The Visual Computer

A system of 3D hair style synthesis based on the wisp model

Lieu-Hen Chen[1, 2], Santi Saeyor[1], Hiroshi Dohi[1], Mitsuru Ishizuka[1]

[1] Department of Information and Communication Engineering, School of Engineering, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
E-mail: {chen,santi,dohi,ishizuka}@miv.t.u-tokyo.ac.jp
[2] Department of Computer Science and Information Engineering, National Chi-Nan University, 1, University RD, Puli, Nan-Tou, Taiwan 545, R.O.C.
E-mail: lhchen@csie.ncnu.edu.tw

Hair plays an important role in the image of the human face. However, it is still difficult and time consuming to synthesize realistic 3D CG hair images. In this paper, we present an integrated system for 3D computer graphics (CG) hair image synthesis based on a trigonal prism wisp model. By applying the trigonal prism wisp model, 2D hair distribution maps and other CG techniques to our system, we can successfully and efficiently synthesize realistic CG hair style images. A 3D hair style editing tool (HET) is incorporated into the system to support the creation of 3D hair styles. We have also integrated this system with other research results to construct a system for 3D CG character synthesis.

Key words: 3D CG hair style – Hair image synthesis – Wisp model – CG human characters – Anthropomorphic agent

Correspondence to: L.-H. Chen

1 Introduction

Hair plays an important role in human face imaging. Hair acts as a "medium" to express characteristics, conditions, and feelings [1, 2]. When the computer graphics (CG) character nods or shakes his (or her) head to express "Yes" or "No", the motion of hair will affect the realism of the image. The CG characters or faces have been used recently in advanced interfaces with multimedia functions, such as in anthropomorphic interfaces. However, it is still difficult and time consuming to synthesize realistic 3D CG hair images for these characters.

The difficulty of hair image synthesis is due to the inherent properties of hair, such as the extremely huge number of hair strands. This makes hair image synthesis computationally very expensive [3]. In addition, one hair strand is very thin compared with the pixel size, and hair strands within a wisp are locally parallel. These two properties cause an aliasing problem in rendering hair images. Most important of all, there are huge variations in hair styles, which are determined by extremely complex interactions of many factors. These factors include gravity, friction, static electricity, strand-to-strand and strand-to-head interactions, articles such as hair pins, hair bands, mousse, hair grease, and so on. Thus, it is extremely difficult to generate complicated hair styles with only a physical formula, some mathematical formulas and some parameters. All these properties of hair make hair image synthesis a very challenging task.

The previous approaches to hair image synthesis can be divided into two categories: the explicit model [4–9] and the volume density model [10–13], according to the methods used. The explicit model is very intuitive. The modeling and rendering target of such an approach is each individual hair strand. The rendering target, hair, is treated in a microscopic way. The particle model for hair synthesis is classified in this category. In contrast, the volume density model defines the 3D space distribution function(s) of the entire hair style, and then applies a ray-tracing technique to calculate the color and illumination on the silhouette surface of hair style. This kind of approach views hair styles macroscopically. The approach of the volume density model is more abstract than that of the explicit model. It is not the purpose of this paper to judge which method is superior. However, considering the needs of presenting enormous vari-

The Visual Computer (1999) 15:159–170

ations of hair styles and animating hair motion, we would prefer the explicit model for its potential flexibility.

In this paper, we present an integrated system for the synthesis of 3D CG hair images based on a trigonal prism wisp model. This modeling method, which we use for our system, basically belongs to the category of the explicit models, though it is more simplified and organized to handle the macroscopic properties of hair styles. By applying the trigonal prism wisp model, 2D hair distribution maps, and other CG techniques to our system, we can successfully and efficiently synthesize realistic CG hair-style images. A 3D hair style editing tool (HET) is incorporated into the system to support efficient 3D hair style creation.

2 Basic concepts of our system

We first introduce the five basic design concepts for our system:

1. *Reality.* We must be able to synthesize hair images very realistically.
2. *Low cost.* We must be able to synthesize realistic hair images in a much shorter time than with previous methods or systems.
3. *High flexability.* Using one and the same hair model, we must be able to create various hair styles with long or short hair, straight or curly hair, and so on.
4. *Ability to animate hair styles.* The ability to introduce some kind of physical model and formula to emulate the complicated phenomenon of hair is required.
5. *Practicality.* We want to provide a way to generate and design various 3D hair styles easily.

It is quite reasonable and intuitive to consider these five concepts as the basic elements that make a satisfying system for CG hair style synthesis. We have designed our system with these concepts. The modeling method, as the core of the system, is explained in Sect. 3. Some techniques and issues that we consider necessary for "realistic" hair image synthesis are discussed in Sect. 4. A hair style design and generating tool, the hair style editing tool (HET), is introduced in Sect. 5. The integration of system and some results of hair style synthesis are discussed in Sect. 6.

3 Trigonal prism wisp model and 2D hair distribution map

3.1 Trigonal prism wisp model

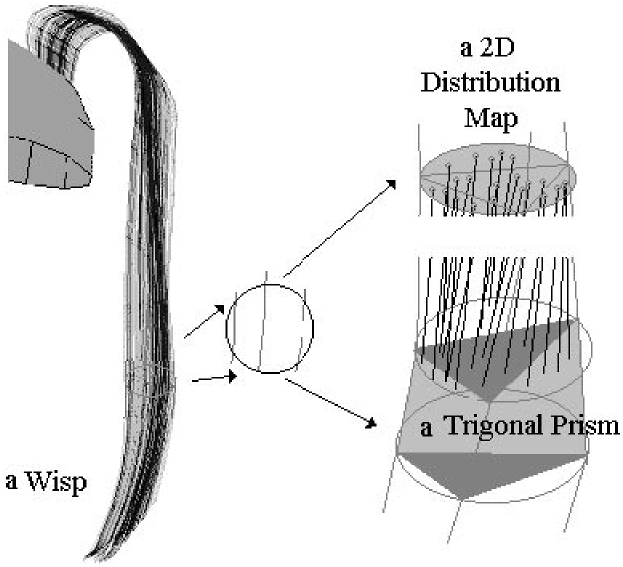
We define our hair model, the trigonal prism wisp model, as follows. First, all the hair is divided into several groups of wisps, according to the position on the skull. Each wisp is constructed by continuous trigonal prisms, as shown in Fig. 1. The trigonal prisms within one wisp are linked by three 3D B-spline curves. The numbers of control points may differ, basically according to the length of the strand of hair.

Using curves to implement hair is a very intuitive idea. It enables our system to create the physical properties of hair easily. Furthermore, because hair strands within a wisp are grouped, we can successfully reduce the computational time. This modeling method keeps the hair style data simple and organized. Thus, it is very flexible for the various hair styles.

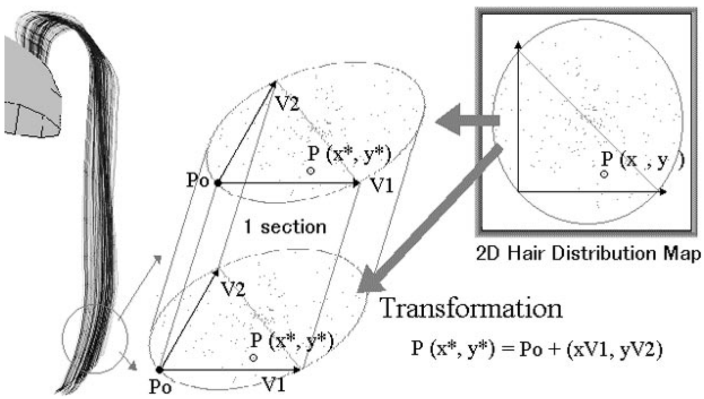
What has to be noted here is the cross-section of the wisp's silhouette: the "actual" image of one wisp is determined by its 2D distribution map, which we will explain later in this section. The trigonal construction of our trigonal prism wisp model does not affect the real shape of synthesized wisps. The cross-section of wisp's silhouette does not need to be a triangle. Its shape is controlled by the 2D hair distribution maps that we assign to the wisp.

The term "wisp", which is used in the title of our modeling method, refers to a group of hair strands. It is a very familiar word to people who study synthesizing hair images. However, it is not really precisely defined. In this paper, we will use the term "wisp" as the smallest element of a hair style.

Within a wisp, there can be 1 to 255 hair strands, which are quite parallel to each other. The 3D orientation and shape of one wisp is defined by three 3D B-spline curves as shown in Fig. 1. Two wisps can share their edges, which we call control lines, with each other. For us, the wisp is the smallest unit used for calculations of 3D calibration, motion emulation, and illumination.



1



2

3

Fig. 1. Trigonal prism wisp model

Fig. 2. The relationship between a wisp and its 2D hair distribution map

Fig. 3. Two hair style images with the same wisp model, but using different link methods (black hair)

3.2 Two-dimensional hair distribution map

We use 2D arrays to define the distribution of hair strands on the cross-section of one wisp, as shown in Fig. 2. Through the continuous trigonal prisms of one wisp, these 2D points are projected onto screen according to each pair of vectors on the edges of the two triangles. The position of each hair strand within a wisp is actually determined in this 2D way. It is noted that this 2D hair distribution map can be defined to have hair distribution outside the triangle cross-section of the triangle prism. This allows us to remove the discontinuity of hair distribution among wisps. However, we need only calculate the three control lines that we use to control the trigonal prisms. Thus, the 2D hair distribution maps contribute to reduce the computational cost of rendering hair.

There is another important issue: the application of the 2D distribution maps to our system for the control of the shape of a wisp's silhouette. For example, if the hair on the 2D map is randomly distributed within a triangle or a circle, the basic shape of the synthesized wisp will be a curved trigonal prism or a curved cylinder, respectively. By controlling the shape and distribution density of the 2D distribution maps, we can create a natural and rich effect on 3D hair forms.

Depending on the way we link these 2D distribution maps, we can also create some interesting visual effects. As shown in Fig. 3, the two different wisp images are created from the same wisp model. The left image is created by linking maps with straight lines, and the right one is linked by sine curves. Using straight lines makes this linkage procedure simple. However, using curves, such as sine waves, instead of straight lines to link the 2D distribution maps can create some interesting visual effects. Hair images synthesized in this way will have an extremely wavy form, though we use the same three control lines and one simple wisp structrue.

4 Rendering hair, shadowing hair, and handling artificial objects

By applying the modeling method discussed in the previous section, we can establish the model of the hair. The next step is to efficiently render hair images. To do this, first, we need to calculate the illumination on a hair strand.

4.1 Rendering hair

To calculate the illumination of one strand of hair, C_h, we use the following formula [7]:

$$C_h = L_a K_a + \Sigma L_i [K_d \cos\theta + K_s \sin^n(\theta + \phi - \pi)],$$

where the subscripts a, d, and s stand for the ambient, diffuse, and specular components, respectively. The Ls are the lightness of the light sources, the Ks are constants, and θ and ϕ are angles as defined in Fig. 4. For different colors of hair, we need to adjust the balance of these parameters. Also, producing the visual effects for the various properties of hair will need more experiment and adjustment. For example, for dry and black hair, we reduce the weight of K_s and K_a, but emphasize K_d. This lighting model still has its limitations, however, such as the lack of a back light visual effect.

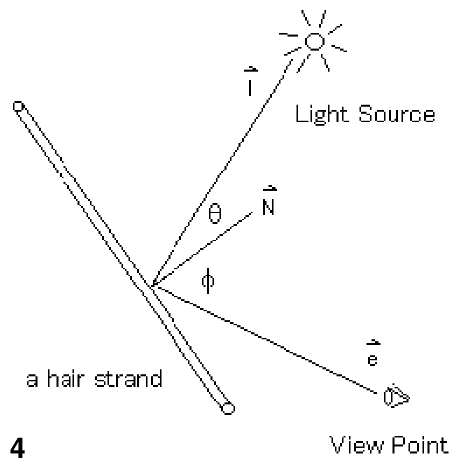
No matter what kind of lighting model is applied to the system, these time-consuming calculation procedures are only performed along the three control lines of each wisp. The illumination of each strand within one wisp is then determined by interpolating of the previous calculated results of the three control lines. With this interpolation process, we can save a large amount of time without loosing the realism of the hair. The result of each wisp is then pixel-blended with the image of other wisps and background to create the transparent visual effect of hair and to remedy the aliasing problem.

4.2 Shadowing hair

Without taking shadow effects of hair into consideration, the synthesized image will lack realism somehow. To add shadow effects efficiently to a hair image, we modify the traditional shadow Z-buffering method. The modified shadowing method, which we call the shadow mask method, is explained next.

The shadow mask method is basically a two-pass process, just as its parent, the traditional shadow Z-buffer method is. As shown in Fig. 5, the first pass of these two methods are very similar. All

4



5

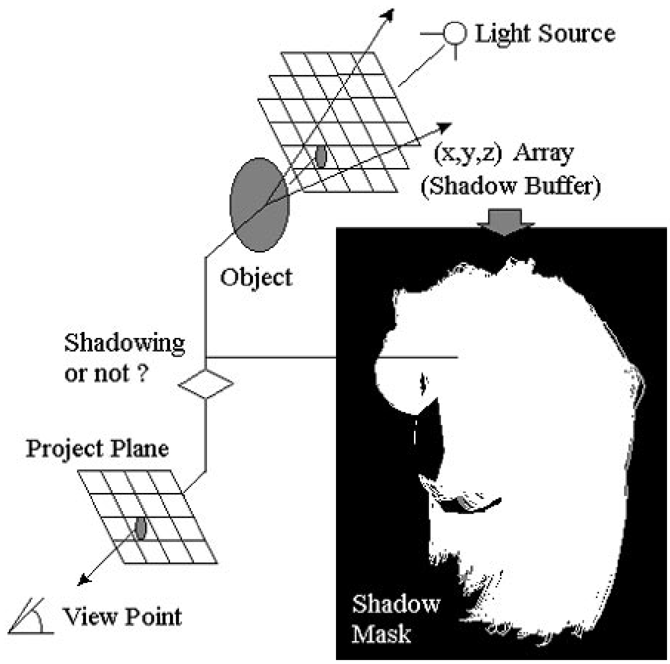
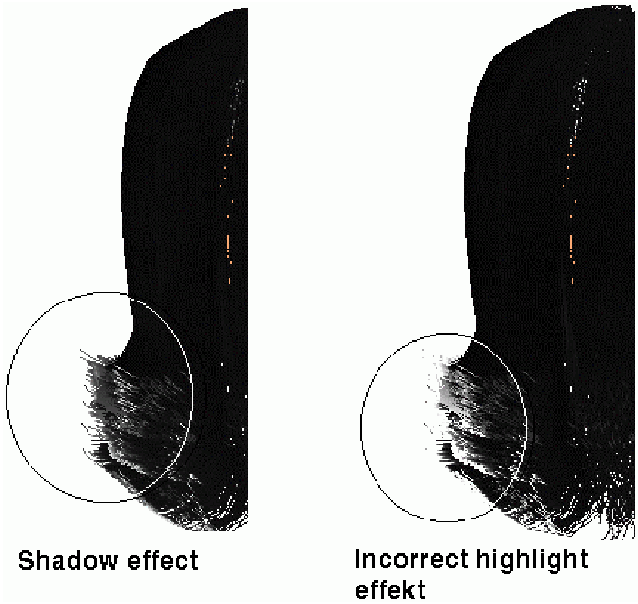


Fig. 4. Lighting on the hair model

Fig. 5. The shadow mask method

wisps are projected to the direction of the light source to check whether they are in the shadow area or not. At this step, the difference between the new and old methods is that we keep the x, y, z coordinates instead of Z-buffer information. Thus, the result of the first pass of our method is a 2D array that contains the 3D coordinate information of all points that are "visible to the light source" on the surface of the hair silhouette. The size of this 2D array and the radius of each point or ball, will determine the quality of the shadowing effects. As a result of our experiments, we keep this array size at the same resolution as that of the viewing window.

The second pass of our method is quite intuitive. By assigning an effective radius to every point, we treat these points as balls in 3D space. The union of these balls will then form a masklike area



6

7

Fig. 6. Hair style images with and without shadow effects (black hair)

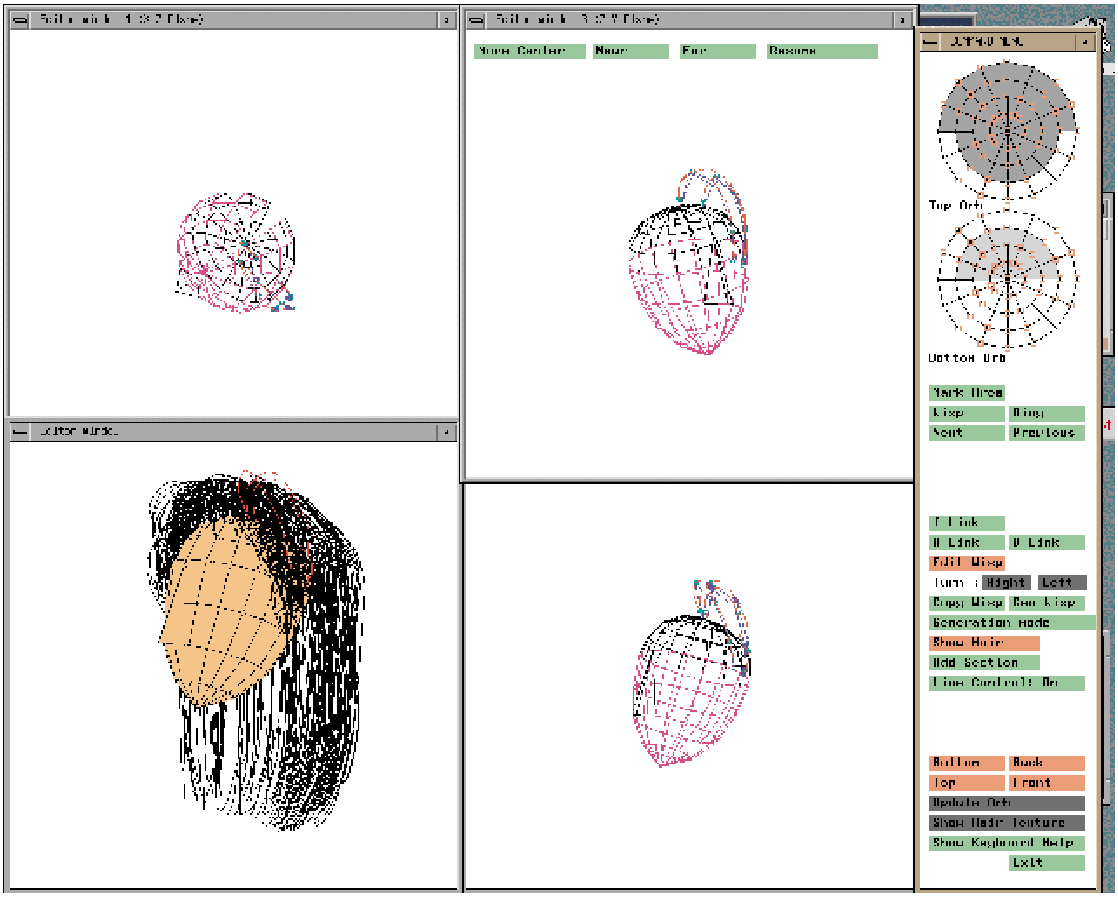
Fig. 7. Ponytail hairstyle (red hair)

that is "visible" to the light source; in other words, it will be outside the dark area. This is the reason why we call this method the "shadow mask". To determine whether a pixel should be shadowed or not, at the second rendering step, we need only to check whether the Z-buffering value of the rendering target is within the Z-range of the related pixel on the shadow mask. Figure 6 illustrates the rendering effect of this showing.

As mentioned before, the number of hair strands is extremely huge. If we apply the traditional shadowing methods to a target that consists of lots of very small rendering objects, such as hair strands, the cost will be proportional to the complexity of the objects. Therefore, adding shadow effects in the traditional way is too expensive for us. Using the shadow mask method, the complexity of second rendering pass is basically proportional to the resolution of the 2D array. Thus, this modified shadowing method helps us to create shadow effects for hair in a more efficient manner.

4.3 Handling artificial objects

Artificial objects, such as hair pins, hair bands, ponytails, and braids, are important components



10

8

9

Fig. 8. Display of the hair style editing tool (HET)

Fig. 9. Hair style – sample 1

Fig. 10. Hair style – sample 2

that cannot be ignored in handling the various kinds of hair styles. With our hair model, adding a hair band or ponytail to a hair style is not so difficult. What we need to do is just give some wisps a certain restriction. These wisps are required to pass through a region of 3D space. For example, we can get a ponytail if we let the control lines of a group of wisps pass through a little circle behind the CG character's head. A hair style with a ponytail is shown in Fig. 7. Handling a braid is a little tricky. A braid is heavier and stiffer than a single hair strand or wisp. It basically consists three wisps. A repeated loop changes the positions of these three wisps. (However, they do not really have to be three individual wisps. They may be just three branches of one wisp. They may also consist of several wisps separately.)

5 The hair style editing tool (HET)

To make our system really practical, we need lots of 3D data for various hair styles. However, the procedure to create these 3D hair styles is a very time-consuming task. The designer's art sense and technique are also required.

We tried to make the design procedure easier. For this purpose, we have developed a hair style editing tool (HET) to help users design and generate 3D hair styles.

As shown in Fig. 8, HET consists of five windows. The command window on the right side contains most of the command buttons. The view window located in the lower left portion is used to view the result in 3D space. We use the other three windows to edit the 3D position and orientation of the wisps with a mouse. Users can change their view angle in these windows whenever they like.

Even though we have successfully simplified the hair data structure by grouping hair into wisps, it still takes lot of effort to position all the wisps in 3D space. For example, we can use 256 wisps to define a hair style which has 256*256=65 536 hair strands within it. To design this hair style, we need to input and edit 3*256=768 3D curves. It will take hours to use the mouse for pointing out all the necessary points for these B-spline curves.

To resolve this problem, we group the wisps according to their position on the skull. These groups look like quarter rings on the skull. We divide the skull into eight layers. Each layer consists of four quarter rings. We assume that the shape of the wisps within the same quarter ring are quite similar. Thus, after a designer has designed one or two control lines on a quarter ring, it is reasonable for HET to automatically generate all the other control lines on the same quarter ring, using the line (or lines) that has already been done as an example. The hair style shown in Fig. 9 is generated by inputting only 14 curves; the other control lines are automatically generated by HET.

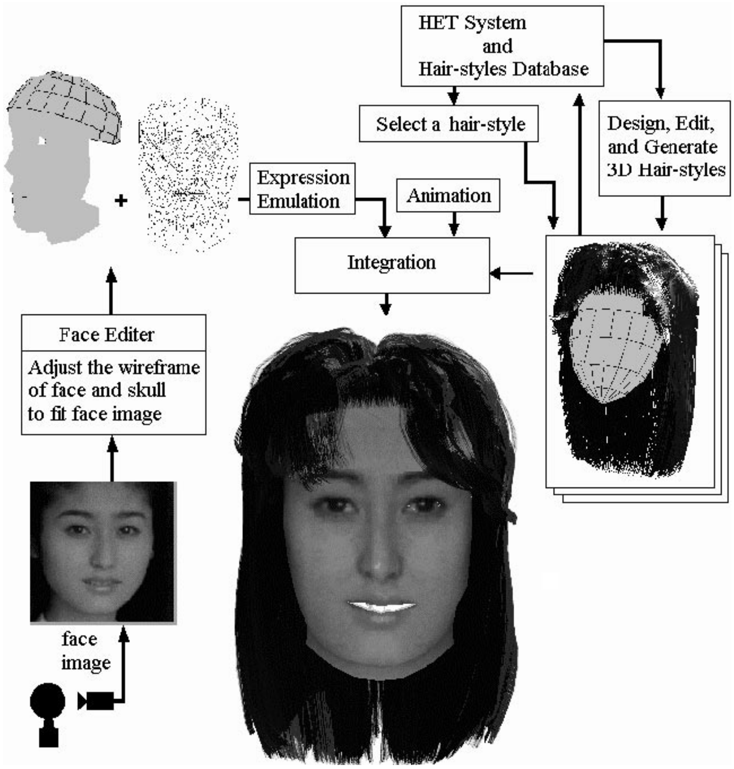
This tool helps the user to roughly design a hair style with only 10–20 3D B-spline curves, and also provides various functions for editing the details of a hair style and the ability of exchanging and sharing parts between different hair styles. One hair style that is a little more complicated than the one in Fig. 9 is shown in Fig. 10. HET also allows users to create various visual effects by changing or blending hair colors for a hair style, and by assigning wavy effects to certain wisps. The time needed to design and edit the wisp models was in total 10 min for Fig. 9, and 30 min for Fig. 10. The rendering time for these CG images is about 1 frame/s on a SGI Onyx workstation. (For convenience in development and experiment, no hardware rendering function, such as line drawing, coordinate translation, Z-buffering, etc., is used in the system at present.)

6 Integration of the system

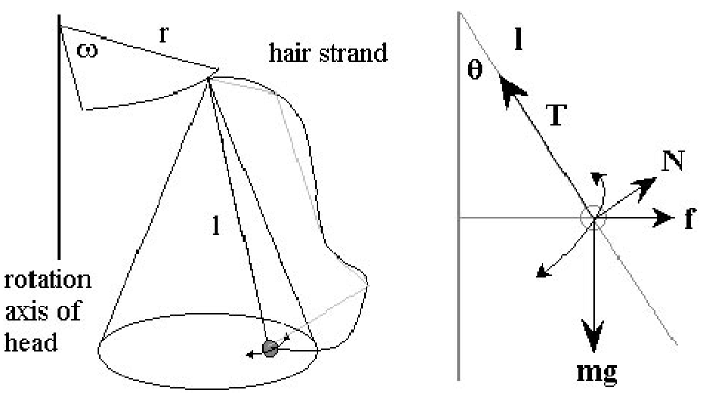
We have integrated this system with a face editing tool to combine users' face images that may be caught from a video camera or a digital camera, as illustrated in Fig. 11. With this integrated system, we can combine a realistic 3D face image with any hair style stored in a database.

As shown in Fig. 11, a skull is attached to the wire frame of a face like a hat. It can be adjusted to fit the wire frame. This flexibility is necessary because the head size of each individual human character is usually different.

As already mentioned, the motion of hair is too complicated to analyze and regenerate completely [14]. It may be caused by an internal force (e.g., head gesture) or an external force (e.g., wind) . We have focused our attention on the case of internal force so far. We have emulated two types of hair motion that is caused when the CG character



11



12

Fig. 11. Integration of a hair style image and a face image

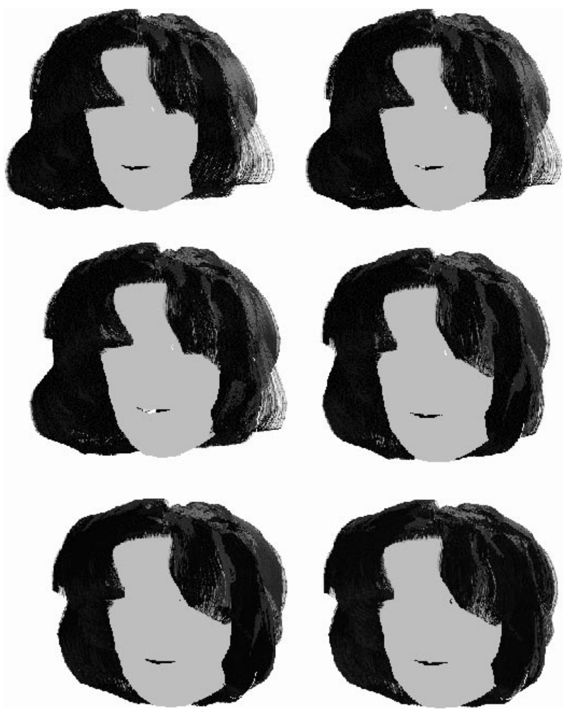
Fig. 12. Pendulum model

nods "yes" or shakes his/her head "no". The motion is achieved by simulating hair as a pendulum resting on a cone as shown in Fig. 12. When a character's head shakes, it gives the pendulum an angular momentum. This angular momentum may or may not be big enough to lift the pendulum from the slope. It also makes the pendulum oscillate along the slope of the cone. The emulation result is shown in Fig. 13. The pendulum model is an extremely simple model and is only suitable for emulating certain motion patterns. To simulate more complicated and general hair motion, more research effort is needed.

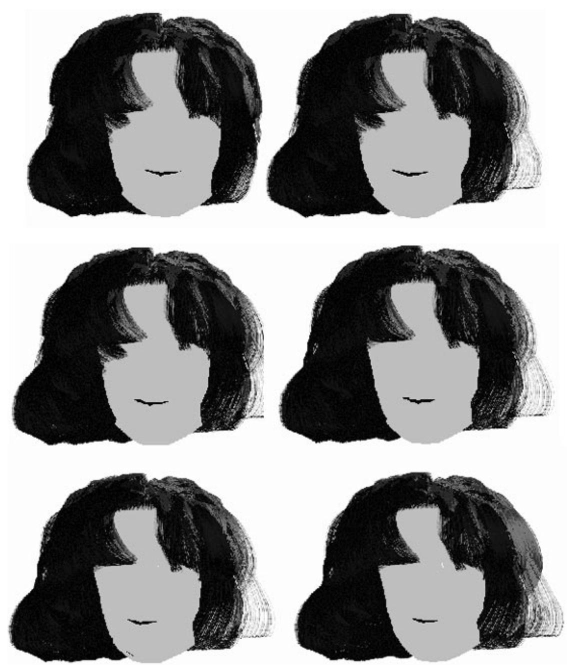
We also propose a system of 3D character synthesis as shown in Fig. 14, in which the system of hair style synthesis serves as one of the basic modules.

a

Fig. 13a–c. Animation (black hair)



b



c

In this system, not only CG hair styles, but also face expression emulation, speech synthesis, and artificial intelligence (AI) components to generate motions based on a scenario script are to be included. The integration of all these functions will make an intelligent life like agent with a realistic figure available; this agent will serve as one important style of advanced interface close to daily face-to-face communication.

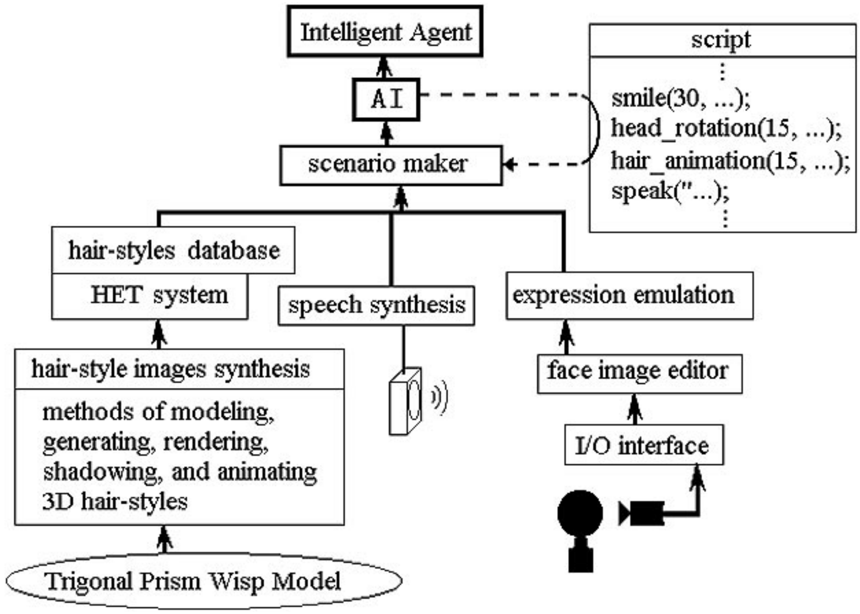


Fig. 14. Diagramm for the system of 3D characters synthesis

7 Conclusion

We have proposed a new hair modeling method, i.e., a trigonal prism wisp model with 2D hair distribution maps, and developed a prototype system to synthesize and animate various kinds of 3D hair styles. This system is integrated to combine face images with the hair styles. Further work is expected to improve the quality of hair images, and to achieve better system performance. There is a plan to introduce the concepts of virtual reality into the next version of HET to allow users to "directly" edit a hair style by hand in a virtual space. The development of the scenario maker will help in isolating the animation module and the AI module. This independence will make our system of 3D CG character synthesis, serving as a platform to incorporate with various AI modules through a standard interface, a scenario maker. Though there are still lots of research issues left for future work in our entire project (Fig. 14), our system has been proven to be an efficient and practical system for creating 3D hair styles, synthesizing realistic hair images, animating hair motion, and manipulating various hair styles.

References

Magnenat-Thalmann N, Thalmann D (1991) Complex models for animating synthetic actors. IEEE Comput Graph Appl 32–44

Magnenat-Thalmann N, Thalmann D (1990) Synthetic actors in computer-generated films. In: (eds) Models and techniques in computer animation, Springer, Berlin Heidelberg New York

Miller GSP (1988) From wire frames to furry animals. Proceedings of Graphics Interface'88, pp 138–146

Reeves WT (1983) Particle systems – a technique for modeling a class of fuzzy objects. Comput Graph 17:359–376

Reeves WT (1985) Approximate and probabilistic algorithms for shading and rendering structured particle systems. Proceedings of Siggraph'85 19:313–322

Watanabe Y, Suenaga Y (1989) Drawing human hair using the wisp model. Proceedings of Computer Graphics International, pp 691–700

LeBlanc AM, Turner R, Thalmann D (1991) Rendering hair using pixel blending and shadow buffers. J Visualization Comput Anim 2:92–97

Watanabe Y, Suenaga Y (1992) A trigonal prism-based method for hair image generation. IEEE Comput Graph Appl 47–53

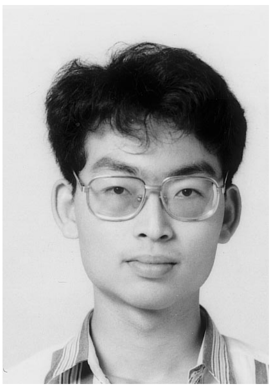
Anjyo K, Usami Y, Kurihara T (1992) A simple method for extracting the natural beauty of hair. Comput Graph 26:111–120

Kajiya JT, Kay TL (1989) Rendering fur with three-dimentional textures. Comput Graph 23:271–280

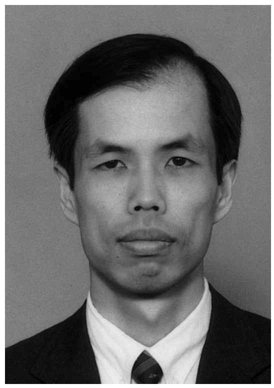
Kajiya JT (1985) Anistropic reflection models. Comput Graph 19:15–21

Yamana T, Suenaga Y (1987) A method of hair representation using anisotropic reflection. IECEJ Technical Report PRU87-3 (in Japanese), pp 15–20
Perlin K, Hoffert E (1989) Hypertexture. Comput Graph 23: 253–262
Rosenblum RE, Carlson WE, Tripp III E (1991) Simulating the structure and dynamics of human hair: modeling, rendering and animation. J Visualization Comput Anim 2:141–148

SANTI SAEYOR is a PhD student at the Department of Information and Communication Engineering at the University of Tokyo, Japan. He received his BE in Electronic Engineering from King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, in 1993, and his ME in Electrical Engineering from the University of Tokyo, in 1996. His current research interests are in the areas of autonomous information agents, mobileagents, and Internet technology.



LIEU-HEN CHEN is currently an Assistant Professor of Computer Science and Information Engineering at the National Chi-Nan University, Taiwan. His current research interests include interactive 3D graphics and virtual reality. He received a BS in Computer Science and Information Engineering from the National Taiwan University in 1989, and his MS and PhD degrees in Electrical and Electronic Engineering from the University of Tokyo, in 1994 and 1998, respectively.



HIROSHI DOHI is a Research Associate at the Department of Information and Communication Engineering of the University of Tokyo, Japan. He received his BS and MS degrees in Electrical Engineering from KEIO University, Japan, in 1985 and 1987, respectively. His current research interests are the Anthropomorphic Intelligent Agent Interface and Internet-based multimodal systems. He ia a member of the ACM and IPS Japan.

MITSURU ISHIZUKA is a Professor of the Department of Information and Communication Engineering of the University of Tokyo. Previously, he worked at the NTT Yokosuka Lab. and the Institute of Industrial Science, University of Tokyo. He received his BS, MS, and PhD degrees in Electronic Engineering from the University of Tokyo. His research interests are in the areas of artificial intelligence, hypothetical reasoning mechanisms, multimodal human interfaces, and intelligent WWW information space. He is a member of the IEEE, AAAI, IEICE Japan, IPS Japan, and Japanese Society for AI.