

**AN ENHANCE FRAMEWORK ON HAIR MODELING
AND REAL-TIME ANIMATION**

LIANG WENQI

NATIONAL UNIVERSITY OF SINGAPORE
2003

**AN ENHANCE FRAMEWORK ON HAIR MODELING
AND REAL-TIME ANIMATION**

LIANG WENQI

A THESIS SUBMITTED
FOR THE DEGREE OF MASTER OF SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF COMPUTING
NATIONAL UNIVERSITY OF SINGAPORE

2003

Acknowledgements

First of all, I would like to thank Dr. Huang Zhiyong. He has been continuously providing me invaluable advice and guidance throughout the course of my study. It is just impossible for me to complete this thesis without his sharing of ideas and expertise in this area.

I would also like to thank School of Computing, which gives me the opportunity and provides me all kinds of facilities that make my thesis possible.

Table of Contents

| | |
|-------------------------------------|----|
| 1. Introduction | 1 |
| 2. Previous Works | 5 |
| 3. U-Shape Hair Strip | 9 |
| 4. Animation | 23 |
| 5. Collision Detection and Response | 31 |
| 6. Hair Modeling | 38 |
| 7. Conclusion | 44 |
| Reference | 45 |

Summary

Maximizing visual effect is a major problem in real-time animation. One work was proposed on real-time hair animation based on 2D representation and texture mapping [Koh00, Koh01]. Due to its 2D nature, it lacks of the volumetric effect as the real hair. This thesis presents a technique to solve the problem. The hair is still modeled in 2D strip. However, each visible strip is warped into U-shape for rendering after tessellation. Alpha and texture mappings are then applied to the polygon meshes of the U-shape. Note that the U-shape is not explicitly stored in the data structure. Because of the U-shape technique, we also adapt the collision detection and response mechanism in the former method. There are collisions between hair strips and other objects, e.g. scalp and shoulder, and the self-collisions of hair. Finally, a hair modeling function that is capable of modeling different hairstyles is also implemented.

Keywords: Hair Modeling, Hair Animation, Collision Detection, Collision Response, and Real-time Simulation

1. Introduction

In this chapter, we give a brief background about the problems arising in hair modeling and animation. Following this, we present the objective of the work. Finally, we outline the organization of the thesis.

1.1 Background

Hair modeling and animation have been a challenging problem for years. This is mainly because of the unique characteristics of human hair.

On average human being has 100,000 to 150,000 hair strands on his head. Practically in order to obtain a sufficiently good animation, it is necessary to use about 20,000 hair strands for a high quality 3D head model. Around 20 segments or more will be used for each hair strand to make it look smooth. This gives us about 400,000 line segments in the entire hairstyle. Comparatively, a high quality 3D human head uses only about 10,000 polygons and a body with clothing requires another 10,000 to 40,000 polygons. Consequently, hair will potentially consume a large portion of computation resource in human animation, despite being only a small part on a virtual character.

Nevertheless, the large count of individual hair strands gives us even more problem. Having thousands of individual hair strands in a single hairstyle, it is very tedious to specify these hair strands one by one. Therefore an automatic process is necessary to release people from this job.

Moreover, in order to model the dynamics of hair motion correctly, collision detection and collision response during hair movement must be considered as well. Due to the large quantity of individual hair strands, precisely calculating and resolving the collision is almost impossible in practice. Some sort of approximate calculation for collision detection and collision response is therefore necessary.

Another problem is the small scale of an individual hair strand as compared to a single image pixel. To cater for this, special rendering techniques are needed. In order to produce visually realistic animation, it is important for these techniques to take the complex interaction of hair, lighting and shadows into account. These are again computation intensive problems.

Different techniques have been proposed to tackle these problems. Instead of modeling hair individually, there have been works using trigonal prism based approach [Wata92, Chen99]. Plante *et al* [Plan01] proposed a wisp model to approximate interactions inside long hair. Unfortunately, these techniques are not suitable for real-time hair animation. Recently, Koh and Huang [Koh00, Koh01] proposed a strip based approach that animates in real-time. But because of the 2D strip used, it is unable to produce the volumetric effect as real human hair.

All these motivated the work to be described in this thesis, which is to propose an enhanced framework on hair modeling and real-time animation with improved visual effect.

1.2 Objective

Our research is on the methods of hair modeling and animation. In the proposed method, the goal is to achieve a real time rendering speed with a good visual quality.

In the previous work by Koh and Huang [Koh00, Koh01], a method based on 2D strip has achieved real-time hair animation. However, the visual effect was not good enough. In particular, it was unable to produce animation with volumetric effect as natural human hair. The proposed method will solve this problem.

For hair modeling, designing new hairstyle from scratch is a tedious job. People need to specify individual hair stands, which come in thousands, and define the relationship among them. To release user from tedious work of designing hairstyles, the hair modeling function in our method should provide a way to easily specify them. In addition to that, different hairstyles must be designed interactively using the same technique.

Nevertheless, to achieve real-time animation, overall performance of the framework is also important. At the same time, to ensure physical plausibility, some sort of physical model needs to be used to simulate the dynamics of hair model and collision must be handled as well.

1.3 Thesis Outline

Chapter 2 briefly reviews the related existing work in the area of hair modeling and animation.

Chapter 3 introduces the idea of improving visual effect. A method using U-shape hair strips is proposed.

Chapter 4 illustrates the underlying Physics model that is used to simulate the dynamics of hair.

Chapter 5 discusses collision detection and response implemented in our method.

Chapter 6 shows how a hairstyle is modeled in the framework.

Chapter 7 concludes the thesis with brief discussions on future work.

2. Previous Works

There are basically four problems to solve in order to produce realistic animated synthetic actors with hair: hair modeling and creation, hair motion, hair rendering, and collision detection and response [Dald93].

The modeling of hair specifies the geometry, distribution, shape and direction of each of the hundreds of thousands of individual hair strands. Various methods have been used to model human hair. Earlier work models individual hair strands as connected line segments [Anjy92, Dald93]. There is also work using trigonal prism based approach [Wata92, Chen99]. Plante *et al* [Plan01] proposed a wisp model for simulating interactions inside long hair. In a trigonal prism based approach, hair strands are clustered into wisps. Each wisp volume consists of a skeleton and a deformable envelope. The skeleton captures the global motion of a wisp, while the envelope models the local radial deformation. However, there is a lack of coherence in motion among nearby wisps.

Recently, Koh and Huang presented a novel approach that explicitly models hair as a set of 2D strips [Koh01]. Hair strands are grouped into 2D strips. Texture mapping is used to increase the visual effect. The framework is capable of producing a real-time hair animation. However it lacks of volumetric visual effect. Later in this chapter, this framework will be described in detail. There are still a number of other approaches like the ones using volumetric visualization techniques and 3D textures [Neyr98, Kong99] and the lately proposal of modeling dense dynamic hair as continuum by using a fluid model for lateral hair movement [Hada01]. However, they are not real-time.

As all motions are governed by laws of Physics, and almost all hair animation work is based on some sort of physical model [Terz88]. Daldegan *et al* [Dald93] and Rosenblum *et al* [Rose91] used a mass-spring-hinge model to control the position and orientation of hair strand. Anjyo *et al* [Anjy92] modeled hair with a simplified cantilever beam and used one-dimensional projective differential equation of angular momentum to animate hair. Recently, Lee *et al* [Lee00] developed on Anjyo's work to add some details to model hairstyles.

An integrated system for modeling, animating and rendering hair is described in [Dald93]. It uses an interactive module called HairStyler [Thal93] to model the hair segments that represents the hairstyle. Hair motion is simulated using simple differential equations of one-dimensional angular moments as described in [Anjy92]. Collision detection is performed efficiently with a cylindrical representation of the head and body [Kuri93]. Detected collisions between hair strands and the body will respond according to the reaction constraint method [Plat88]. However, due to the complexity of the underlying geometric model of hair, the simulation of the hair dynamics as well as collision detection and response could not be done in real-time even after taking huge liberties with approximating the Physics model for animating hair. In recent work of Chang *et al* [Chan02], a sparse hair model with a few hundred strands is used. Each strand from the sparse model serves as the guide hair for a whole cluster. Once an animation sequence is generated, additional hairs are interpolated to produce a dense model for final rendering.

Now let us take a closer look at the work done by Koh and Huang [Koh00, Koh01] that is most relevant to our work. To reduce the large number of geometric objects to be handled, hair strip (Figure 2.1) is used to represent a group of hair strands. It is in the shape of thin flat patch, which is modeled geometrically by NURBS surface.



A) One hair strip

B) All hair strips overlaying on the scalp

Figure 2.1. Hair modeling in strips *from [Koh00]*

For rendering, the NURBS representation is tessellated into polygon mesh using Oslo algorithm [Bohm80]. Finally, texture maps of hair images are applied on each surface patch. The alpha map defines transparency and creates an illusion of complex geometry to the otherwise “rectangular” surfaces and adds to the final realism (Figure 2.2).

The Physics model used is similar to the one proposed by Anjyo [Anjy92] and later extended by Kurihara [Kuri93]. Hair strand is modeled as connected line segments and polar coordinate system is used to solve the differential equations.

Collisions between hair strips and external objects are detected and handled explicitly.

Collisions in-between hair strips are avoided.



A) Texture map



B) Alpha map



C) Resultant map



D) Collection of hair strips



E) With texture and alpha maps applied

Figure 2.2. The strip-based hair model with texture and alpha maps *from [Koh01]*

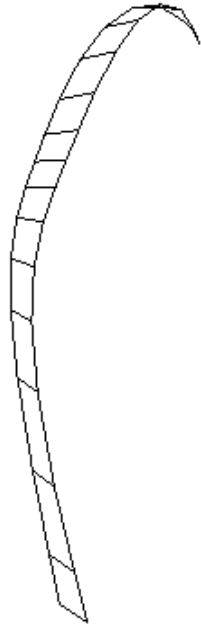
3. U-Shape Hair Strip

In this chapter, we describe the framework in detail. We are going to propose a method to enforce the volume effect for the simple strip based model. In subsection 3.1, details of the U-Shape hair strip are to be discussed. Then, in subsection 3.2, the overall structure of the implementation is to be addressed. And finally, results will be presented in subsection 3.3.

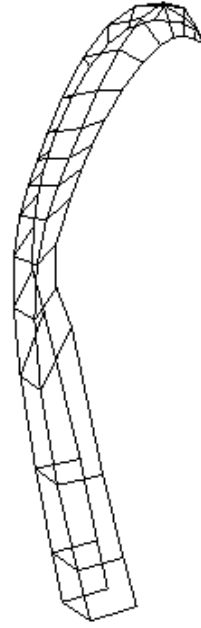
3.1 U-Shape Hair Strip

Using only the basic 2D hair strip, the resulting hair model looks not volumetric. Planting multiple layers of hair strips onto the scalp can of course solve this problem. However with the presence of more hair strips, more computation power is needed to animate these additional layers of hair strips, difficult to be real-time.

In order to enforce the volumetric effect without introducing additional hair strips, U-shape hair strips are used in the framework. The idea is to project the surface meshes of a strip onto the scalp and insert 2 additional surface meshes by connecting the original vertices and the projected vertices (Figure 3.1B). In order to enhance the visual effect, texture and alpha maps are applied to both the original and the projected polygon meshes (Figure 3.1D).



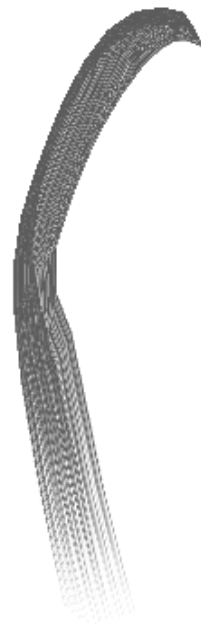
A) Tessellated polygon meshes



B) Polygon meshes with projection



C) Tessellated polygon meshes
with texture



D) Polygon meshes
with projection and texture

Figure 3.1. Illustration of U-shape polygon meshes

The boundary of the scalp is approximated using several spheres. Initially, a point in the scalp is used as the projection center. Vertices of the polygon meshes are connected with the projection center. The intersection of the connected lines and the boundary spheres are taken as the projections of the vertices on the scalp. By connecting the original vertices and the corresponding projections, the projected polygon meshes are retrieved (Figure 3.2).

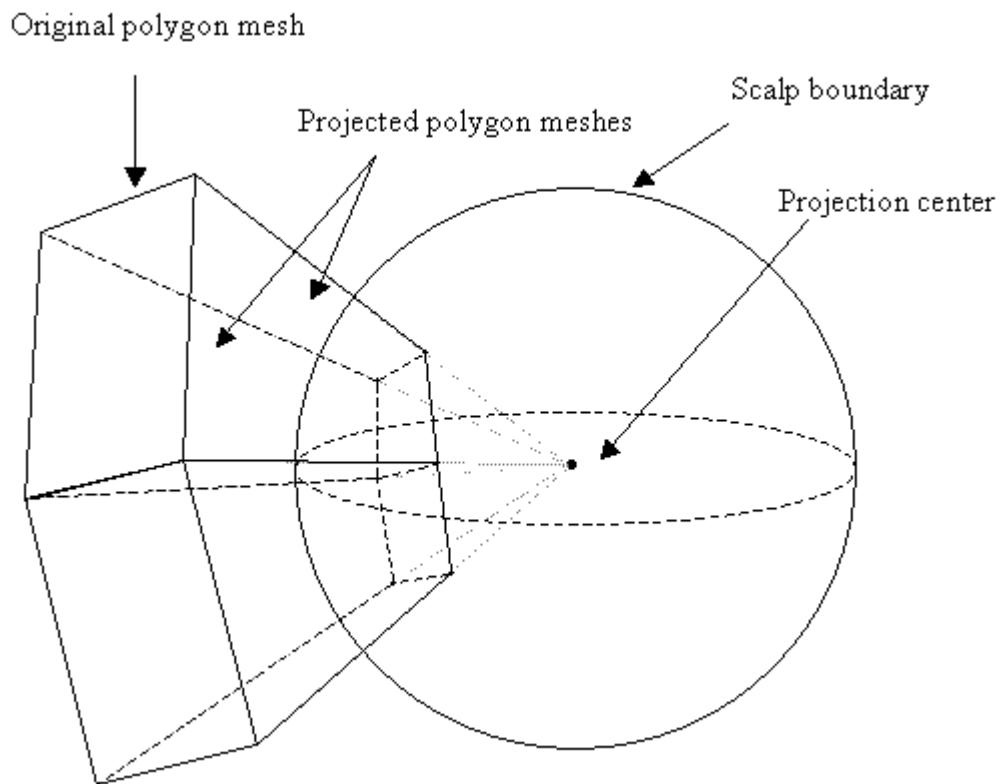


Figure 3.2. Demonstration of projecting polygon mesh onto scalp for deriving U-shape meshes

However, we found that using a point as the projection center could cause problems. It is possible that two neighboring polygon meshes do not fully cover the scalp between them. Figure 3.3A is the cross section view demonstrating the problem. Consequently, part of the scalp is uncovered with hair when observing from certain direction.

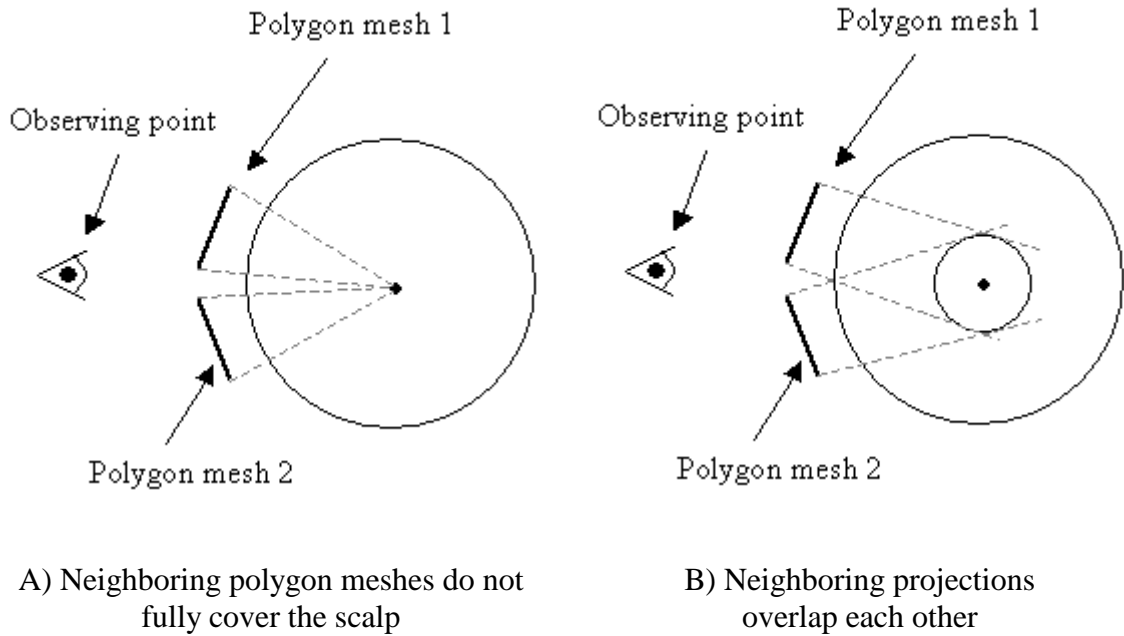


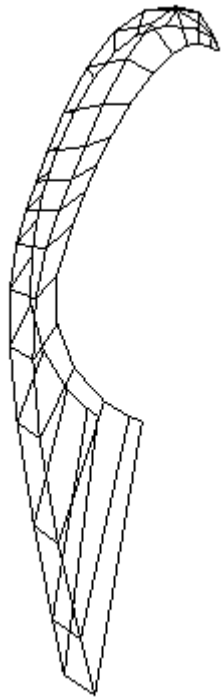
Figure 3.3. Use a sphere as projection center instead of a point to solve the problem that two neighboring polygon meshes do not fully cover the scalp between them

To overcome this problem, we are going to use a small sphere as the projection center instead (Figure 3.3B). By using a sphere as the projection center, there will be some overlapping between the projections of neighboring polygon meshes.

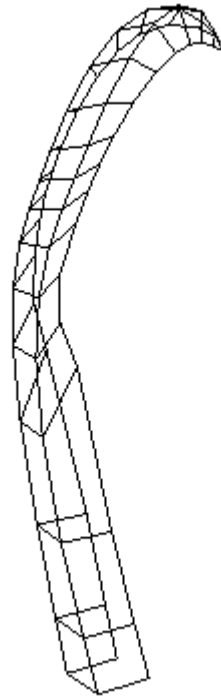
Once the projected polygon meshes are created, it may be necessary to modify them a bit when one of the following undesired characteristics occurs:

- **The projection on the scalp is too far away from the original vertex.**

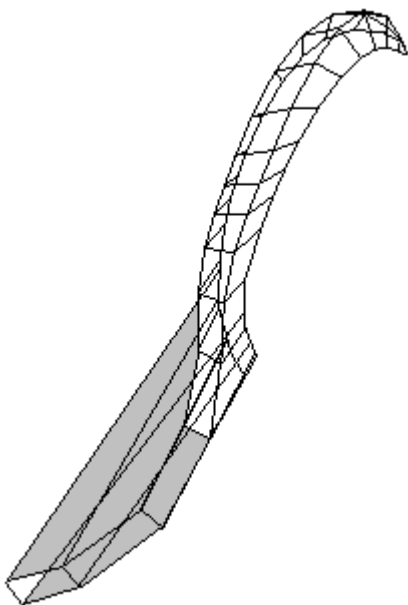
When animating long human hair, the tail of the hair strands would be quite far away from the scalp (Figure 3.4A). If the whole area were used for texture and alpha mapping, the resulting effect would be quite unnatural.



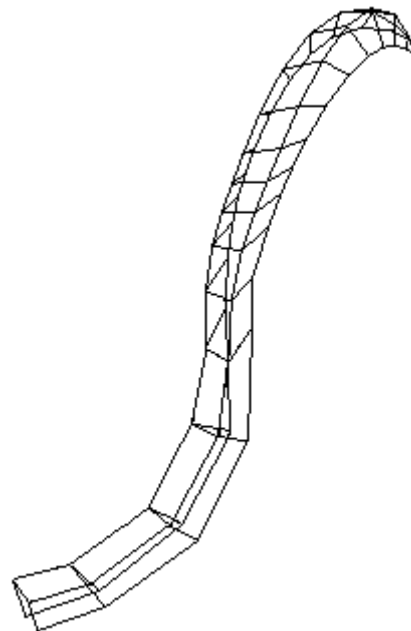
A) Tail of the hair strip is too far away from its projection



B) Modified hair strip projection



C) The shaded part on the left is the *reverse volume*



D) Modified hair strip projection handling *reverse volume*

Figure 3.4. The undesired characteristics and their solutions

- **By connecting the projection and the original vertex, *reverse volume* appears.**

When projecting the hair strips, the projected polygon meshes are expected to be on one side of the hair strips that face the scalp. However, as hair swings in the wind, it is possible that the projected polygon meshes or part of them appear on the other side. They are so called *reverse volume* (Figure 3.4C). It is obvious that this will give us problem during rendering as the hair strip is twisted.

When either of these two undesired characteristics appears, it is necessary to modify the projected polygon meshes. The approach that we use is described below.

Suppose A' is the projection of vertex A and it carries either of the undesired characteristics. A' is then modified so that it carries the following three properties (Figure 3.4B, Figure 3.4D).

1. Vector $\overrightarrow{AA'}$ is perpendicular to the polygon that A belongs to.
2. $|\overrightarrow{AA'}| = C$, where C is a preset constant.
3. The direction of $\overrightarrow{AA'}$ is set to be the same as the side of the hair strip that faces the scalp.

3.2 Implementation

An object-oriented paradigm is adopted for the implementation of the proposed framework by using Java™ SDK standard edition version 1.3.0_01 and Java3D™ 1.3 OpenGL version. Figure 3.5 below is an overview of the scene-graph structure of the proposed framework for hair modeling.

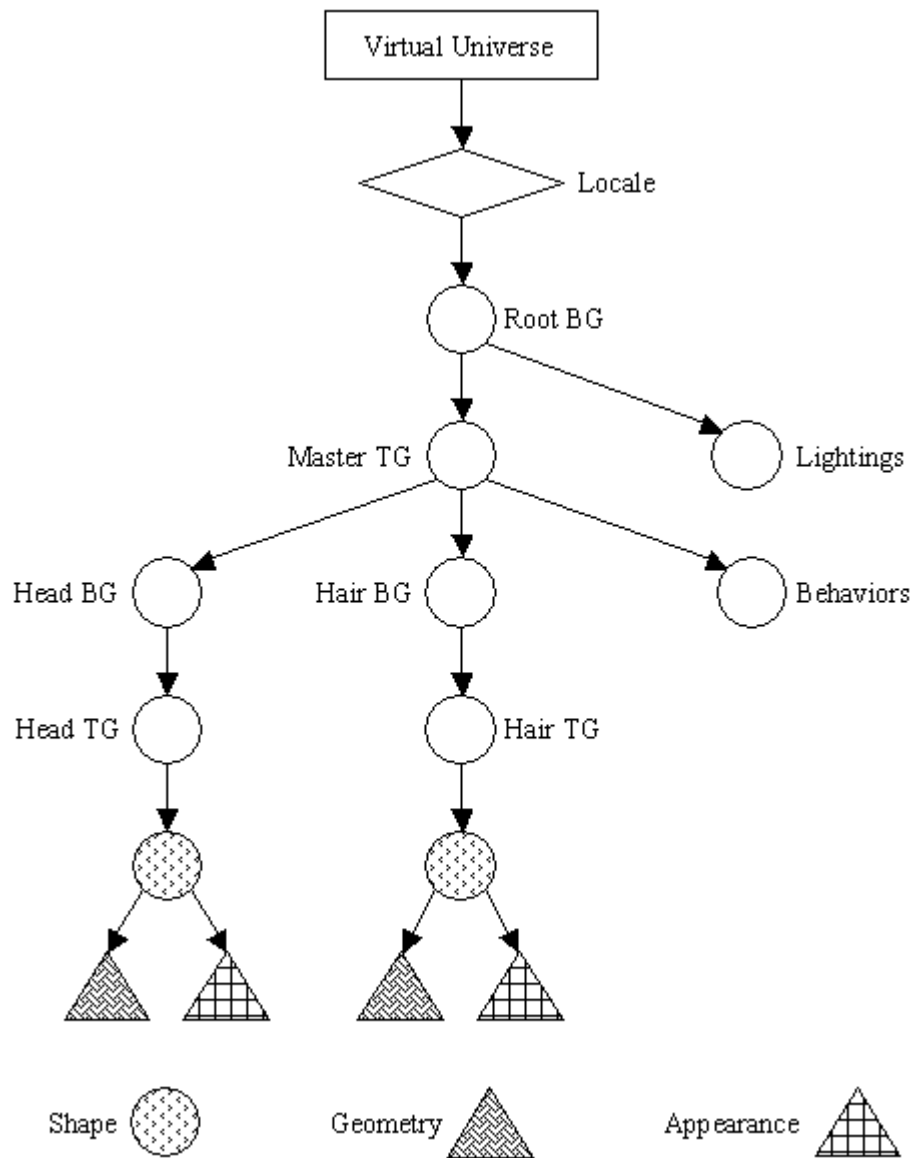


Figure 3.5 Overview scene-graph structure of the proposed framework

On top of the scene-graph is a Virtual Universe, which contains all the objects. A Locale is attached to the Virtual Universe to get a viewpoint of it. A BranchGroup node is a container for other nodes, such as TransformGroups, Behaviors, Lights, and Shapes etc. A Behavior node manipulates the interactions between users and the scene-graph. A TransformGroup node provides matrix transformation to all its children. A Shape node represents an object in the scene. The form of the object is defined by Geometry and Appearance, which are leaf nodes.

The position, orientation and scale of the Shapes are controlled by modifying the transform matrix of their parent TransformGroups.

Keyboard and mouse interruptions from users are parsed via the Behavior nodes, which in turn change the transform matrix of corresponding TransformGroup to rotate, scale or move the objects.

Shadowing and lighting effects are achieved by the Light nodes.

The Head BranchGroup contains a baldhead, onto which hair strips are to be planted.

The Hair BranchGroup holds the list of hair strip objects. Each hair strip object stores the information about its geometry coordinates, which is generated from its NURBS control points, in the Geometry node. To improve visual effect, texture and alpha mappings are applied and stored in the Appearance of the hair strip objects.

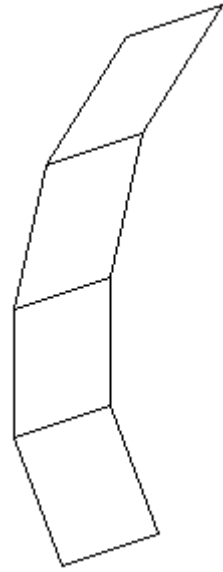
The calculation of the projected polygon meshes

Once the NURBS control points are tessellated into polygon meshes, each of these polygon meshes is to be projected onto the scalp to form two projected polygon meshes. The boundary of the scalp is approximated by a group of spheres. A smaller sphere is placed in the middle of the boundary as the projection center. Let us call this sphere S for simplicity in the following context.

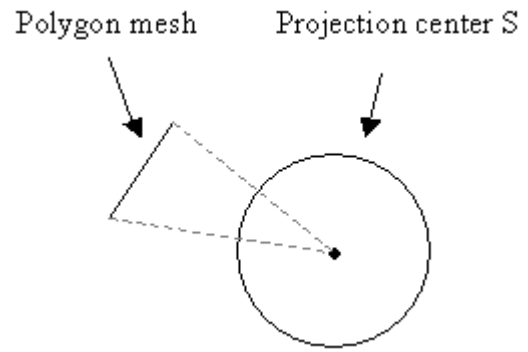
Figure 3.6A shows an example of a tessellated polygon meshes. Horizontally adjacent vertices of the polygon meshes are grouped into pair. For each pair of vertices, they are connected to the center of S (Figure 3.6B). A point is located on S for each vertex in the vertex pair. Let us take Figure 3.6C as an example to describe the property of these newly located points. A and B is the vertex pair; and A' is the newly located point. A is set to be the origin of the local coordinate system. The x-axis points the same direction as vector \overrightarrow{AO} . The y-axis is then set to be $(\overrightarrow{AB} \times \overrightarrow{AO}) \times \overrightarrow{AO}$. The z-axis is set to be pointing the same direction as $\overrightarrow{AB} \times \overrightarrow{AO}$. The point A' carries the following three properties:

- A' is a point in the plane x-y.
- The line that connects A and A' is tangential to S , and line AA' touches S exactly at A'
- The angle between \overrightarrow{AB} and \overrightarrow{AO} is smaller than the one between \overrightarrow{AB} and $\overrightarrow{AA'}$.

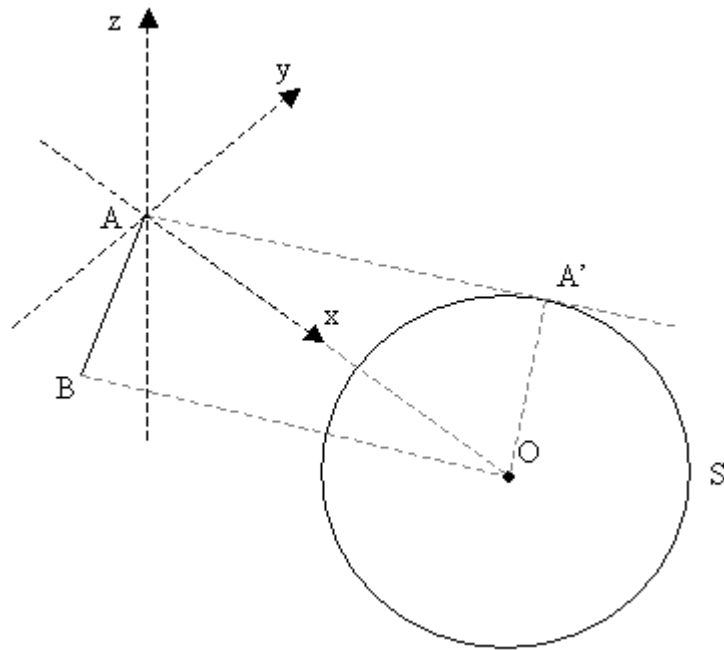
The corresponding point for B can be found in a similar manner.



A) A tessellated polygon mesh



B) Vertices connected to center of S



C) A local coordinate system is used

Figure 3.6 The calculation of the projected polygon meshes

Now we are ready to find the projection of the vertices on the scalp. The algorithm is shown below. It recursively takes the middle point, until the projection is found.

```

A : A vertex on the tessellated polygon mesh
A' : The point on S found by the up-described procedure
Let X = A , Y = A' and Δ be a preset threshold
While |XY| > Δ
    Let Z be the middle point of X and Y
    If Z is inside the boundary of the scalp
        Y = Z
    Else
        X = Z
End
X is output as the projection of A

```

We now proceed to handle the undesired characteristics.

The first undesired characteristic could be easily detected by calculating the distance between each vertex and its projection. The second undesired characteristic, the *reverse volume*, could be detected as follows.

Let us take an example as in Figure 3.7. If the projection of a vertex, say A in Figure 3.7, could cause *reverse volume*, the angle between $\overrightarrow{AA'}$ and $\overrightarrow{AB} \times \overrightarrow{AC}$ would be greater than 90 degree. A similar approach can be used for vertices at other positions.

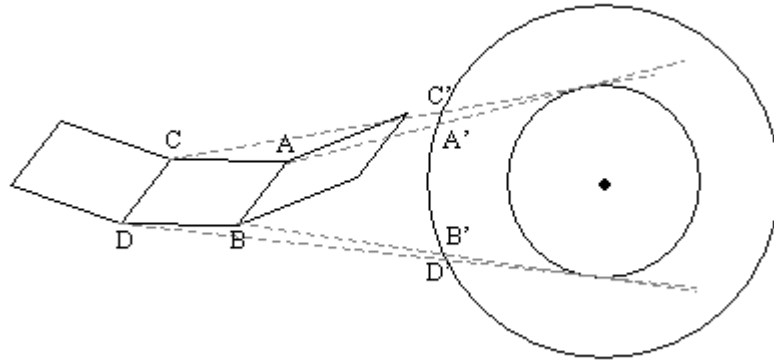


Figure 3.7. The detection of *reverse volume*

If any of the undesired characteristics is detected for a vertex, its projection on the scalp is to be modified as follows.

We will use Figure 3.7 again for easier illustration. Suppose A carries one of the undesired characteristics. The vector $\overrightarrow{AC} \times \overrightarrow{AB}$ is calculated and its length is scaled to a predetermined constant. Name this vector \vec{V} . Let O be the origin of the universal coordinate system. A' , the projection of A , is determined as follows: $\overrightarrow{OA'} = \overrightarrow{OA} + \vec{V}$.

3.3 Results

Figure 3.8 compares the visual improvement when U-shape strips are used.

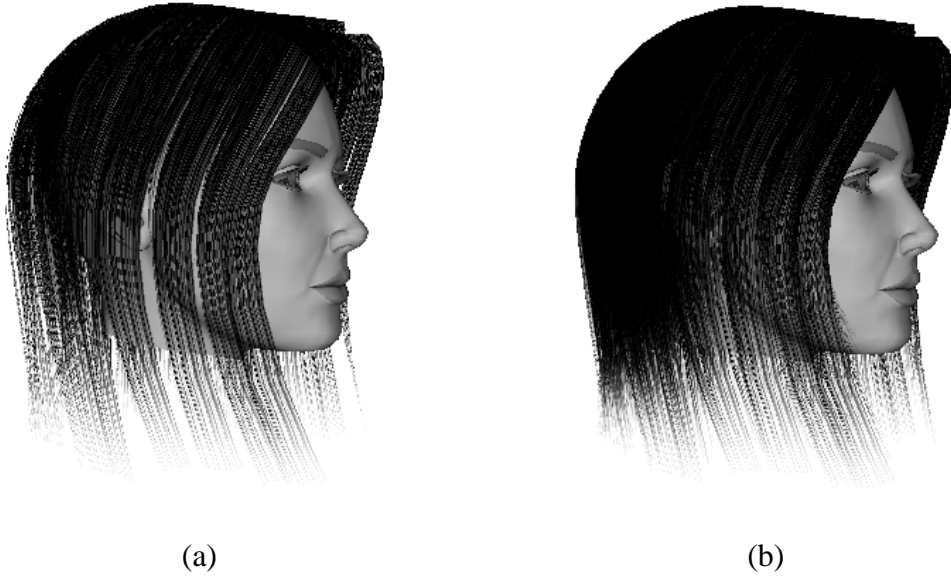


Figure 3.8 Comparing the Visual Improvement of U-shape Strip (b) over Normal Strip(a)

Figure 3.9 below shows a hairstyle made up of 52 hair strips.



Figure 3.9 A hairstyle with 52 hair strips

It is clear that we have improved the volumetric visual effect.

4. Animation

In this chapter, we are going to further extend the framework so that real-time animation of hairstyle with proper collision detection, respond and avoidance is enabled.

In order to produce a realistic hair animation, the movement of the animated hair should be smooth and physically plausible. Additionally, the interaction between hair and external objects (human body) as well as the interaction between neighboring hairs should be handled properly. Therefore, some sort of collision detection and respond mechanism must be enforced. As for inter-hair interaction, techniques like collision detection and avoidance, and collision avoidance are all used.

Our major goal is to devise a physically based model that works with the hair modeling technique presented in the previous chapter. Since we are using a strip-based modeling technique, the objects to be modeled are substantially less than those of an explicit modeling technique. For an acceptable strip-based model, there are about 50 hair strips presented in a hairstyle. While for an explicit model, 20,000 individual hair strands need to be used to produce a nice hairstyle. Therefore, the processing power is reduced substantially. Together with hardware accelerators that are widely available today, real-time physically based hair animation can be achieved.

Our model is based on one-dimensional angular moments proposed by [Anjy92], which is later extended in [Plat88]. The animation of hair is an event-triggered

procedure in which wind, gravity as well as head motion are considered. Additional springs are inserted between strip and scalp, and neighboring strips. Springs between strip and scalp are used to keep certain thickness of the hairstyle. And springs between neighboring strips are used for collision avoidance. Collision between hair strip and external objects, head and shoulder for example, are detected using ellipsoid approximation for bounds and the reaction is resolved using the reaction constraints method [Plat88].

4.1 Overview of the animation framework

Below is the overview of the proposed animation framework:

Wait for an event to trigger (wind, head movement, etc) the animation.

For each time step.

 Tessellate each hair strip.

 For each individual hair strip.

 For each vertex in the hair strip.

 Compute force F_1 due to gravity.

 Compute force F_2 due to head movement.

 Compute force F_3 due to wind.

 Compute force F_4 due to springs between neighboring hair strips.

 Compute force F_5 due to springs between hair strip and scalp.

 Compute $F_{external} = F_1 + F_2 + F_3 + F_4 + F_5$.

 Break $F_{external}$ into two components $F_{\theta external}$ and $F_{\phi external}$.

 Compute $M_{\theta external}$ and $M_{\phi external}$ from $F_{\theta external}$ and $F_{\phi external}$.

 Compute $M_{\theta spring}$ and $M_{\phi spring}$.

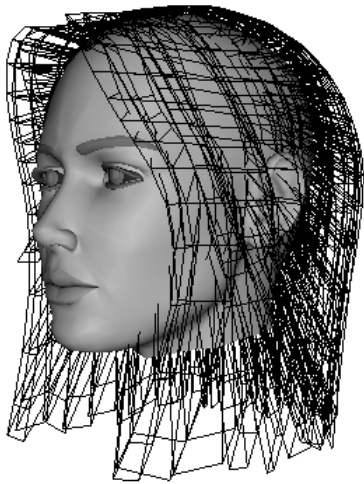
 Compute M_{θ} and M_{ϕ} .

 Compute the new position.

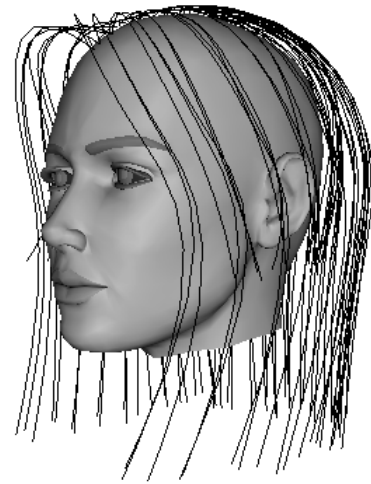
 Collision detection, and respond if necessary.

4.2 The physically based model

In the one-dimensional angular moment model, hair strands are modeled as connected line segments. So as to make this model applicable to our framework, the NURBS representation of the hair strip must be transformed into a suitable form. The method that is adopted is to tessellate the NURBS into polygon meshes as shown in Figure 4.1A. Those horizontal line segments (Figure 4.1B) can be removed. What is left now can be thought of as a sparse set of hair strands that are made up of connected line segments.



A) Full view of tessellated hair strips



B) Vertical lines only

Figure 4.1. The tessellation of hair strips

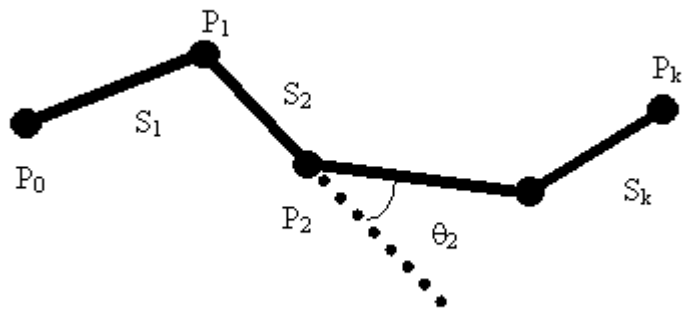


Figure 4.2. Modeling a hair strand as connected line segments

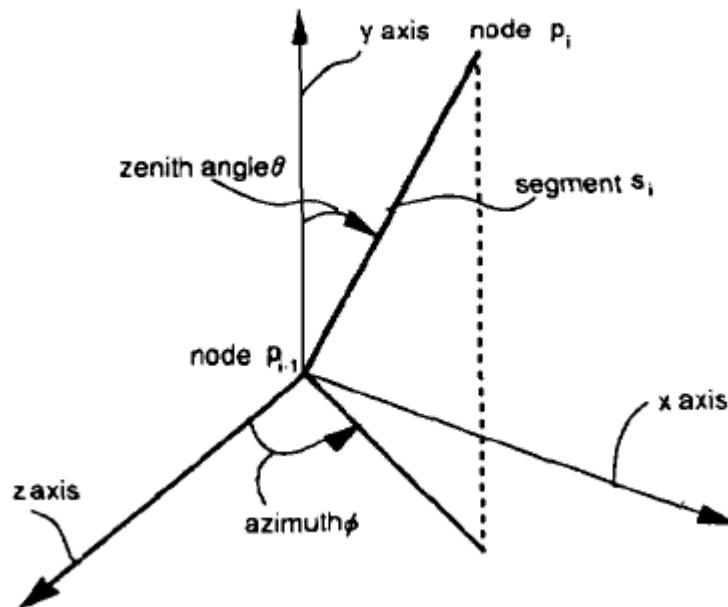


Figure 4.3. Polar coordinate system for a hair strand

Let us now consider a hair strand as shown in Figure 4.2. Assume that the hair strand is not stretchable, which means the length of each line segment does not vary. Then the shape of a hair strand can be represented by the angles between segments. Taking the polar coordinate system as shown in Figure 4.3, the behavior of the zenith angle θ_i and

the azimuth ϕ_i of segment S_i are observed. The variables $\phi_i(t)$ and $\theta_i(t)$ with the time parameter t are governed by the ordinary differential equations:

$$\begin{aligned} I_i \frac{d^2\theta_i}{dt^2} + \gamma_i \frac{d\theta_i}{dt} &= M_\theta \\ I_i \frac{d^2\phi_i}{dt^2} + \gamma_i \frac{d\phi_i}{dt} &= M_\phi \end{aligned} \quad (1)$$

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.

The torque M_θ and M_ϕ applied to the segment S_i are derived from the hinge effect $M_{\theta spring}$ and $M_{\phi spring}$ between two segments, and external moment $M_{\theta external}$ and $M_{\phi external}$ from external forces, such as gravity, wind and external springs.

$$\begin{aligned} M_\theta &= M_{\theta spring} + M_{\theta external} \\ M_\phi &= M_{\phi spring} + M_{\phi external} \end{aligned} \quad (2)$$

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

$$\begin{aligned} M_{\theta spring} &= -k_\theta (\theta - \theta_o) \\ M_{\phi spring} &= -k_\phi (\phi - \phi_o) \end{aligned} \quad (3)$$

where k_θ and k_ϕ are spring constants, and θ_o and ϕ_o are the initial angles.

External moments are defined as

$$\begin{aligned} M_{\theta external} &= uF_\theta \\ M_{\phi external} &= vF_\phi \end{aligned} \quad (4)$$

where u is half of the length of the S_i , v is half-length of the segment that is the projection of S_i onto the ϕ plane. F_θ and F_ϕ are the θ and ϕ components of external force F respectively.

The external force F is defined as

$$F = \rho d(g + a) + df \quad (5)$$

where g is the acceleration due to gravity, a is the acceleration due to the movement of the head itself, and f is the density of the applied force, such as wind and external spring forces.

In the numerical simulation, equation (1) is discretized as

$$\begin{aligned} \theta_i^{n+1} - 2\theta_i^n + \theta_i^{n-1} + \gamma_i \Delta t (\theta_i^n - \theta_i^{n-1}) &= (\Delta t)^2 M_\theta \\ \phi_i^{n+1} - 2\phi_i^n + \phi_i^{n-1} + \gamma_i \Delta t (\phi_i^n - \phi_i^{n-1}) &= (\Delta t)^2 M_\phi \end{aligned} \quad (6)$$

The calculation starts with the segments S_1 , and the new angles of subsequent segments S_i are successively determined using (6).

4.3 Results

Figure 4.4 shows four snapshots of an animation sequence under the wind force. It shows that a satisfactory visual result. More experimental results can be found in the next chapter.

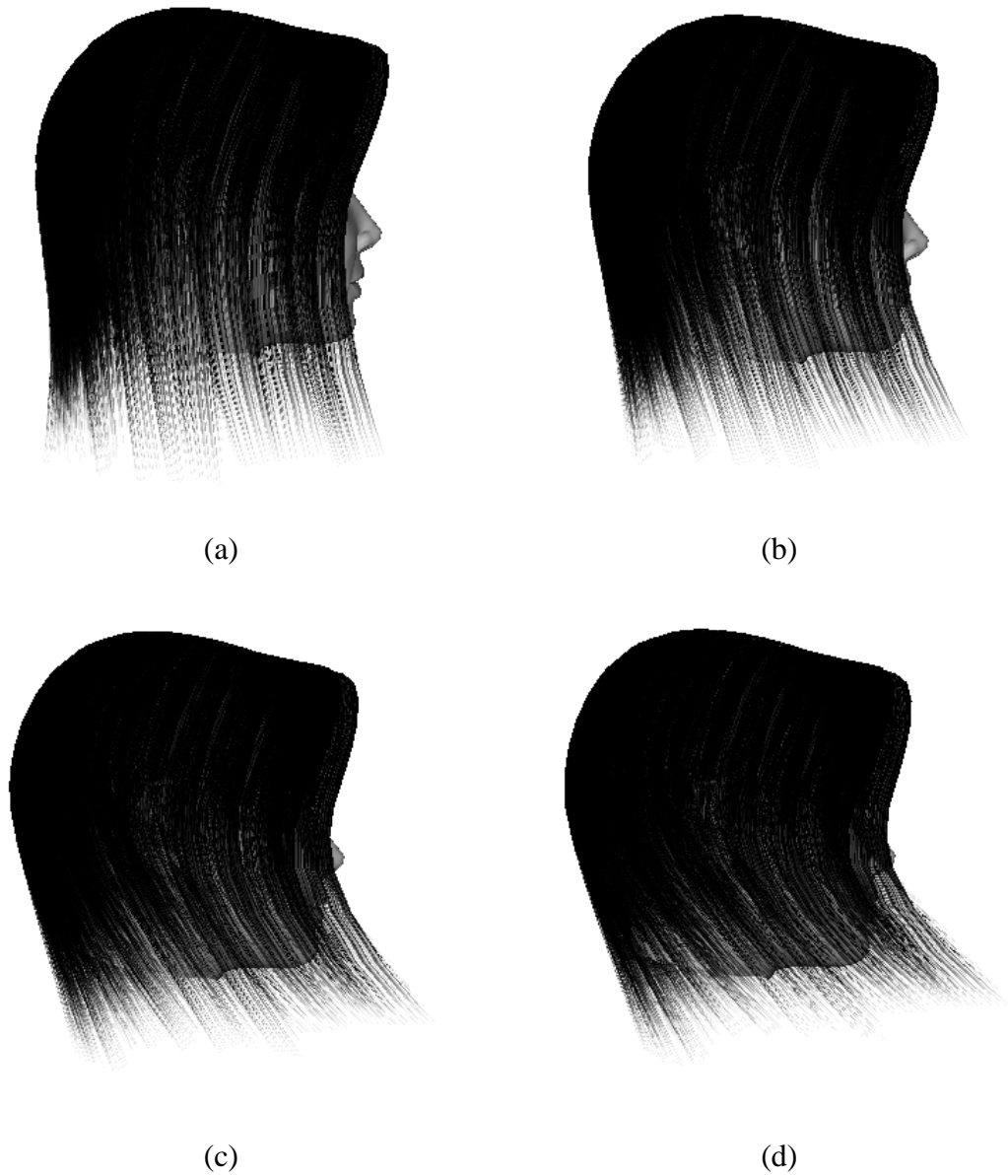


Figure 4.4 Four Snapshots of an Animation Sequence

5. Collision Detection and Response

In order to generate naturalistic hair animation, collision between hair and external objects as well as collision between neighboring hair strips must be considered and handled.

5.1 Handling of collision between hair strips and external objects

The goal is to have a quick and robust method to detect and respond to the penetration of hair strips with external objects. There are basically two kinds of collision. One is the collision of a vertex against external objects. The other is the collision of the line connecting two vertices against external object. Note that it is not necessary that any of the two vertices collide with external object when the line connecting them does collide.

A set of spheres is used to approximate the boundary of external objects, mainly head to accelerate collision detection. Since we are using spheres instead of ellipsoids, detecting the collision of a vertex against external object can be simply achieved by calculating the distances between that vertex and the centers of the spheres and comparing them with the radiuses of the spheres. To detect the collision between the line connecting two vertices and external object, the method on the next page is used.

For each vertex except the top most one

For each sphere that approximate the boundary of external objects

Calculate the distance between the center of the sphere and the line connecting this vertex with the previous vertex.

If the distance is less than the radius, this vertex collides with external objects.

If a vertex intersects external objects, the reaction constraint method [Plat88] is applied to prevent the vertex or the line from penetrating external objects. Let F_{input} be the applied force to the vertex P colliding external objects. Then the unconstrained component of F_{input} is

$$F_{unconstrained} = F_{input} - (F_{input} \cdot N)N \quad (7)$$

where N is the normal vector at point T that is the nearest point on the surface of the sphere to the colliding vertex P .

The constrained force to avoid the collision is

$$F_{constrained} = -(kPT + cV \cdot N)N \quad (8)$$

where V is the velocity of point P , k is the strength of the constraint and c is the damping coefficient.

The output force that is applied to point P is a summation of $F_{unconstrained}$ and $F_{constrained}$.

$$F_{output} = F_{unconstrained} + F_{constrained} \quad (9)$$

Using (7) and (8), (9) can be rewritten as

$$F_{output} = F_{input} - (F_{input} \cdot N)N - (kPT + cV \cdot N)N \quad (10)$$

However, when collision of the second type is detected (collision between a line and external objects), this method cannot be applied directly. What we do is to choose a point on the line that is inside external object. The input force F_{input} of that point is interpolated using the input forces on the two ending vertices. Then the above method is applied and F_{output} is transferred back onto one of the ending vertices using interpolation.

5.2 Inter hair collision avoidance

To prevent the collision between neighboring hair strips, springs are inserted to the hair model. There are three different kind of springs used

1. Spring connecting the vertices of neighboring hair strips.
2. Spring connecting a vertex with the projection of a vertex on the neighboring hair strip.
3. Spring connecting a vertex with the projection of itself.

The first two kinds are used for collision avoidance, while the last one is used for keeping certain thickness of hair strips. Each vertex in our model has a spring of type one connected to it. But spring of type two is not. Only when a vertex is *close* enough to its neighbor or the projection of it, there will be a type-two spring inserted between that vertex and the projection of its neighbor. These springs are inserted because it is not desirable to see a hair strip penetrate into the projection polygon mesh of its neighbor.

For a vertex P that is connected to one or more springs

$$F_P = \sum_s -k_s \times x_s \quad (11)$$

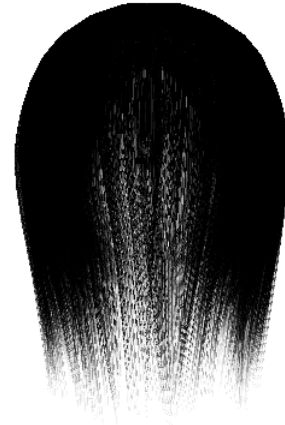
where k_s is the spring constant, and x_s is the extension of the spring from its initial rest length.

5.3 Results

Figure 5.1 shows the effectiveness of the inter hair collision handling mechanism. By enabling inter strip springs, different animation effect can be achieved. Figure 5.1(a) is the initial position of the hair. Figure 5.1(b) and Figure 5.1(c) show the different effect achieved with inter hair collision detection of hair strips enabled (Figure 5.1(b)) and disabled (Figure 5.1(c)) respectively. Figure 5.1(d) is the rest position of the hair. With wired frame, Figure 5.2 shows a clearer view of the inter hair collision happening in Figure 5.1(b) and 5.1(c).



(a)



(b)

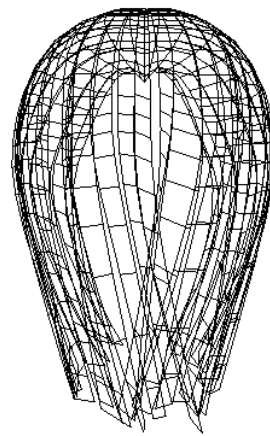
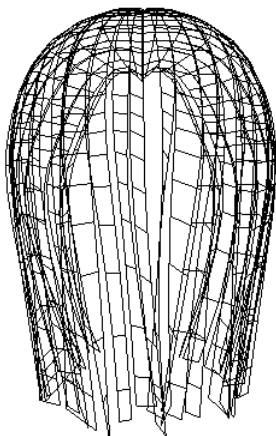


(c)



(d)

Figure 5.1 Inter Hair Collision



(a) Inter hair collision handling enabled (b) Inter hair collision handling disabled

Figure 5.2 Inter Hair Collision in Wired Frame

The whole framework is implemented using Java3D 1.3.1 and JDK 1.4.0 on Windows XP professional edition version 5.1 with service pack 1. The hardware configuration is dual Xeon processors at 2.8G, 2G RAM, and a 3DLabs Wildcat III 6110 graphics card. Experiments have been conducted on this machine to evaluate the performance of our proposed framework. The result is shown in Table 5.1. It showed that real-time has been achieved. Normally about 50 hair strips are enough for a typical hairstyle model.

| Number of Strips | Number of Frames per Second |
|------------------|-----------------------------|
| 20 | 81 |
| 52 | 79 |
| 84 | 67 |
| 116 | 59 |
| 148 | 46 |
| 180 | 41 |

Table 5.1 Frame Rate

6. Hair Modeling

It is a tedious job to design a new hairstyle from scratch, since a hairstyle consists of tens of hair strips and each of these hair strips is made up of tens of vertices. For example, the sample hairstyle used in this thesis has 46 hair strips and each hair strip is made up of 10 vertices, which gives us a total of 460 vertices. It would be a disaster if we have to specify these vertices manually.

To release hair designers from this burden, a hair modeling process is developed in our framework consisting of two steps: interactive and automatic steps.

6.1 Interactive step

Given a scalp model, the designer specifies a few "control hair" of the same number of control points (Figure 6.1). These control hairs are used to define the overall appearance of the hairstyle. Notice that these control hairs must have exactly the same number of control points.

In our implementation, the user starts the design of a new hairstyle by typing the total number of control hairs and the number of control points on each of them (Figure 6.2.a). Then the user can specify the coordinate of the control points in each control hair (Figure 6.2.b). If the user is not satisfied with the position of some of the control points, he can simply select that control point and edit the coordinate of it at any time. By manipulating the control points, the system can easily generate new hairstyle from existing one. Figure 6.4 shows an example of this.

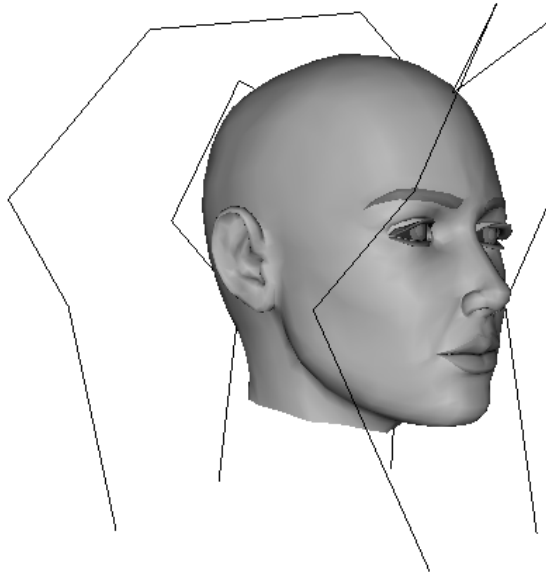
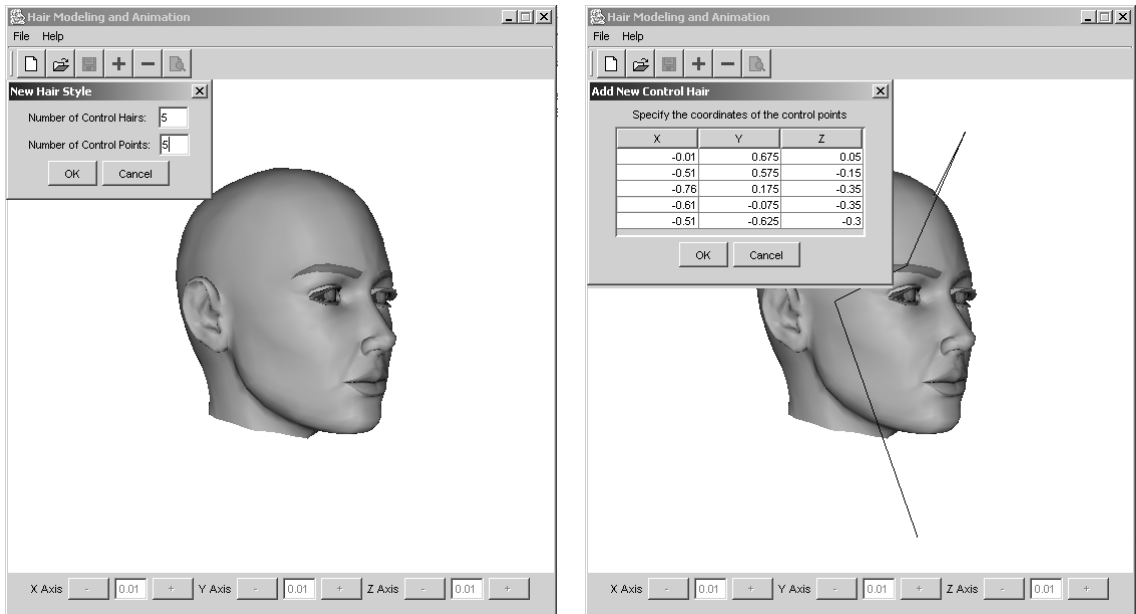


Figure 6.1. Control Hairs



(a) A new hairstyle

(b) Add in control hair

Figure 6.2 Design of new hairstyle

6.2 Automatic step

In automatic step, the control points of different control hairs are first connected horizontally. These horizontally connected control points are treated as NURBS control points and they are tessellated using Oslo algorithm [Bohm80]. Then these tessellated points are connected vertically and used as NURBS control points to be tessellated again vertically. Two neighboring sequences of these tessellated points are connected to form a hair strip. A hairstyle result is shown in Figure 6.3. This can be done at a user specified levels of detail. Finally, in order to improve the visual effect, we apply the texture mapping with the alpha-channel on hair strips of the hair model.

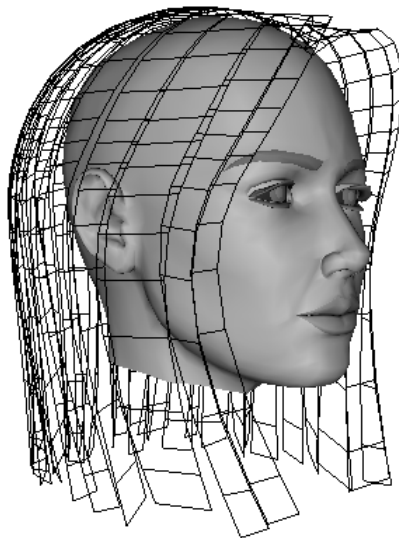
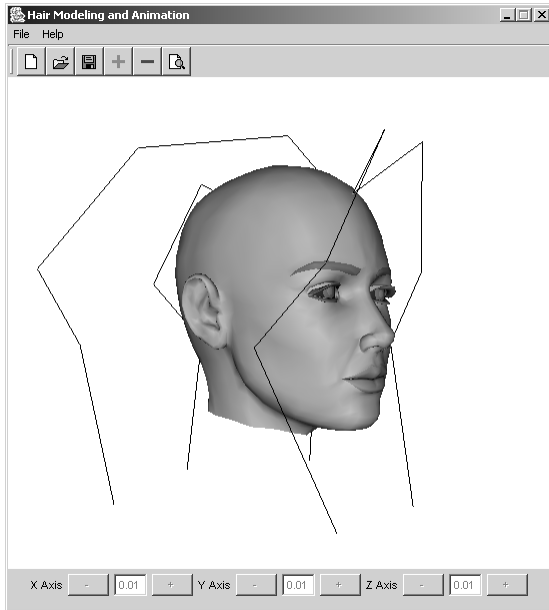


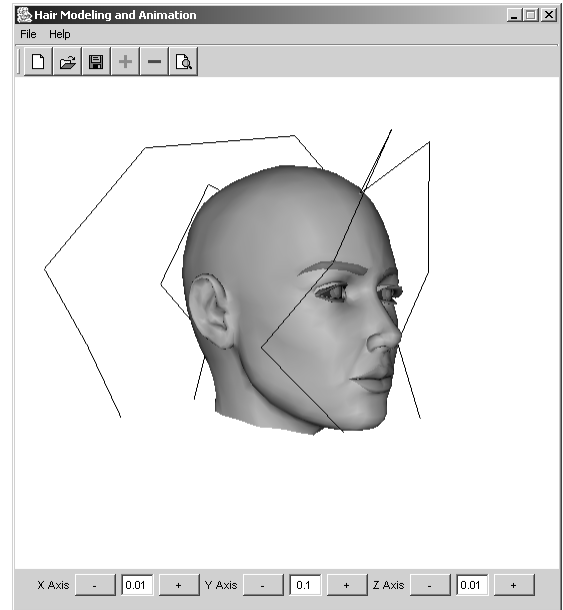
Figure 6.3. Fully tessellated hairstyle

6.3 Results

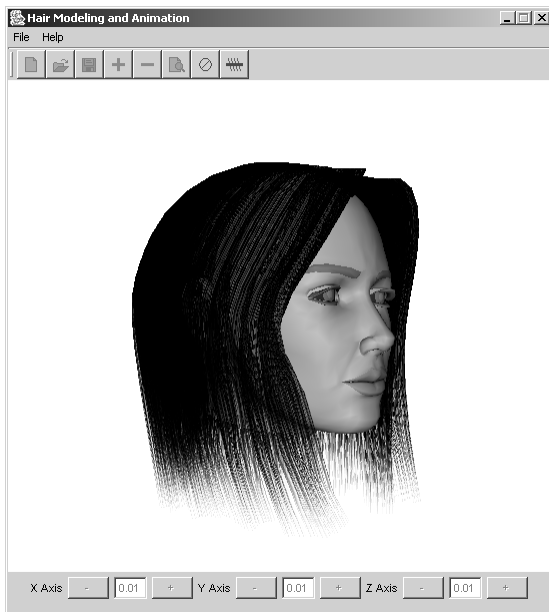
The hair modeling is easy and intuitive such that we can either derive a hairstyle from an existing one or design a brand new hairstyle very fast. Figure 6.4 shows an example of deriving a short hairstyle from a long one.



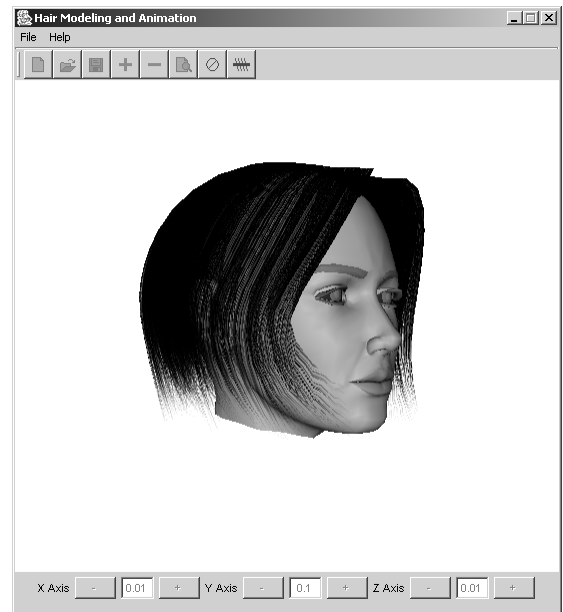
(a) Original Long Hairstyle



(b) Derived Short Hairstyle



(c) Original Long Hairstyle with Texture



(d) Derived Short Hairstyle with Texture

Figure 6.4 Demonstration of Deriving New Hairstyle from Existing One

More examples of different hairstyle are shown in Figure 6.5.

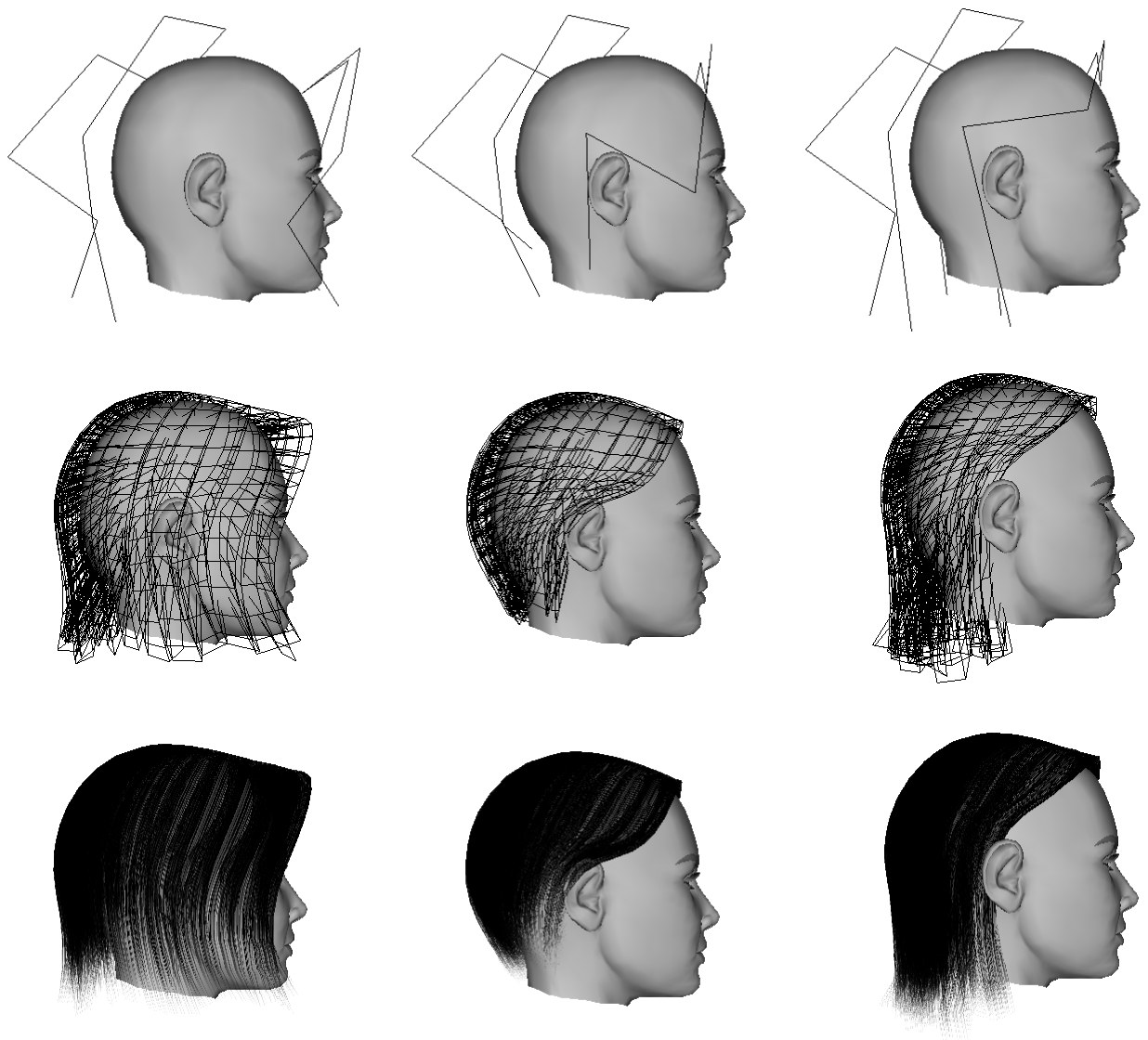


Figure 6.5 More examples of hairstyles

7. Conclusion

Hair modeling and animation is an important area in computer animation. This thesis presented a framework that is suitable for hair modeling and real-time hair animation. In this framework, 2D strips are used to model a hairstyle and each of them represents a group of hair strands. There are typically less than 100 hair strips in a single hairstyle. This greatly reduces the animation cost. To produce a better visual effect, U-shape hair strips are generated and rendered from the visible 2D hair strips. In this way, it enforces the volumetric effect of natural human hair. Texture and alpha mappings, which are supported by hardware accelerator, are applied to both the polygon meshes and those projected polygon meshes.

The physics model used for simulating the dynamics of hair animation is based on one-dimensional angular moments. Springs are inserted between neighboring hair strips and between scalp and hair strips to provide a realistic animation. A model of collision detection, response and avoidance is also specified. Collision of hair strips against scalp and collision between neighboring hair strips are carefully handled.

We have implemented the framework and conducted experiment study. As the testing result demonstrates, the real time hair animation framework is capable of producing visually attractive results in a physically plausible manner.

This framework is applicable to areas that need real-time animation, such as 3D games, virtual human characters in an interactive visual environment, and interactive hair style design.

Reference

[Anjy92] K. Anjyo, Y. Usami, and T. Kurihura. *A Simple Method For Extracting The Natural Beauty Of Hair*, SIGGRAPH (92), pp. 111-120 (1992).

[Bohm80] W. Böhm, *Insert New Knots into B-Spline Curves*, Journal of Computer Aided Design, 12(4), pp. 199-201, 1980

[Chan02] Johnny T. Chang, J. Jin, and Y. Yu, *A Practical Model for Hair Mutual Interactions*, ACM SIGGRAPH Symposium on Computer Animation, pp. 73-80, (July 2002).

[Chen99] L. H. Chen, S. Saeyor, H. Dohi, and M. Ishizuka. *A System of 3D Hair Style Synthesis Based on the Wisp Model*, The Visual Computer, 15 (4), pp. 159-170 (1999).

[Dald93] A. Daldegan, T. Kurihara, N. Magnenat Thalmann, and D. Thalmann. *An Integrated System for Modeling, Animating and Rendering Hair*, Proc. Eurographics (93), Computer Graphics Forum, Vol.12, No3, pp.211-221 (1993).

[Hada00] S. Hadap and N. Magnenat-Thalmann. *Interactive Hair Styler Based on Fluid Flow*. Computer Animation and Simulation 2000. Proc. of the Eleventh Eurographics Workshop (2000).

[Hada01] S. Hadap and N. Magnenat-Thalmann. *Modeling Dynamic Hair as Continuum*, Eurographics Proc. Computer Graphics Forum, Vol.20, No.3 (2001).

[Kim01] T.-Y. Kim and U. Newmann, *Opacity Shadow Maps*, Proc. Of Eurographics Workshop on Rendering, pp. 177-182 (2001).

[Koh00] C. K. Koh and Z.Y. Huang, *Real-time Animation of Human Hair Modeled in Strips*, In: Computer Animation and Simulation, Springer-Verlag, pp.101-110 (2000).

[Koh01] C. K. Koh and Z.Y. Huang, *A Simple Physics Model to Animate Human Hair Modeled in 2D Strips in Real Time*, Proc. of Computer Animation and Simulation (2001).

[Kong99] W. Kong and M. Nakajima. *Visible Volume Buffer for Efficient Hair Expression and Shadow Generation*, Computer Animation (99), IEEE Computer Society, pp. 58-65 (May 1999).

[Kuri93] T. Kurihara, K. Anjyo, and D. Thalmann. *Hair Animation with Collision Detection, Models and Techniques in Computer Animation (93)*, Springer-Verlag, Tokyo, pp. 128-138 (1993).

[Lee00] D. W. Lee and H. S. Ko, *Natural hairstyle modeling and animation*. In Proc. of International Workshop on Human Modeling and Animation, pp. 11–21, Seoul, Korea, (June 2000). Korea Computer Graphics Society.

[Neyr98] Fabrice Neyret. *Modeling, Animating, and Rendering Complex Scenes Using Volumetric Textures*, IEEE Transactions on Visualization and Computer Graphics, 4(1), pp. 55-70 (January-March 1998).

[Plan01] E. Plante, M.P. Cani, and P. Poulin. *A Layered Wisps Model for Simulating Interaction Inside Long Hair*. In Proc. of Eurographics Computer Animation and Simulation 2001.

[Plat88] J. C. Platt and A. H. Barr. *Constraint Methods for Flexible Models*, SIGGRAPH (88), pp. 279-288 (1988).

[Rose91] R. Rosenblum, W. Carlson, and E. Tripp, *Simulating the structure and dynamics of human hair: Modeling, rendering and animation*. Journal of Visualization and Computer Animation, 2:141–148, (June 1991).

[Terz88] D. Terzopoulos and K. Fleischer, *Deformable Models*, The Visual Computer, Vol 4. No. 6 pp. 306-331 (1998).

[Thal93] N. Magnenat Thalmann and A. Daldegan. *Creating Virtual Fur and Hair Styles for Synthetic Actors. In Communicating with Virtual Worlds*, Springer-Verlag, Tokyo, pp. 358-370 (1993).

[Wata89] Y. Watanabe and Y. Suenaga. *Drawing Human Hair Using Wisp Model*, Proc. Computer Graphics International, pp.691-700 (1989).

[Wata92] Y. Watanabe and Y. Suenaga. *A Trigonal Prism-Based Method For Hair Image Generation*, IEEE Computer Graphics and Applications (12), No. 1, pp. 47-53 (1992).

[Yu01] Y. Yu, *Modeling Realistic Virtual Hairstyles*, Proc. of Pacific Graphics, pp. 295-304 (2001).