國立交通大學

多媒體工程研究所

碩士論文

即時頭髮模擬使用之有效率單線動力學

Efficient Strand Dynamics for Real-Time Hair Simulation

研 究 生：王上文

指導教授：林文杰　教授

中華民國 九十七 年十月

Efficient Strand Dynamics for Real-Time Hair Simulation

Student: Shang-Wen Wang　　Advisor: Dr. Wen-Chieh Lin

Institute of Multimedia Engineering
College of Computer Science
National Chiao Tung University

ABSTRACT

Simulating hair motion is difficult due to their high-stiffness property, which usually causes numerical stability problem and needs the numerical simulation be performed at a small time step. We observed that a force applied to one part of a strand will transport to other parts of the strand where the force can contribute the deformation of the strand. We model this force transport process and simplify the special Cosserat Theory to simulate inextensible hair strands. Our experiments show that the proposed approach can simulate hair dynamics robustly and efficiently such that hair simulation with user interaction can be executed in real time.

Acknowledgments

I would like to thank my advisor, Dr. Wen-Chieh Lin for his guidance. Thanks to my colleagues in the CG Group.: Fang-Shuo Li, Kuei-Li Fang, Chin-Hsian Chang, Hsin-Hsiao Lin, Kuang-Wei Fu, Ying-Tsung Li, Yueh-Tse Chen, Zhi-Han Yan, Ming-Shang Hsu, Jia-Ru Lin, Ying-Shou Lan, Yi-Jheng Huang, Wei-Chien Cheng,and Hsi-Chou Huang for their assistance and discussion.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Realistic hair has been studied for years, including modeling, simulating, and rendering. Simulating hair strands is hard because of their high stiffness property. Simulating such hair usually surfers from numerical stability problem. It needs a very small time step to simulate high stiffness of hair.

The focus of this thesis is the transport of force in a hair strand with one fixed end and one free end. Finding where the force contributes is helpful to the stability of strand simulation, especially in high stiffness model, because force will not take so many time steps to transport. In experiment, our method indeed gives fast stability and good performance in hair simulation and make it be real-time.

For high-stiffness strands, we observed that forces applied to one segment will transport to root segment by segment to find where to contribute deformation. In a zero stiffness model, force is transported along the tangent of the strand, which is called the axial force. In a high stiffness model, force is not only transported by the axial force but also the stiffness torque which generate by material properties. The speed of transport through energy method is bounded by simulation time step and spatial step,though the speed is fast in high stiffness material. Too slow force transport also reduces stability of simulation. This is the bottleneck of traditional energy methods, like the spring-damping model. Finding the Jacobian matrix to map a local force to global may help, but update the matrix is a huge overhead.

We observed that hair strands is a constrained object rather than a free-deform object.The difference of hair shape at each time step is small. That means the motion of hair strand is more similar to a rigid body than a free rigid body chain. Therefore we assume that there will be no large local change in hair shape between two time step, and hair strands tend to have a global motion. Shape and stiffness torques will be similar between two time step. This make it able to compute the distribution of forces from its current shape and stiffness torques.

We then use the the Special Cosserat Theory[1] to model the curliness of strands; However solving the Special Cosserat Theory is hard and computationally expensive[2], which is not suitable for real time simulation. We did some simplification to make it suitable for our force distribution.

CHAPTER 2

Related Work

There have been much work on hair simulation, including hairstyle modeling, hair dynamics, and hair rendering. It is beyond the scope of this thesis to survey all literature here. As our goal is to simulate hair motion, we will only review the related work on the physics-based model of single hair strand motion in this chapter. For a thorough survey on hair simulation, please see [3].

2.1 Spring-Damper Model

The spring-damper model is a popular model for simulating elastic objects for its simplicity. In this model, a hair strand is often modeled as a chain of connected particles[4][5]. This method is fast and easy to implement, but is not suitable for simulating high-stiffness objects, like hair. Since the stiff dynamics usually require using small time step, hair motion simulation is usually a bottleneck of the whole system. Additionally,simulating torsion or curliness of hair is also a hard problem. Although Selle et al.[6] simulates torsion and curliness successfully, their solution for spring stretch still makes it lack of physical accuracy.

2.2 Rigid-body Chain Model

In order to simulate high stiffness of hair, the Rigid-body Chain model is more appreciated. The Rigid-body Chain removes stretching DOFs of strand, makes each strand inextensible[7][8]. The Rigid-body Chain model needs more computation than the Spring-Damper Model, but still an $O(n)$ algorithm[9]. Although Rigid-body Chain solve the high stiffness problem of the Spring-Damper model, it still has problem in simulating curliness.

2.3 Cosserat Model

Cosserat Rod is first introduced to the computer graphics field by Pai [10] in 2002. The kinematic state of a Cosserat rod is generated by integrating the spatial speed and spatial angular speed over the rod, which can preserve local coordinate at every point of the whole strand. Therefore, Cosserat Rod model can model torsion and torsion stiffness. Pai used the Cosserat model to get only the statics of a whole strand, not the dynamics.

Bertails et al. (Super Helices)[11] simulated curly hair dynamics using the Cosserat Rod. They assumed a curly hair strand can be discretized into several segments. This method preserves torsion of curly hair and improves computational efficiency with reduced DOFs. When the hair length is long, external force is applied or hair strands interact with other objects, curly hair needs to be simulated using more DOFs. The computational cost grows rapidly as $O(n^2)$ where n is the DOFs of the hair strand.

Spillmann et al. also proposed a 1-D elastic model [12] based on Cosserat Rod. They suggested the strand is composed of many finite elements. The dynamics of the model can be solved in $O(n)$. Unlike the Super-Helices, the method proposed by Spillmann et al. simulated each control point on the strand. A drawback of their method is that the stability of the method will be affected by increasing number of segments when we model high stiffness of hair.

CHAPTER 3

Strand Simulation

In this chapter, we describe our strand simulation method. First, we briefly review the Cosserat Model and Special Cosserat Theory. Second, we describe our reduced coordinate that makes the conversion between torque and force possible and reduces the computational time. Third, we describe our forces transport method. Then we rewrite the Special Cosserat Theory to obtain the dynamics equation. Finally we solve this equation numerically to simulate hair motion.

3.1 The Cosserat Model and Special Cosserat Theory

Cosserat Rod was first introduced to the computer graphics field by Pai [10] in 2002. The dynamics of the rod is governed by an integration of positional variation and orientational variation[1] over the rod:

$$d_3{}' = u \times d_i \text{ i = 1 to 3,} \tag{3.1}$$

where d_i is the local axes on the rod as shown in Fig 3.1, and u is the Darboux vector. In order to model the unstretchedness, the tangent along the center line of

[1] Notation: We use f' to denote spatial derivative along the center line of a strand, $f' = \partial f / \partial s$, and $\dot{f} = \partial f / \partial t$ to denote time derivative.

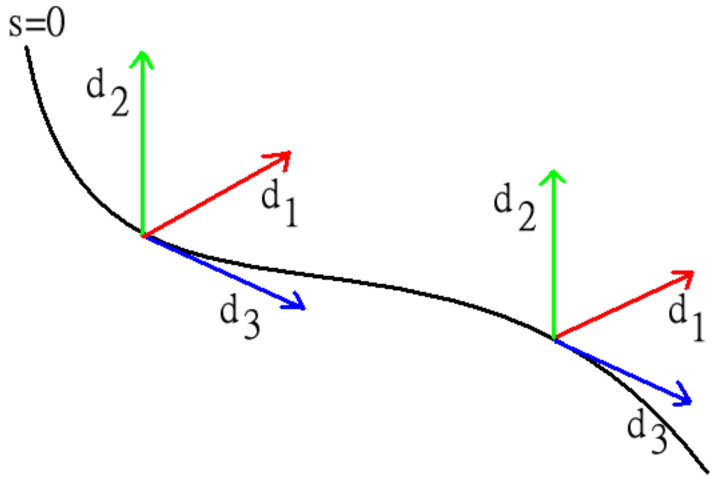


Figure 3.1: Rod Representation

the rod is:

$$r' = d_3 \tag{3.2}$$

where r is the center line position of the strand. If we integrate r' along the center line, we can get the position of the Cosserat Rod (Strand):

$$r(l) = \int_0^l d_3(s)ds + r(0) \tag{3.3}$$

where l is the distance from the root along the center line and $r(0)$ is the position of the root of the strand.
Because the Cosserat Strand can model torsion and curliness of strands, we use this model as our hair strand model.

The Special Cosserat Theory uses two equations to describe the dynamics of the strand. One is for position:

$$n' + df = \rho A\ddot{r} \tag{3.4}$$

where n' is the constraint force generated by the material law, df is an external force applied to an infinitesimal segment, ρ is the linear density, and A is the area of the cross section. This is basically Newton's Second Law. Because n is affected by the shape of strand and torque, n usually does not have a simple formula to compute[2]. The other equation of the Special Cosserat Theory describing the orientation of the rod is:

$$m' + r' \times n + d\tau = \rho I \dot{\omega} \tag{3.5}$$

where m' is the torques of stiffness generated by the material law, $d\tau$ is an external torque applied on infinitesimal segment, I is the moment of inertia of an infinitesimal segment,and ω is the angular velocity.

In the Special Cosserat Theory, there are two unknowns, n and $\dot{\omega}$)in the equation. The resulting acceleration is affected by the force and torque simultaneously.

From Equation 3.4, we know that a force that do not contribute to the linear acceleration of the strand will become an internal force. Furthermore, we know that the component of a force that does not induce any effective torque at a point of the rod will also become an internal force. Based on these two findings, we compute the effects of any external force applied to the rod by propagating the force segment by segment. We consider this problem as a force transport problem by finding where the force contributes to effective torque. However, different from the Special Cosserat Theory, this force will be the summation of many segments's force. We need to consider the affected mass point when we compute dynamics. After computing transport forces, we use an approximated equation of Equation 3.5 to compute strand dynamics.

3.2 Representing the Rod State in Reduced Coordinate

In order to represent the 3D position and 4D quaternion orientation of an infinitesimal segment on a rod and their derivatives, a 14-dimension vector is needed to represent the state of the strand. However, based on the Cosserat Model's definition, we can compute the position and velocity of each point on the centerline using Equation 3.3.

Also, from the rigid body dynamics we know:

$$\dot{d_3} = \omega \times d_3$$

where ω is the angular velocity.

Therefore the speed on center line, which is the temporal differentiation of equation 3.3, is

$$\dot{r}(l) = \int_0^l \dot{d}_3 ds = \int_0^l (\omega \times d_3) ds \tag{3.6}$$

The temporal derivative of $\dot{d}_3$ is

$$\ddot{d}_3 = (\dot{\omega} \times d_3 + \omega \times \dot{d}_3) = (\dot{\omega} \times d_3 + \omega \times (\omega \times d_3))$$

so the linear acceleration on the center line is

$$\ddot{r}(l) = \int_0^l \ddot{d}_3 ds = \int_0^l (\dot{\omega} \times d_3 + \omega \times (\omega \times d_3)) ds \tag{3.7}$$

which means the position of a segment on the rod can be obtained from the orientation of the segment if we add the centrifugal force to the external force. The centrifugal force will be described in the appendix. Therefore, we only need an 8-dimensional vector to represent the state of the rod.

3.3 Force Transport

From the classical mechanics, we know that a force f applies to a point with distance l_s to the rotation center can be converted into a torque τ:

$$l_s d_3 \times F = \tau = I\dot{\omega} \tag{3.8}$$

This means a non-twisting force can be transformed into a bending torque. Therefore, we can find where the force results a torque by checking shapes and torque on one strand. This equation makes our force transport possible.

Beginning from the tip, we find where force becomes a torque segment by segment using Equation 3.8. In a strand segment, force can be decomposed into two components,tangential force and normal force, as shown in Fig 3.2. We observed that there are two situations that force will be transported to proximal segment:

1. Tangential Force
   A force component, which is a tangential force, results no effective torque. It will be transported to next segment.

2. Normal Force which results no ineffective Force
   A force component, which neutralized by a applied torque, will result no effective torque. It will be transported to next segment.

We will describe these two cases below.

3.3.1 Tangential Force

It is obvious that a tangential force does not induce any torque on the segment, and is just transported to other segments as shown in Fig 3.3. If a tangential force is transported to a segment that can induce a torque to the segment, we remove it from the transported force and set is as a torque to bend the segment.

The net force F_{in} at position $r(l)$ is the sum of the external force applied to the position $r(l)$ and the tangential force transported from the position $r(l + ds)$.

$$F_{in}(l) = (f_{ex}(l) + F_T(l + ds)) \tag{3.9}$$

where f_{ex} is the external force, and $F_T(l) = (F_{in}(l) \cdot d_3(l))d_3(l)$ is the tangential force at $r(l)$. $F_{in}(l)$ is the summation of external force and transported axial force. Thus, we can obtain the normal force $F_N(l)$ at $r(l)$.

$$F_N(l) = F_{in}(l) - F_T(l) \tag{3.10}$$

The normal force will result a torque at $r(l)$.

3.3.2 Normal Force

Recall our assumption in Section 3.1 that the component of force which can not result a effective torque will become a internal force. When torques involves, we need to compute how much the effective torque resulted by F_N. We call the component of force which results a torque as $f_{effective}$,

$$f_{effective}(l) = F_N - F_{neutralized}(l)$$

$F_{neutralized}$ is the component of force neutralized by stiffness torque, which can be written as:

$$F_{neutralized}(l) = c\overline{\nu} \tag{3.11}$$

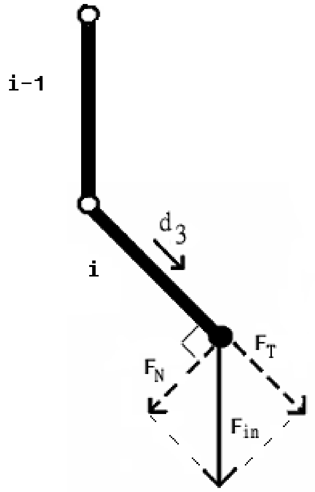


Figure 3.2: Tangential Force and Normal Force: The component of F_{in} parallel to d_3 is the tangential force, and the other is the normal force.

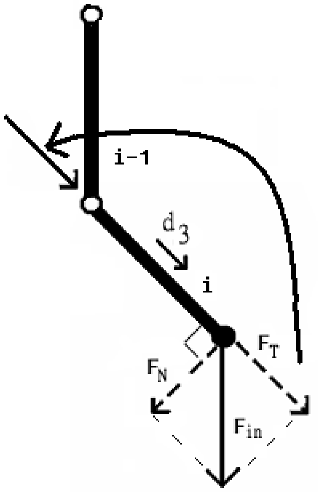


Figure 3.3: Transport of Tangential Force, the component parallel to d_3 of force in segment i will be transported to proximal segment $i-1$.

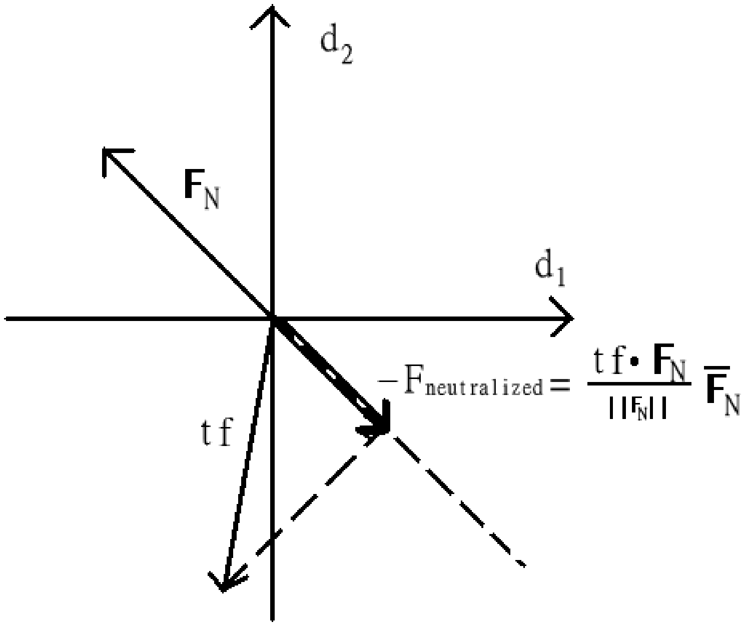


Figure 3.4: Normal Force and Torque: d_1 and d_2 are the axes of the local coordinate on centerline. F_N is the normal force, and tf is the local force transformed from torque. $\overline{F_N}$ is a normalized vector of F_N. The neutralized part is $max(\|F_N\|, -\frac{tf\cdot F_N}{\|F_N\|})\overline{F_N}$

where c is a constant and $\overline{\nu}$ is a normalized vector of $F_N(l)$.

In order to compute c, we transform the total torque τ applied here to a force to find F_N's contribution.

$$tf = T \times d_3/l_s \tag{3.12}$$

where l_s is the length of segment. Then the length of F_N neutralized by tf is:

$$c = \begin{cases} 0, & \text{if } (projLength(F_N(l), tf(l)) > 0 \\ min(-projLength(F_N(l), tf(l)), \|F_N(l)\|) & \text{otherwise} \end{cases} \tag{3.13}$$

$projLength$ is:

$$projLength(F_N(l), tf(l)) = \frac{tf(l) \cdot F_N(l)}{\|F_N(l)\|} \tag{3.14}$$

The force neutralized by torque is also shown in Fig 3.4.

We need to rewrite Equation 3.9 for $F_{neutralized}$:

$$F_{in}(l) = f(l) + F_T(l + ds) + F_{neutralized}(l + ds) \tag{3.15}$$

After integrating from tip to root, we get a series of F_N, which is the force make the strand deform.

3.4 Discretization of Strand Dynamics Equation of the strand

We describe how we discretize the Special Cosserat Theory in this section. As we only concern the orientation of the rod, we do not need to discretize Equation 3.4. Recall that the orientation equation of the Special Cosserat Theory is:

$$m' + r' \times n + d\tau = \rho I \dot{\omega}$$

If we discrete it, we get

$$\Delta m_i/\Delta s + (d_{3,i} \times n_i) + \tau_i/\Delta s = \rho I \dot{\omega}_i$$

$$\Delta m_i + (d_{3,i} \times n_i) \cdot \Delta s + \tau_i = \rho I \dot{\omega}_i \cdot \Delta s \tag{3.16}$$

Equation 3.16 is only for single segment case. In practice, the torque applied to a segment will result accelerations of other segments and decrease the acceleration of itself. The resulting acceleration is $d_3 \times n$. Since n does not have a simple formula , we approximate n by considering segments that are accelerated by τ_i.

First we need to know how many segments will be accelerated by the torque. We observe that in a high-stiffness strand, the local deformation caused by other segments' acceleration will be canceled by stiffness very fast. This is similar to a rotating rigid body.

So we consider the number of segments being accelerated by the force. We assume that the accelerated segments are located at the distal side of the segment. The total mass will be $x \cdot mass$ which x is the number of the accelerated segments.

If we consider the force contributed to acceleration of $x \cdot mass$ as $(x \cdot mass)\ddot{r}$. Then the dynamics equation will become:

$$(d_3 \cdot l_s) \times (F - (x \cdot mass)\ddot{r}) = \rho I \dot{\omega}$$

where $F = \tau \times d_3$ is the bending force transformed from torque, and l_s is the length of segment. Then we write $\ddot{r}$ as angular acceleration:

$$d_3 \times F = \rho I \dot{\omega} + (x \cdot mass)(d_3 \times \omega \times d_3)(l_s)^2$$

If we consider it in bending direction (torque on d_1 or d_2):

$$d_3 \times F = (\rho I + (x \cdot mass) \cdot (l_s)^2)\dot{\omega}$$

Therefore the bending moment of inertia will be $I + (x \cdot mass) \cdot (l_s)^2$. This equation is the same with Equation 3.16 where n is approximated by $-(x \cdot mass)\ddot{r} \cdot l_s$.

Since we have find a approximated n, we can solve Equation 3.5.

3.5 Dynamics Integration

From above section, we know the approximation of Equation 3.5 will have this form:

$$\frac{\partial I\dot{\omega_i}}{\partial t} = \tau_{total,i}$$

where τ_{total} is

$$\tau_{total,i} = F_{T,i} \times d_3 + \tau_i^{stiffness} + \tau_i^{external} + \tau_i^{damping} \tag{3.17}$$

We use Spillmann's method to compute the stiffness torque$\tau^{stiffness}$. The detail is describe in the appendix.
$\tau^{damping}$ is

$$\tau^{damping} = -D_q \cdot \omega \tag{3.18}$$

where D_q is the damping constant. Then we can solve $\dot{\omega}_i$:

$$\dot{\omega}_i = \tau_{total,i} - \omega_i \times I\omega_i \tag{3.19}$$

Then integrate ω:

$$\omega_i^{t+h} = I^{-1}(\tau^t_{total,i-\omega_i \times I\omega_i})h + \omega_i^t \tag{3.20}$$

Here we use a method similar as Spillmann's semi-implicit method[12] but remove dimensions of position.

First, transform ω to quaternion velocity $\dot{q}$:

$$\dot{q}_i = \frac{1}{2}Q_i \begin{pmatrix} 0 \\ w \end{pmatrix} \tag{3.21}$$

where Q is the quaternion matrix,and q is the orientation of the segment i. Then we can integrate it on quaternion:

$$\hat{q}_i^{t+h} = \frac{1}{2}Q_i^t \begin{pmatrix} 0 \\ w_i^{t+h} \end{pmatrix} h + q_i^t \tag{3.22}$$

$$q_i^{t+h} = \frac{\hat{q}_i^{t+h}}{\|\hat{q}_i^{t+h}\|} \tag{3.23}$$

After this integration, we have the simulation result of hair strands.

CHAPTER 4

Hair Simulation

A whole hairstyle usually has tens of thousands of hair strands. We describe here how we apply strand dynamics model to hair simulation, and implement interaction between hair to hair and hair to other objects.

4.1 Hair Clustering

In order to simulate the motion of styled hair which have been designed, we cluster modeled hair strands to find relations between guide hairs and hair strand geometry.

We choose some guided strands from hair model first. Then we compute average distances E between a guide hair g and a strand geometry h as an error function:

$$E(g,h) = \frac{1}{N}\sum_{i=1}^{N} minDis(g, v_i) \tag{4.1}$$

where $minDis$ is the minimal distance between g and v_i. v_i is the i-th vertex of h, and n is the number of vertices of h. Then we use the K-means clustering algorithm in textbook [13] to find the cluster of hair model. There are two guide hair for each strand.

4.2 Guide Hair Interpolation with GPU

We use the vertex shader to accelerate our hair interpolation. From the Section 4.1 , we have got the relation of guide hair and hair geometry. In order to interpolate vertex between two guide hairs, we need the weights between guide hair for each vertex of geometry. We compute the weight as:

$$w(g,h) = \frac{1}{E(g,h)} \sum^{foreachg} \frac{1}{E(g,h)} \tag{4.2}$$

where g is a guide hair, and h is a strand geometry.

After we compute weight of each vertex of hair geometry for corresponding guide hair using Equation 4.2, we put them into texture coordinate of hair geometry. We also put the corresponding guide hair index into texture coordinate. The above tasks are completed in off-line. At each render step in runtime, we put the vertex data of guide hair which we simulated in strand simulation part into texture's rgb channel. Then we interpolate vertex between multiple guide hair in vertex shader, using the method in the GPU Gems for skinning.

4.3 Collision Detection and Response

We use the interaction method proposed by Choe [14] to handle collision detection and collision response, which handles collision detection and response with two cylinders. Their method uses the outer cylinder to handle damping effects between hair cluster and use the inner cylinder to avoid deep intersection between hair cluster. Different from their method, we use the 1D Swept Sphere Volume(SSV) as our bounding volume instead of cylinder for the concern of performance. We use a user-define inner SSV radius and a outer SSV radius from hair cluster. The collision detection algorithm for SSV is described in the textbook by Ericson[15].
The damping force and resist force are describes in Equation 4.3 and 4.4.

$$F_s(b_i, b_j) = K * P_{inner} * \|(r(b_i) - r(b_j))\| \tag{4.3}$$

where F_s is the resist force to push two clusters away, b is the bounding volume, K is the spring parameter, the P_{inner} is the penetration depth of inner SSV, which is generated from collision detection, and the rest is the force direction. r is the position of SSV.

$$F_{viscous}(b_i, b_j) = -D * P_{outer} * (\dot{r}(b_i) - \dot{r}(b_j)) \tag{4.4}$$

where $F_{viscous}$ is the viscous force, D is the damping parameter of hair interaction, the P_{outer} is the penetration depth of outer SSV, and the rest is the relative velocity to b_j. $\dot{r}(b_i)$ is the velocity of SSV.

CHAPTER 5

Experiment and Result

We compute the stiffness torque from Young's Modulus by using the method proposed by Spillmann[12]. The detail of Young's Modulus and stiffness torque is described in the appendix.

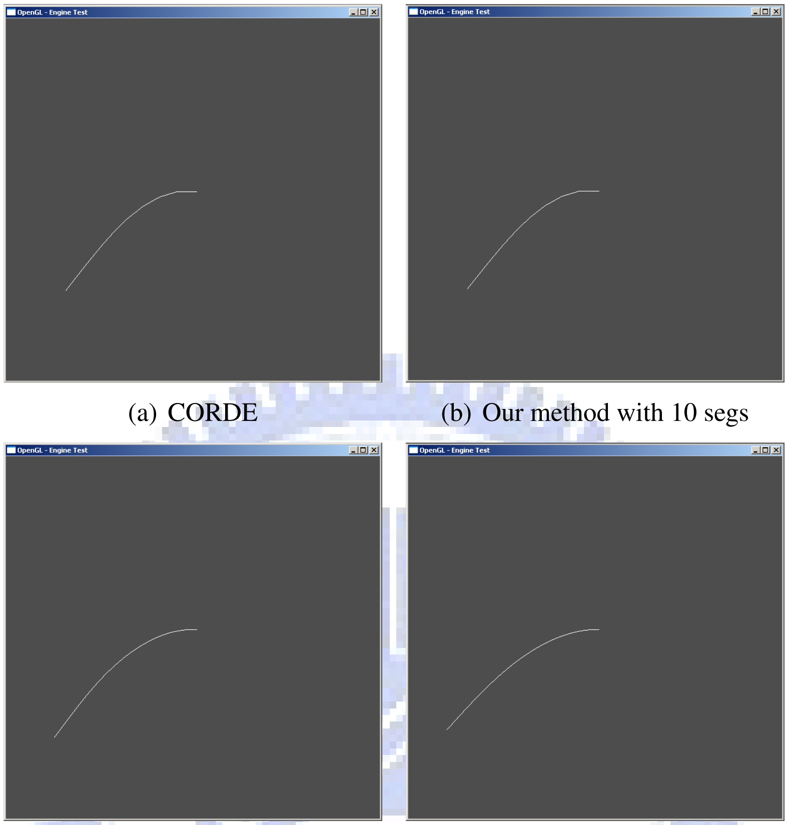
5.1 Single Strand Test

5.1.1 Statics Test

In order to check the accuracy of statics, we applied gravity to strands and record its final shape. The static result is compared with CORDE, which is shown as Fig 5.1. We set parameters of strand property as:

Radius	5 mm
Density	1300 kg/m^3 ρ

The result shows that the statics is almost the same with the CORDE. The force transport we use results a little force lost. The deform becomes smaller with more segments. But from the result, we know the lost of force is very small.



(a) CORDE

(b) Our method with 10 segs

(c) Our method with 50 segs

(d) Our method with 100 segs

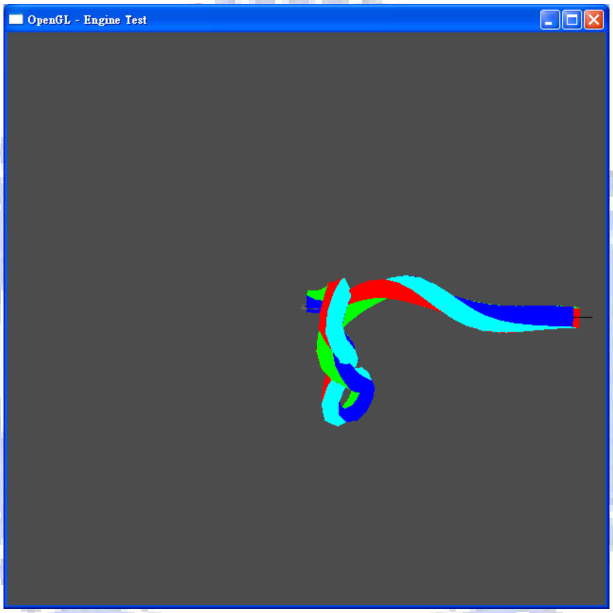
Figure 5.1: Statics Test: Young's modulus is set as 10MPa, and shear modulus is 5MPa. The error of our force transport may increase as the segment numbers increases.

5.1.2 Dynamics Test

Here we verify our dynamics model by releasing strand from a force, which is compared with CORDE. The result is shown in Fig 5.2. The CORDE result is colored by green, and our method is blue. There is a big difference between CORDE and our method. Remember our assumption in Section 3.4.1 that the deform of shape without a force will be absorbed fast in high stiffness object, which is one of properties of high stiffness object. As our assumption, we have the result of dynamics which looks like a high stiffness object. The curly strand dynamics is also shown in Fig 5.3

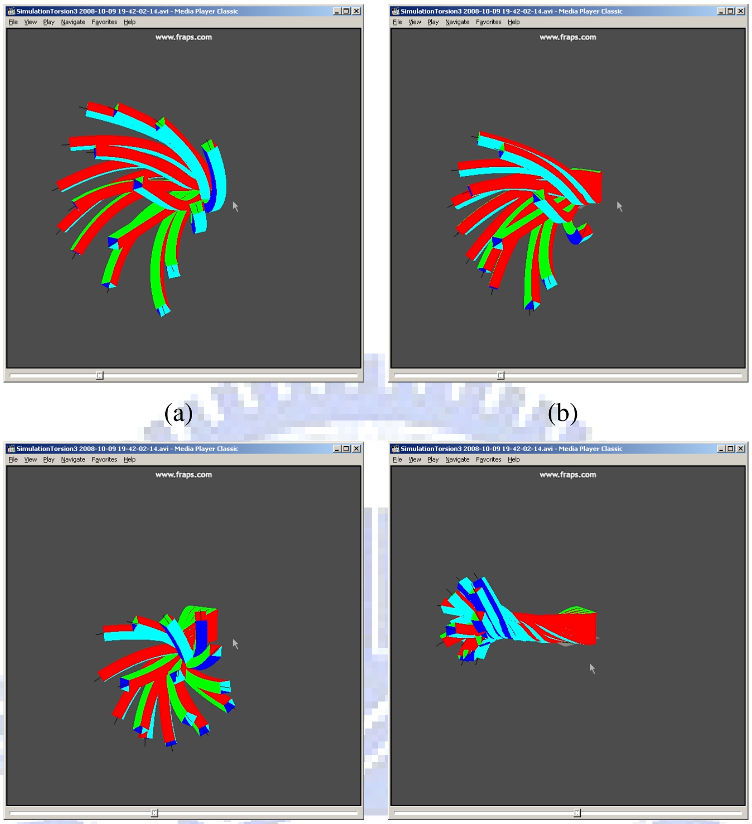
5.1.3 Curliness and Twist Test

In order to make sure our model has the advantage of the Cosserat Model that can deal with curliness. We use 50 segments to simulate a 1m strand. The Young's and shear modulus are 200KPa. The result is shown as Fig 5.4. The average FPS of simulation is 47.6.
Also we do a twist experiment with many strands. We control 16 strands with constrained tip to make them entwined. The average FPS is 20. 5.1.3



(a)

Figure 5.4: The Curliness Phenomena by twist: After we applied a twist torque to the whole strand and self-collision, we have a phenomena that the strand entwined.


SimulationTorsion3 2008-10-09 19-42-02-14.avi - Media Player Classic
File View Play Navigate Favorites Help
www.fraps.com

(a) (b)

(c) (d)

Figure 5.5: The strands entwined by rotating constrained point.

5.1.4 Performance

Our test environment is Intel Core Duo 2 6700 + GeForce 8800
The computing time of simulation and self-collision, and the stability test compared to CORDE are shown below. The stretching Young's Modulus in CORDE is set to 1MPa. We test the stability by applying gravity and release.
The stability of our method is better than CORDE. This is because we remove the stretching DOF and approximate of the Special Cosserat Theory. Removing the stretching DOF make it more stable when segment number increases and our assumption removes a local deformation by force which may make it unstable.

Computational time in one step

Segment #	Simulation Time(ms)	CORDE Simulation Time	Self-Collision Time(ms)
50	0.051	0.058	0.235
100	0.087	0.108	0.270
200	0.170	0.22	0.472
500	0.443	0.548	2.049

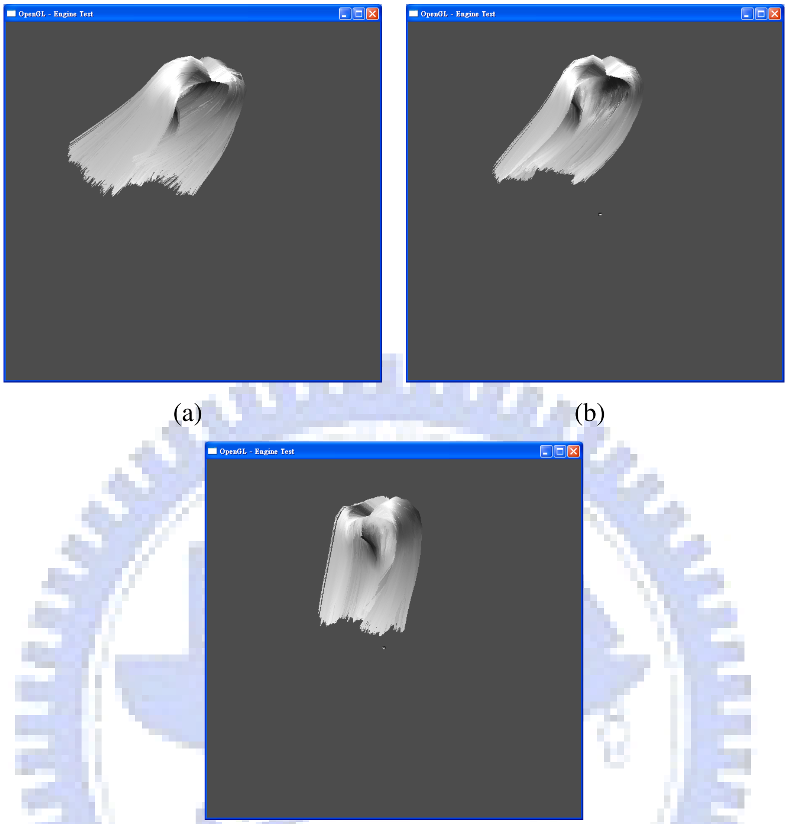
The stability of single strand with length 5m

Segment #	Young's Modulus (MPa)	Max Time Step(ms)	CORDE Max Time Step
50	1	7.0	3.1
50	10	2.4	1.6
50	100	0.7	0.5
100	100	0.3	0.2
200	100	0.15	0.1

5.2 Hair Simulation

5.2.1 Result

We use our strand model to simulate some animation of hair. Since there are not many self contact in hair simulation, we turn off the self-collision to get better performance. The Young's Modulus of the material parameters are set to 1MPa. The hair under wind is as Fig 5.6. The wind is modeled as a constant wind multiplied by a sin function. The wind force is set to $10^{-4}kg$.

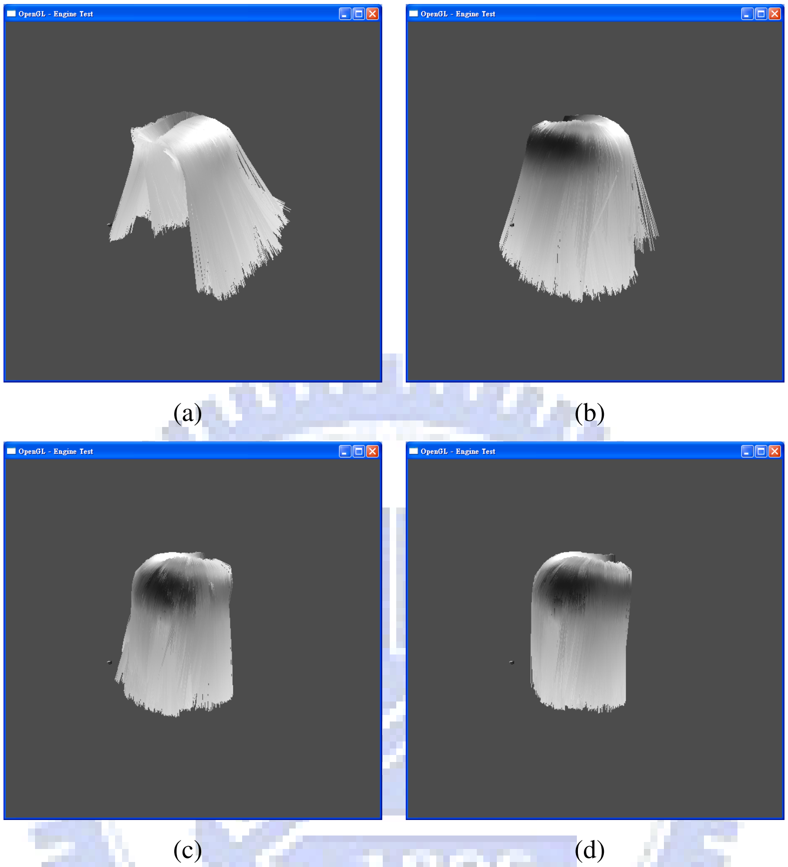


(a) (b)

(c)

Figure 5.6: Under Wind: hair deformed by a constant wind force.

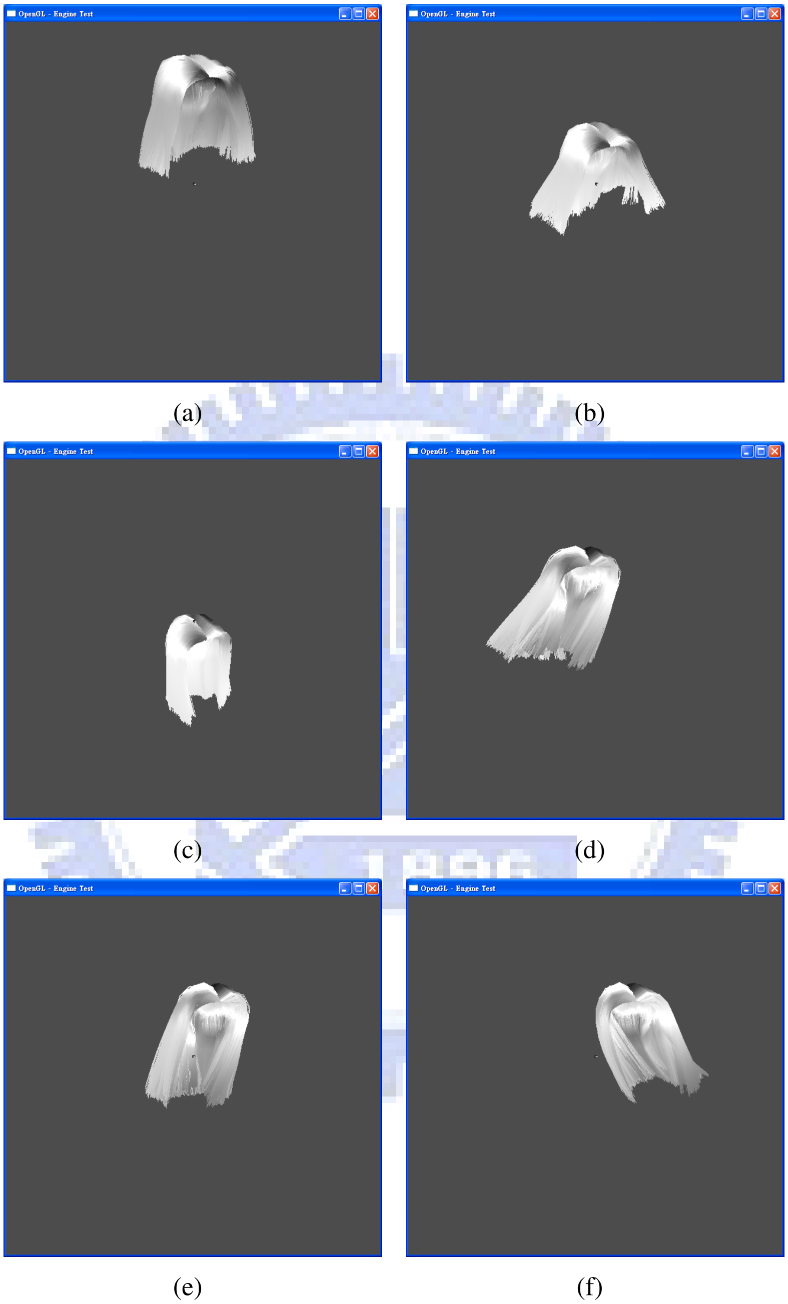
The dynamics results from head shaking and moving are as Fig 5.7 and Fig 5.8. The head dynamics is modeled by rigid body dynamics.



(a) (b)

(c) (d)

Figure 5.7: Simulated Hair Motion with Head Shaking: We simulate the head as a rotating rigid body. Then we get a dynamics of head shaking.



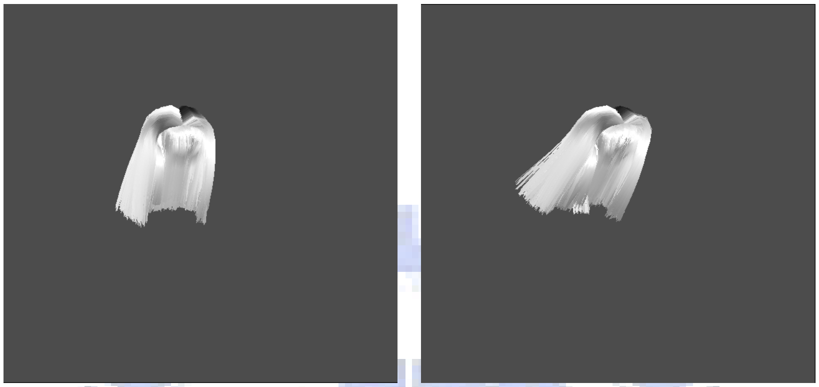
(a) (b)

(c) (d)

(e) (f)

Figure 5.8: Hair Moving: (a)(b)(c) the result that head moves vertically. (d)(e)(f)the result that head moves horizontally.

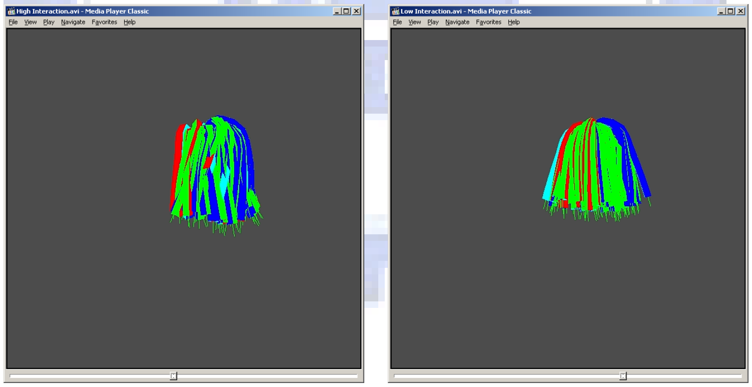
The difference motion of different Young's Modulus is shown as Fig 5.9. We can see the stiff hair style has less deformation when head moves.



(a) (b)

Figure 5.9: The difference between different stiffness in moving: (a) stiff (b) soft

The difference resulted by D in hair interaction is presented in Fig 5.10.

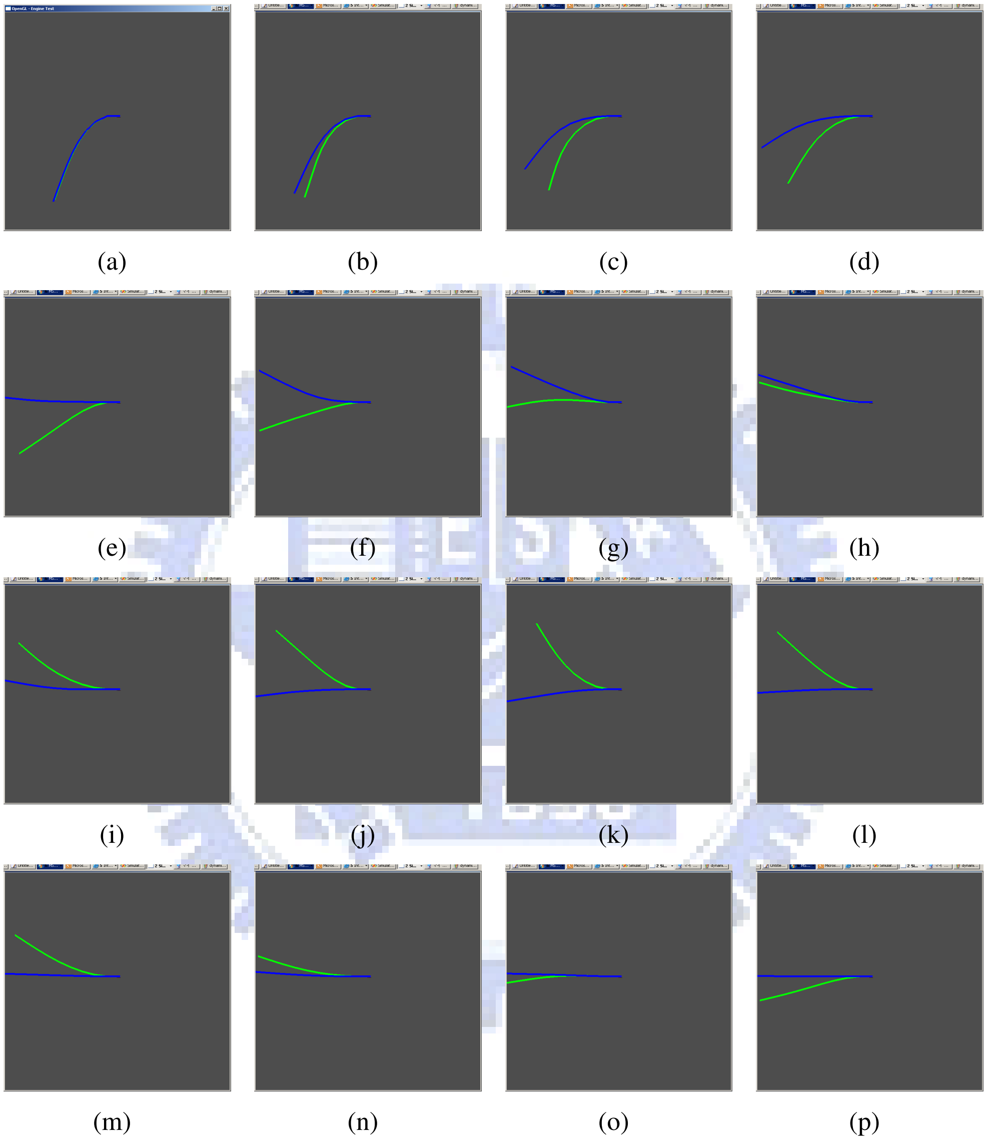

High Interaction.avi - Media Player Classic
File View Play Navigate Favorites Help
Low Interaction.avi - Media Player Classic
File View Play Navigate Favorites Help

(a) (b)

Figure 5.10: The difference of hair after head shaking between different interaction parameter (a)with $D = 2 \times 10^{-5}$ (b) with $D = 2 \times 10^{-6}$. Other interaction parameters are set to $K = 2 \times 10^{-6}, P_{inner} = 3mm$. The result with high viscous force is more cluttered.

5.2.2 Performance

We does above demo with simulating 100 guide hair of about 1000 segments. The time step of simulation and collision detection is $3ms$. The average simulation time is $0.64ms$, the average rendering time is 1.4ms, and the average collision detection and response time is $2.3ms$. The average FPS is 23.2



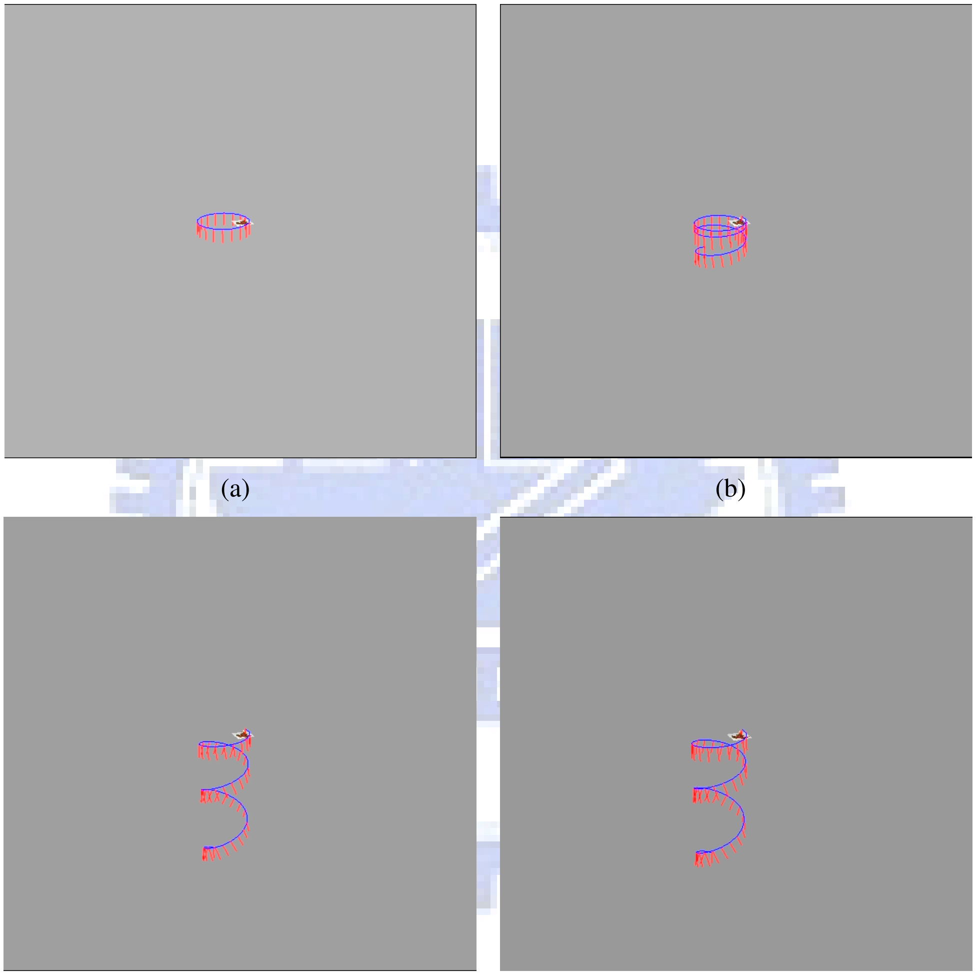
(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

Figure 5.2: Dynamics Result: Our method is blue. The CORDE is green. We can observe our result of dynamics is more stiff. This is because we assume the rest segments will move together in the simplification of Special Cosserat Theory. There is a big difference, but it is still acceptable in the hair simulation.



(a)

(b)

(c)

(d)

Figure 5.3: Curly Strand Dynamics

CHAPTER 6

Conclusion

We have proposed a dynamic model for simulating high stiffness model more efficiently. Our method is faster, more stable than traditional method, and also can model the curliness phenomena of strand. With our proposed method, we make the simulation of hair with full interaction able to be performed in real time. But there are still some problems needed to be solved.

Limitation

Our method is based on a shape analysis. It means the shape of simulated strand is needed to be continuous in time and space. It needs enough stiffness to keep strand continuous. In the case which we consider about, the shape is constrained strongly. But in low stiffness case, the shape may be very discontinuous in time. In this case, it will result some obvious artifacts in dynamics.

Future Work

We can observe from the statics experiment that the force transport method has still some force lost in each integration. As the segment number increases, the error increases. Besides, from the performance result, we can observe that the computational time for handling interaction is still the bottleneck in hair simulation. We would like to solve the hair interaction problem in the future.

CHAPTER 7

Appendix

Centrifugal Force

Before searching axial force, we consider the internal force generate by angular speed from Equation 3.7. We need to add it into the external force to compute $\dot{\omega}$. If we multiply a mass on Equation 3.7's both side, we get:

$$F = mass \cdot \ddot{r}_l = mass \cdot \int_0^l (\dot{\omega} \times d_3 + \omega \times (\omega \times d_3))ds$$

where $\omega \times (\omega \times d_3)$ is the centripetal force. What we want to find is the relation between force and angular acceleration, so we move it to the left side:

$$F - mass \cdot \int_0^l \omega \times (\omega \times d_3)ds = mass \cdot \int_0^l (\dot{\omega} \times d_3)ds \tag{7.1}$$

The left side is total force.

Stiffness

The potential energy of quaternion proposed by the CORDE:

$$V_k = \frac{l_i}{2}K_{kk}(B_k(q_j + q_{j+1}) \cdot \frac{1}{l_j}(q_{j+1} - q_j) - \hat{u_k})^2 \tag{7.2}$$

where

$$K_{11} = K_{22} = E\frac{\pi r^2}{4}, K_{33} = G\frac{\pi r^2}{2}$$

where E is the Young's Modulus for bending stiffness, and G is the Shear Modulus for torsional stiffness.

We can do symbolic differentiation to get the force τ_q on the quaternion:

$$\tau_q = \sum_{k=1}^{3} \frac{\partial}{\partial q_j} V_k$$

where

$$\frac{\partial}{\partial q_j} V_k = \frac{l_i}{2} K_{kk} \frac{\partial}{\partial q_j} \left(B_k \left(q_j + q_{j+1}\right) \cdot \frac{1}{l_j} \left(q_{j+1} - q_j\right) - \hat{u_k} \right)^2$$

$$= l_i K_{kk} \cdot \left(\frac{B_k}{l_j} \frac{\partial}{\partial q_j} \left(q_j + q_{j+1}\right)\left(q_{j+1} - q_j\right) \right) \cdot \left(B_k \left(q_j + q_{j+1}\right) \cdot \frac{1}{l_j} \left(q_{j+1} - q_j\right) - \hat{u_k} \right)$$

$$= l_i K_{kk} \cdot \left(\frac{B_k}{l_j} \left(\left(q_{j+1} - q_j\right) - \left(q_j + q_{j+1}\right)\right) \right) \cdot \left(B_k \left(q_j + q_{j+1}\right) \cdot \frac{1}{l_j} \left(q_{j+1} - q_j\right) - \hat{u_k} \right)$$

$$= l_i K_{kk} \left(\frac{B_k}{l_j} \left(-2q_j\right) \right) \left(B_k \left(q_j + q_{j+1}\right) \cdot \frac{1}{l_j} \left(q_{j+1} - q_j\right) - \hat{u_k} \right)$$

$$= -2 l_i K_{kk} \frac{B_k}{l_j} q_j \cdot \left(B_k \left(q_j + q_{j+1}\right) \cdot \frac{1}{l_j} \left(q_{j+1} - q_j\right) - \hat{u_k} \right)$$

But the mass to quaternion is a dense matrix,which is hard to inverse. We use [16] method to transform a quaternion force to a torque.

$$\begin{pmatrix} 0 \\ \tau \end{pmatrix} = \frac{1}{2} Q^T \tau_q \qquad (7.3)$$

where τ is the stiffness torque.

Bibliography

[1] Stuart S. Antman. *Nonlinear Problems of Elasticity*. Springer New York, 1995.

[2] Lay T. Weber A. Sobottka, G. Stable integration of the dynamic cosserat equations with application to hair modeling. In *International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2008.

[3] Kelly Ward, Florence Bertails, Tae-Yong Kim, Stephen R. Marschner, Marie-Paule Cani, and Ming C. Lin. A survey on hair modeling: Styling, simulation, and rendering. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):213–234, 2007.

[4] E. Plante, M.P. Cani, and P. Poulin. A Layered Wisp Model for Simulating Interactions inside Long Hair. In *Computer Animation and Simulation 2001: Proceedings of the Eurographics Workshop in Manchester, UK, September 2-3, 2001*. Springer, 2001.

[5] F. Bertails, TY Kim, MP Cani, and U. Neumann. Adaptive Wisp Tree: a multiresolution control structure for simulating dynamic clustering in hair motion. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 207–213. Eurographics Association Aire-la-Ville, Switzerland, Switzerland, 2003.

[6] Lentine M. G. Selle, A. and R. Fedkiw. A mass spring model for hair simulation. In *SIGGRAPH '08: ACM SIGGRAPH 2008 Papers*, 2008.

[7] S. Hadap and N. Magnenat-Thalmann. Modeling Dynamic Hair as a Continuum. In *Computer Graphics Forum*, volume 20, pages 329–338. Blackwell Synergy, 2001.

[8] J.T. Chang, J. Jin, and Y. Yu. A practical model for hair mutual interactions. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 73–80. ACM New York, NY, USA, 2002.

[9] R. Featherstone. *Robot Dynamics Algorithms*. Kluwer Academic Publishers, 1987.

[10] Pai D.K. Strands: Interactive simulation of thin solids using cosserat models. *Computer Graphics Forum*, 21:347–352(6), September 2002.

[11] Florence Bertails, Basile Audoly, Marie-Paule Cani, Bernard Querleux, Frédéric Leroy, and Jean-Luc Lévêque. Super-helices for predicting the dynamics of natural hair. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 1180–1187, New York, NY, USA, 2006. ACM.

[12] J. Spillmann and M. Teschner. Corde: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects. In *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 63–72, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

[13] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, 2006.

[14] B. Choe, M.G. Choi, and H.S. Ko. Simulating complex hair with robust collision handling. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 153–160. ACM New York, NY, USA, 2005.

[15] C. Ericson. *Real-Time Collision Detection*. Morgan Kaufmann, 2005.

[16] AL Schwab and JP Meijaard. HOW TO DRAW EULER ANGLES AND UTILIZE EULER PARAMETERS. In *Proc. IDETC/CIE 2006, ASME 2006 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, 2006.