3 regions and take each region as a cluster. Then we choose a key hair of a wisp from each cluster (usually the central wisp of the cluster) as the key hair of the whole cluster.

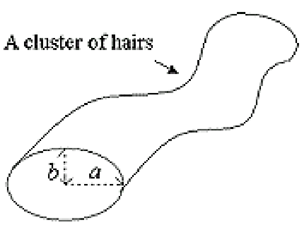


(a) (b)



(c)

(d)



(e)

In Fig. 11(e), a is half of the length of an ellipse's long axis, and b is half of the length of the short axis. By our observation, $a \approx 2 \cdot b$.

By Fig. 11, we know that the cross section's shape of a cluster is similar to an ellipse. By observing of Fig. 11(d) and Fig. 11(e), we can simulate the three clusters of a braid by making use of a sine function. The pseudo code is:

dir0 = horizontal; // *x*-axis direction
dir1 = vertical; // *y*-axis direction
dir2 = CrossProduct(dir0,dir1);
dir1_q = *a* * sin(angle); // angle: the angle difference dir2_q = *b* * sin(angle2);
// angle2: the angle difference
hairs->wisp[i].Path[j] = hairs->wisp[i].Path[j] + dir1_q*dir0 + dir2_q*dir2;

Besides, when observing the side of the woolen braid (Fig. 11(c)), each color repeats showing up every three time. Therefore, we define the initial angle of these three key hairs of each cluster as 0^o, -120^o and $+120^o$. Then we assign key hair *B* in Fig. 11(a) is the major axis for braiding, and properly pull *A* and *C* to *B*'s position.

We use the clump concept of the wisp model to make each cluster of hairs more realistic.

4.2 A ROTOR-PLATFORM BASED 3D STYLING TOOL

Although the system provides a flexible editing environment, some hairstyles are still difficult to imitate because of the limitation of the UI controlled by a mouse. The difficulty emerges at trying to use a 2D mouse to control a 3D model and trying to move the head model and the hairstyling tool at the same time.

To overcome the problem, we adopt Polhemus FastTrak as our sensors. Two sensors are added to the system. One sensor is attached to a plastic head model on a rotor platform to represent the head model of the system. And the other one is used for handling the movement of the editing tool. To add convenience to the selecting and cutting operation, a mouse is tied to a sensor providing three useful buttons. After initialization, the plastic head can be mapped to the head model of the system hence the head model can be controlled by turning the rotor platform. Besides, the position of the styling tools such as scissors or a comb will be located by moving the sensor around the plastic head. And clicking the buttons on the sensor can apply the operations such as cutting or combing to the selected hair wisps. By adopting these hardware tools, hairstyling can be done rapidly and handily.

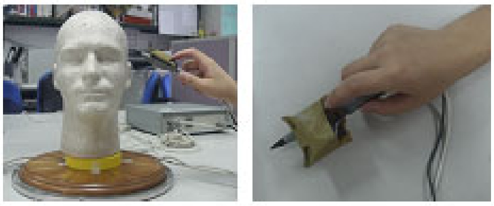


Figure 12: The 3D styling tool needs two sensors: one sensor is attached to a plastic head model, and the other one is used for controlling the editing tool.

Figure 13: Clicking the buttons on the sensor can enable the scissors operation and cut selected hair.

5 HAIR RENDERING

Base on the existing Z-buffer algorithm and the shadow algorithm, the rendering pipeline is derived:

1. Create the shadow map and the density map for each light source.
2. Raster non-hair objects with shadow at the output image resolution.
3. Raster hair segments using a modified illumination model with shadow and back-lighting effects. The raster hair image resolution is (filter size) × (output image size).
4. Filter the hair image and blend with the non-hair image.

The illumination model of hair rendering refers to Goldman's research [7]. The colors are calculated for end points of hair segments; the in-between colors are obtained by linear interpolation.

5.1 SHADOW MAP AND DENSITY MAP

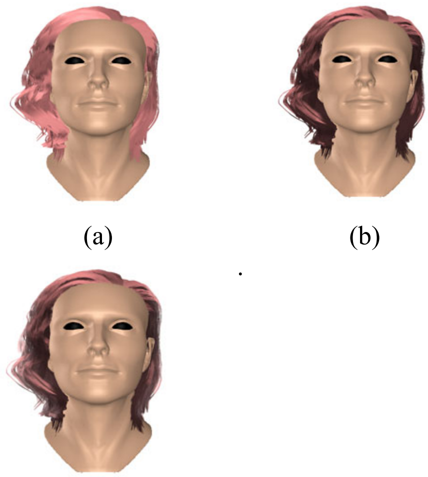
Among existent shadow algorithms, shadow map is adequate and practical in the case of hair rendering. This technique has been developed in previous researches [12, 6]. For creating a "halo" effect for hair as thought it were back-lit, the density maps are required from the point of view of light sources. Density map proposed by Yang et al [23] is implemented in this system.

5.2 ANTI-ALIASING AND IMAGE BLENDING

We use the supersampling method to reduce the jagged lines in the hair image. For the purpose, we raster images at resolution higher than the output resolution, and then average them down. At the same procedure, hair and non-hair images are composed appropriately. If the filter size is 4×4, system produce a shadow buffer, a density buffer and a hair image in (4×output width) × (4×output height) resolution. This image will be filtered and re-sampled at output resolution. At the same time, according to the pixel coverage of hair and non-hair, we blend two images.

5.3 LIGHT EFFECTS FOR HAIR RENDERING

Fig. 14 shows the hair with or without shadow/ back-lighting effects. In this figure we can clearly see the influence on apparent realism.



(a) (b)

(c)

Figure 14: (a) Hair without shadow and back-lighting effects; (b) Hair with shadow effect but without back-lighting effect; (c) Hair with shadow and back-lighting effects.

6 RESULTS

The system is implemented in C++ language on PC with PⅢ 800 CPU and 128 MB RAM.

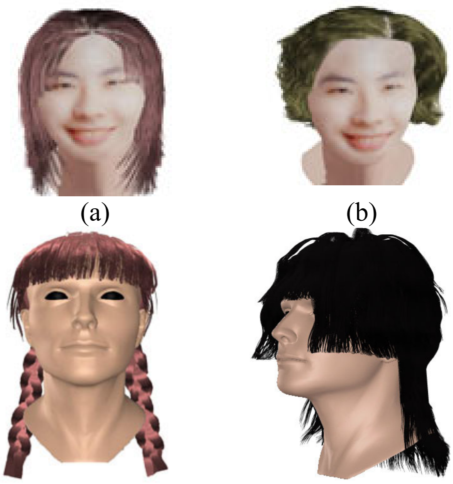
Table 1: The rendering time of hair with shadow and back-lighting effects in 512×512 resolution.

Filter size	2×2	4×4
82571 hairs	75 sec	120 sec
144360 hairs	95 sec	140 sec

The time spent on styling operations depends on the complexity of hairstyles. In general, the styling stage needs half of an hour. To compare to using a mouse, users with the assistance of a rotor platform can save about 60% time in the styling phrase. Although the styling operations are still limited, a variety of hairstyle images can be created and rendered by our system. Fig. 1, Fig. 15 and Fig. 16 illustrate the effectiveness of the proposed methodology.

(a) (b)

Figure 15: (a) The original photo of a male; (b) The 3D head model with CG hairstyles.



(a) (b) (c) (d)

Figure 16: Resulting images of four hairstyles.

7 CONCLUSION

We have proposed an interactive system for hair styling. Users can edit hairstyles interactively with a rotor platform and render natural hair images with shadow and back-lighting effects.

The contributions of this paper are:

1. The combination of a rotor platform and UIs improve the ease and flexibility of hairstyling. Traditionally, making hairstyles is a very tedious job. 60% time can be saved in our experience.
2. Seven styling operations are developed, including braiding. Braiding is a rare function in existing modeling tools.

8 FUTURE WORK

Though our system cannot yet produce all types of hairstyles, the limitations can be solved by developing new styling algorithms. In the future, the system can be enhanced in two aspects:

1. Other hairstyle effects and tools can be developed. There are still some tools that need to be implemented to make the system more powerful. Some effects such as clip are also desirable.
2. Enhance the rendering realism. The translucent hair will change its appearance under lighting, and we will apply the subsurface light transport model proposed by Jensen et al [9] for realistic hair rendering.

REFERENCES

[1] Anderson,J.D: Fundamentals of aerodynamics. McGraw-Hill, Inc., 1991.

[2] Anjyo.K, Usami.Y, Kurihara.T: A simple method for extracting the natural beauty of hair, Computer Graphics (Proceedings of SIGGRAPH 92), vol. 26, no. 2, pp. 111–120, July 1992.

[3] Berney,J: Stuart Little: a tale of fur, costumes, performance, and integration: breathing real life into a digital character, SIGGRAPH 2000 Course Note.

[4] Bruderlin,A: A method to generate wet and broken-up animal fur, Pacific Graphics ’99, October 1999.

[5] Chen,L.H, Saeyor,S, Dohi,H, Ishizuka,M: A system of 3d hair style synthesis based on the wisp model, The Visual Computer, vol. 15, no. 4, pp. 159–170, 1999.

[6] Daldegan,A, Magnenat-Thalmann,N, Kurihara,T, Thalmann,D.: An integrated system for modeling, animating and rendering hair, Computer Graphics Forum (Eurographics ’93), vol 12, no 3, pp 211–221, 1993.

[7] Goldman,B.D: Fake fur rendering, Proceedings of SIGGRAPH 97, pp. 127–134, August 1997.

[8] Hadap,S and Magnenat-Thalmann,N: Interactive hair styler based on fluid flow, Eurographics Workshop on Computer Animation and Simulation'2000, Springer-Verlag, pp. 87-99, August 2000.

[9] H. W. Jensen, S. R. Marschner, M. Levoy, P. Hanrahan, “A Practical Model for

Subsurface Light Transport", Proceeding of SIGGRAPH 2001, pp. 511-518, Los Angeles, August 2001.

[10] Kajiya,J.T, Kay,T.L: Rendering fur with three dimensional textures, Computer Graphics (Proceedings of SIGGRAPH 89), vol. 23, no. 3, pp. 271–280, July 1989.

[11] Kaswell,E.R: Textile fibers, yarns and fabrics, Reinhold Publishing Corporation , 1953, pp 52.

[12] LeBlanc,A, Turner,R, Thalmann,D: Rendering hair using pixel blending and shadow buffer, Journal of Visualization and Computer Animation, vol. 2, pp. 92–97, 1991.

[13] Lewis,J.P: Algorithms for solid noise synthesis, Computer Graphics (Proceedings of SIGGRAPH 89), vol. 23, no. 3, pp. 263–270, July 1989.

[14] Ling,L, Damodaran,M, Gay,R.K.L: Aerodynamic force models for animating cloth motion in air flow, The Visual Computer, vol. 12, no. 2, pp. 84–104,1996.

[15] Lokovic,T, Veach.E: Deep shadow map, SIGGRAPH 2000 proceeding, pp385-392. July 2000.

[16] Neyret,F: Modeling, animating, and rendering complex scenes using volumetric textures, IEEE Transactions on Visualization and Computer Graphics, vol. 4, no. 1, pp. 55–70, January-March 1998.

[17] Perlin.K, Hoffert,E.M: Hypertexture, Computer Graphics (Proceedings of SIGGRAPH 89), vol. 23, no. 3, pp. 253–262, July 1989.

[18] Rosenblum,R, Carlson,W, Tripp,E: Simulating the structure and dynamics of human hair: modeling, rendering and animation, Journal of Visualzation and Computer Animation, vol. 2, pp. 141–148, June 1991.

[19] Thalmann, M.S. Hadap, P. Kalra: Sate of the art in hair simulation, International Workshop on Human Modeling and Animation, Korea Computer Graphics Society, pp. 3-9, June 2000.

[20] Watanabe,Y, Suenaga,Y: Drawing human hair using wisp model, in Proceedings of Computer Graphics International'89, pp. 691–700, Springer Verlag, 1989.

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

[22] Wejchert,J, Haumann,D: Animation aerodynamics, Computer Graphics, Proceedings of SIGGRAPH 91, vol. 25, no. 4, pp. 19–22, July 1991.

[23] Yan,X.D, Xu,Z, Yang,J, Wang,T: The cluster hair model, Journal of Graphics Models and Image Processing, in press 1999.

[24] Yang,T.J, Ouhyoung,M: Rendering hair with back-lighting, Proceeding of CAD/Graphics' 97 Dec. 1997, pp. 291-296.