Physically Based Real-time Animation of Curtains

Ying-Qing XU, Chiyi CHENG, Jiaoying SHI, Heung-Yeung SHUM

April 20, 2000

Technical Report
MSR-TR-2000-34

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Abstract

We propose a novel model for curtain real-time animation, which not only significantly reduces the time of rendering, but is also able to maintain visually appealing results. Conventional physically based models for cloth simulations are typically dense sheets of interwoven strings, with particles of finite mass at the intersections. While it seems to be an accurate model for cloth, it also implies that significant computation is required in solving a time-varying partial differential equation. Under certain circumstances, specific simplifications can be made. Our simplified curtain model is represented as a semi-rigid object. The longitudinal (warp) and latitudinal (woof) connections are processed separately. Furthermore, we provide a uniform treatment of internal and external forces.

Keywords: Physically Based Modeling, Virtual Reality, Curtains and Draperies, Real-time Animation, Collision Detection, Semi-rigid Object.

Physically Based Real-time Animation of Curtains

1 Introduction

Clothing simulation has always been a challenging topic in computer graphics. Both the textile industry and computer graphics are interested in how to describe and simulate the motions of clothing. A lot of fundamental techniques and standardized measurement systems have been developed in the textile industry to obtain empirical data to describe the properties of fabric[1]. A large amount of clothing simulations and animations have also been presented in computer graphics[2,3,4,5,6,7,8,9]. However, the real-time realistic clothing animation is still absent. One reason is that most mathematical models employed by these methods are too complex to be managed in real-time. Another reason is due to the expensive computation of collision.

In computer graphics, Weil pioneered the simulation of clothing using pure geometric equations and transformations[10]. Geometric techniques require user intervention and cannot be easily adapted to different materials or different cases[11,12, 13]. Hadap et. al. took a geometric and texture based approach to animate wrinkles on clothes[14].

Recent work on clothing animation is mainly based on physical laws. Physically based modeling remedies the problem of producing realistic animation by including forces, masses, strain energies, or other physical factors. Terzopoulos et al. first proposed a deformable model intended for flexible objects [2, 3]. By describing large deformations as the result of internal elastic force that counteracts deformation, external force and viscous damping force, he established a physical basis for the cloth model and solved the problem numerically. After Terzopoulos, lots of works have been done using the theory of elasticity[4,15,16]. Thalmanns and their team did extensive research in modeling clothes worn by fashion models, garments design[6,7,17], and especially in collision detection[18,19,20]. Breen et al. studied the rigid properties of different patterned fabrics and incorporated these properties into his particle system[5]. Particle system uses coupled particles connected by strings to model clothing through a minimized energy function. Though the works listed above differ in their representation, numerical solution method or collision detection methods, most of them can be categorized as the mass-spring model. The classic mass-spring model is composed of thousands of particles, each particle interacting with at least 8 of its neighbors to attain realistic cloth, resulting in a large number of iterations to converge to the equilibrium state. Some of the work used finite element method that is also time consuming[2,9].

On the other hand, physically based modeling usually involves solving a partial differential equation(s). A good deal of efforts have been devoted to real-time numerical solution of the equations[21,22]. The simulation time has been greatly reduced with Baraff' s implicit integration method[22]. However, it is still impossible to generate clothing animations in real-time. Breen' s SGD

(stochastic gradient descent) method required three days to one week to minimize the energy function on an IBM RS6000. Even in Baraff's implicit integration method, it still requires an average of 10 seconds to calculate a frame of 4530 vertices on a SGI Octance R10000 195MHz processor.

As we know, appearance is generally more interesting and important than physical accuracy in computer graphics. Therefore, it is possible to reduce the computation time if we can correctly simplify the physical model. We observe that a simplified model is adaptable for a class of special cloths, e.g., curtains or draperies. We have developed an efficient physically based model to overcome the performance issues in the previous models. After the decoupling of the longitudinal (warp) and latitudinal (woof) connections, we can process them separately. Our method is very simple and works well with non-iterative procedures, e.g. it can generate animations of two curtains of 1760 nodes with collision detection in real-time (see Table.1 in section 7) on a common PC (Pentium III 400MHz CPU without hardware acceleration).

The rest of the paper is organized as follows. In section 2, we give a brief overview of our method's architecture, Section 3 describes the specifics of the forces. Section 4 derives how we separate the longitudinal and latitudinal connections, and then process them separately. Section 5 describes a fast collision detection technique. Constrained initialization is discussed in section 6. Conclusions are drawn in section 7 with simulation results.

2 Overview of the Architecture

Our method consists of six components, namely initialization, forces calculation, longitudinal (warp) calculation, latitudinal (woof) calculation, collision detection, and smoothing of the mesh. In this section, we will give a brief introduction of our model.

2.1 Notation and Geometry

In our model, a curtain is divided into a *n by m* rectangular grid, it gives the internal woof and warp structure of a woven cloth naturally (Fig.1). The node in the i^{th} row and j^{th} column is represented as (i, j) with its world coordinate $\vec{x}_{i,j}$, the mass $m_{i,j}$, and the velocity $\vec{v}_{i,j}$ ($\vec{v}_{i,j}$, $\vec{x}_{i,j} \in R^3$). Force $\vec{f}$ exerts a force $\vec{f}_{i,j} \in R^3$ on node (i, j). Real-world curtain is cut from flat sheets of material and tends to resist deformations away from this initial flat state. We capture the resting state of the curtain by assigning each node (i, j) to a static coordinate $(u_{i,j}, v_{i,j})$ in the plane. The length

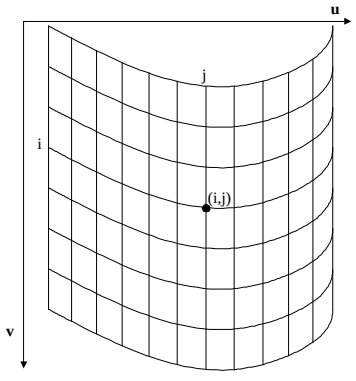
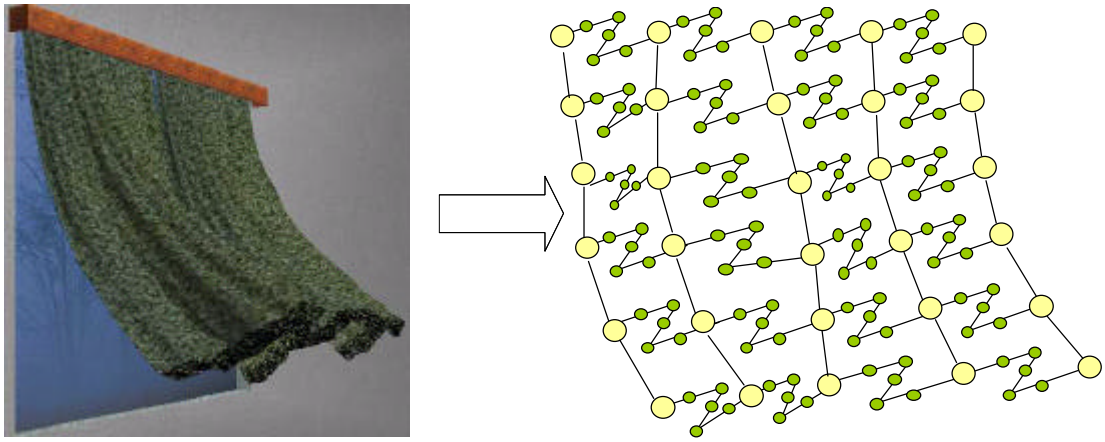


Fig.1 curtain grid

between node (i, j) and its upper neighbor $(i-1, j)$ in the free state is presented as $l^{v}_{i,j} = v_{i,j} - v_{i-1,j}$, and the length between node (i, j) and its left neighbor $(i, j-1)$ in the free state is presented as $l^{u}_{i,j} = u_{i,j} - u_{i,j-1}$.

2.2 Semi-rigid rods



(a) (b)

Fig.2 Semi-rigid

Investigating a curtain moving in airflow (Fig.2-a), we observe that if we divide it into a number of segments in its longitudinal direction, each segment sways like a pendulum except that its motion has no periodicity and it has more degrees of freedom (Fig.2-b). All the segments are connected together one by one composed of a piecewise smooth curve or spline, which is like a complex pendulum. We call it a semi-rigid object since it is deformable while at the same time it can preserve the total length. The semi-rigid object can also be applied in the latitudinal direction.

A semi-rigid rod cannot be stretched or compressed, but it can undergo some deformation such as bending. The distance (along the arc length) of each rod between node (i, j) and node $(i, j+1)$ (or node $(i+1, j)$) is fixed. It is determined by the resting state of the cloth, that is, the constant 2D coordinates of the two nodes: $(u_{i,j}, v_{i,j})$ and $(u_{i,j}, v_{i,j+1})$(or $(u_{i+1,j}, v_{i,j})$).

In comparison to most of the previous works, semi-rigid rods can better preserve the overall shape of the cloth. Most elastic models try to perform local optimization to find the local equilibrium state. Global optimization of the overall cloth is comparatively of less concern. This local method may work well in creating wrinkles and creases in cloth, but it cannot preserve the overall shape of the cloth. For example, the overall length of previously developed cloth models may lengthen a lot under external force of in-plane stretching. Provot[16] observed that abnormal deformations occur in the constraint points in classic models and try to solve the problem by limiting the elongation to 10%. Though measurement system of the textile industry proves that a

cloth experiences small elongation under this circumstance, we believe such elongation for most densely woven cloth is too small to be recognized by human eyes. Meanwhile, the semi-rigid property makes the internal force between two adjacent nodes easy to handle

2.3 Two Detached Directions

As mentioned above, we try to detach the longitudinal and latitudinal connections to simplify the physical model. In the latitudinal direction, the nodes and rods in the same column can be viewed as a complex pendulum that sways under the influence of the gravitational, airflow and damping forces. In the longitudinal direction, stiff multi-section rods connect adjacent nodes of the same row. The overall length of all the sections length is constant, but the length of each section may change according to the physical and geometrical state of the rods. In our method, the longitudinal connections are first calculated and then the latitudinal connections are added to form realistic cloth. Since these two are managed separately, we do not have to establish any differential equations or large numerical matrices. Instead, we can calculate the motion of each node one by one, which greatly reduce the computation time.

3 Forces

3.1 Unified Forces

Mass-spring models classify forces into two catalogs: internal force that counteracts deformation, and external forces that provokes cloths' motion. Internal forces are determined by elastically laws. In our model, the rigid property of the rods has released us from sticking with the complicated deformation laws and the annoying experiments for the elastic parameters. There are no elasticity functions in our model. However, the deformation properties are incorporated into the mass of each node. The mass of each node is determined by the cloth' s woven density, the elastic property of the yarns, and the area the corresponding rectangle covers. In our future work, we will try to quantify the correlation between the elastic property and the mass of our model.

As the rods connecting adjacent nodes cannot be elongated or compressed, the interacting forces between two adjacent nodes can be tension that resist the trend of elongation or compression. Either case can be determined by the relative velocity $\vec{v}(=\vec{v}_{i,j}-\vec{v}_{i,j-1})$ and relative coordinate $\vec{x}(=\vec{x}_{i,j}-\vec{x}_{i,j-1})$ of the two nodes at the end of each rod. If $\vec{v}$ is in the same direction as $\vec{x}$, the force is to resist compression (in fig. 3, the angle $q_1 < p/2$); if $\vec{v}$ is in the opposite direction as $\vec{x}$, the force is to resist elongation (in fig. 3, the angle $q_2 > p/2$).

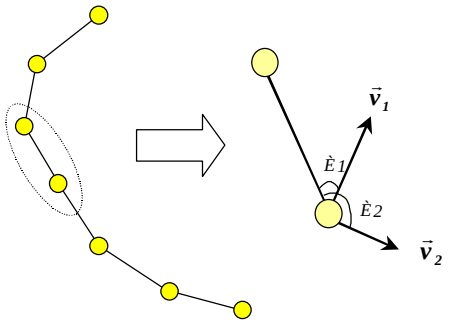


Fig.3 If the relative velocity of two connected nodes is $\vec{v}_1$ the internal force will resist compression; if relative velocity is $\vec{v}_2$, internal force will resist elongation.

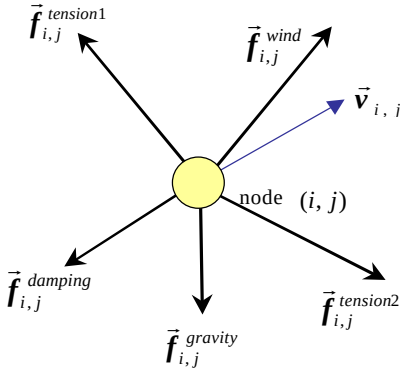


Fig.4 Analysis of the forces

External forces include wind, gravity, damping force, contacting force with surrounding objects, and so on. Since the internal forces are not relevant to each node' s displacement, they can be handled the same way as external forces. We can combine the internal forces and external forces into a net force vector $\vec{f}$. The acceleration of node (i, j) is simply:

$$\frac{d^2\vec{x}_{i,j}}{dt^2} = \frac{\vec{f}_{i,j}}{m_{i,j}} \tag{1}$$

Solving this ordinary differential equation, we can obtain the position $\vec{x}_{i,j}$ of each node at any instant, and thus, the motion of the cloth can be determined.

Fig. 4 lists some common forces in our method, they are

$\vec{f}_{i,j}^{\,tension1}$ is the internal force between (i, j) and its upper neighbor $(i-1, j)$;

$\vec{f}_{i,j}^{\,tension2}$ is the internal force between (i, j) and its lower neighbor $(i+1, j)$;

$\vec{f}_{i,j}^{\,wind}$ is the wind force;

$\vec{f}_{i,j}^{\,damping}$ is the damping force, which is in the opposite direction of (i, j)' s velocity;

$\vec{f}_{i,j}^{\,gravity}$ is the gravity of node (i, j).

3.2 Wind Force

In our simulator, we use a uniform wind force model that can take into account the porosity of the cloth and the viscidity of air. Suppose we have a predefined conical wind source: $\vec{f}_w$. (It can

exert force on the cloth in a parallel fashion if the top and bottom radii of the cone are the same, that is, a cylinder. We can also adjust which portion of the cloth is under the wind force by tuning the position of the wind source and radius of the cone.) We calculate the force acting on each nodes of the cloth by:

$$\vec{f}_{temp} = \frac{\vec{f}_w \bullet \vec{d}}{\left\|\vec{d}\right\|^2} \vec{d} \qquad (2)$$

$$\vec{f} = \frac{(1 - c_m m)\cos\Psi}{c_d \cdot (1 + \left\|\vec{d}\right\|^2)} \vec{f}_{temp} \qquad (3)$$

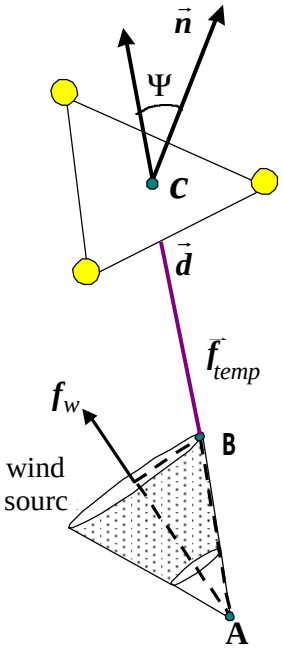


Fig.5 Wind source

where point **A** is the vertex of the cone; **C** is a node on the cloth; **B** is the intersection of edge **A C** and the bottom plane of the cone. $\vec{f}_w$ is the wind force at the center of the bottom plane and $\vec{f}_{temp}$ is the corresponding wind force putting forth by point **B**. $\vec{d}$ is the vector connecting **BC**. $m(0 \le m \le 1)$ is the porosity ratio which controls the amount of flow going perpendicularly through the cloth surface as a fraction of the free stream velocity normal to the cloth surface., Ψ is the angle between the direction of wind force $\vec{f}_{temp}$ and the mean normal of each node, (the mean normal is the normalized average normal vector of all the triangles connected to the node); c_u and c_d is the corresponding weight of m and $\left\|\vec{d}\right\|$, determined by our experiments. In addition, we add some random disturbance on the force acting on each node to simulate the unsteady property of airflow. These random turbulence forces yield good results.

3.3 Damping force

Curtain simulation is critically dependent on the damping forces to dissipate accumulating energy and to avoid oscillations. In previously developed models, a suitable damping force should accompany the internal spring force of each category. As we have provided a uniform treatment of the external and internal forces, we only need to define a simple viscous damping function on the assumption that the curtain is placed in free air. Air resistance is calculated by the following formula:

$$\vec{f}_r = \begin{cases} r \cdot c_r \cdot \vec{i}, & \left\|\vec{i}\right\| \le threshold \\ r \cdot c_r \cdot \left\|\vec{i}\right\| \cdot \vec{i}, & \left\|\vec{i}\right\| > threshold \end{cases} \qquad (4)$$

r is the density of air, which, in our case, is assumed to be constant by the precondition that the air is incompressible; c_r is the resistance coefficient; and the damping force can be linear or quadric according to the relationship between threshold and $\vec{i}$ of each node.

It is known that the values of c_r will be changed for distinct motion speeds. We assign nodes of different positions in cloth with different values of c_r . For nodes "inside" the cloth, that is, with four neighbors in the grid, are assigned smaller values of c_r ; for nodes with three neighbors on the border, their c_r values are larger; while for nodes on the square, that is, nodes with only 2 neighbors, their c_r values are the largest.

4 Decoupled Connections

Our simulation system has two phases: longitudinal connection and latitudinal connection. In this section we will introduce the two phases respectively. As we have stated before, the two connections are detached. Accordingly, the nodes connected by different connections are classified into two levels (They are distinguished by spheres of different colors in Fig.2. the first level is yellow and the second level is green). We will explain this in detail later.

4.1 Longitudinal Directions

Firstly, we calculate the longitudinal connection. Each longitudinal connection is treated like a complex pendulum (see Fig.6). Semi-rigid rods connect adjacent nodes of the same column. All the nodes constituting a complex pendulum are called nodes of the first level (In Fig.2 and Fig.6, they are represented by the yellow spheres.).

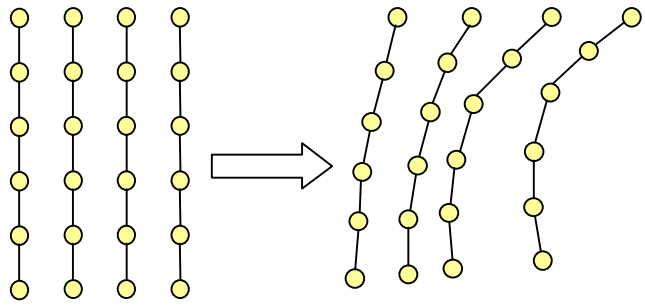


Fig.6 Longitudinal connections

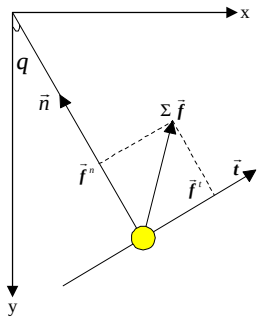


Fig.7 A pendulum

A pendulum is easy to model. But we still have to specify some points because the longitudinal direction is not a classical pendulum. Referring to Fig.4, for each first level node, we calculate the resultant force of all the external and internal forces: $\sum \vec{f}$. Then solving Eq.1, we get the position,

velocity and acceleration of each node. The processing is done according to the follow steps (suppose we are processing nodes of the j^{th} column):

1. $i = n$, n is the number of nodes in each column;
2. For node (i, j), do a coordinate transformation by transforming its world space coordinate $\vec{x}_{i,j}$ to local coordinate $\vec{x}'_{i,j}$, where the origin of the latter is (i, j)'s upper neighbor $(i-1, j)$, and the rod connecting (i, j) and $(i-1, j)$ in the local coordinate is in the $x'-y'$ plane.
3. Calculate the resultant force $\sum \vec{f}$ acting on node (i, j), and decompose it into three forces: $\vec{f}^{\,n}$ is the force parallel to the direction $\vec{n}$ of the rod, $\vec{f}^{\,t}$ is perpendicular to the direction of the rod, $\vec{f}^{\,z}$ is the force in the direction of the z axis (in Fig.7, for simplicity, we omit $\vec{f}^{\,z}$). $\vec{f}^{\,n}$ guarantees that (i, j) can move about in a sphere centered at $(i-1, j)$, with its radius being the length of the rod; $\vec{f}^{\,t}$ and $\vec{f}^{\,z}$ are in charge of the acceleration of the node. The coordinate $\vec{x}'_{i,j}$ is correspondingly decomposed into: $\vec{x}^{\,n}{}'_{i,j}$, $\vec{x}^{\,t}{}'_{i,j}$ and $\vec{x}^{\,z}{}'_{i,j}$.
4. According to $\vec{f}^{\,n}$, $\vec{f}^{\,t}$ and $\vec{f}^{\,z}$, calculate the corresponding acceleration, velocity and position, and the tension of the rod connecting (i, j) and $(i-1, j)$.

$$\vec{f}^{\,n} = m_{i,j} \cdot w_{i,j}{}^2 \cdot l^v_{i,j} \tag{5}$$

$$w_{i,j} \cdot l^v_{i,j} = \frac{d\vec{x}^{\,n}{}'_{i,j}}{dt} \tag{6}$$

$$\frac{d^2\vec{x}^{\,t}{}'_{i,j}}{dt^2} = \frac{\vec{f}^{\,t}}{m_{i,j}} \tag{7}$$

$$\frac{d^2\vec{x}^{\,z}{}'_{i,j}}{dt^2} = \frac{\vec{f}^{\,z}}{m_{i,j}} \tag{8}$$

$m_{i,j}$, $l^v_{i,j}$, $\vec{f}^{\,n}$, $\vec{x}^{\,n}{}'_{i,j}$, $\vec{x}^{\,t}{}'_{i,j}$, and $\vec{x}^{\,z}{}'_{i,j}$ have been defined before, $w_{i,j}$ is the angular velocity of the node.

5. to transform the local coordinate back to the world space coordinate.
6. $i = i - 1$, go back to step 2, till $i = 0$, that is, at the top of the complex pendulum.

The idea of a complex pendulum provides us a natural numerical solution listed above. There are always free ends in the longitudinal direction in any fabric, draperies, and clothing. The free end of a complex pendulum has internal forces with only one neighbor. Therefore, we can first calculate the motion of the free end first. Then, one by one, we can expand from the free end to its neighbors and finally all the nodes in the whole complex pendulum are processed. For example, in Fig.6, first we calculate the motion of the bottom nodes, then we process their upper neighbors, and progressively go upwards. From bottom to top, the whole grid is traversed. In this way, we can avoid the large sparse matrices. And most encouragingly, we do not have to worry about the integration methods, neither the steps of the integration or whether the integration will converge.

4.2 Latitudinal Directions

Second, we calculate the latitudinal connections. In the latitudinal direction, the multi-section rods connect adjacent first-level nodes of the same row.

We sum up the distance between two adjacent columns. If the sum exceeds the fixed distance $l^{u}_{i,j}$ defined in section 2.1, which means that the cloth is stretched in the longitudinal direction and that it should unfurl flatly, adjustments are made for the position of the nodes so that the distance between the two nodes is exactly the fixed distance. This is a hard constraint in that all the nodes should obey because the semi-rigid rod cannot elongate. This process can be viewed as a supplement to the first phase in determining position of the first level nodes.

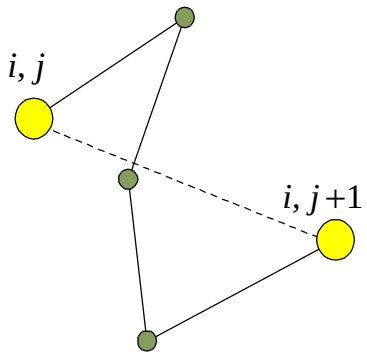


Fig. 8 Multi-section rods can only move in the normal plane defined by edge ((i,j), (i, j+1))

Now that the position of the first-level nodes have been obtained. They are used to determine the movement of the stiff multi-section rods. We suppose that each section of the rods is rigid, but the cross-point where two adjacent sections intersect can move flexibly. It' s like a gimmal connecting two rods. These cross-points are called nodes of the second level (in Fig.2 and Fig.8,, they are represented as green spheres).

As an example, the first-level node (i, j) and $(i+1, j)$ are connected by 4-section rods in Fig.8. We define the normal of edge $((i, j),(i, j+1))$ as the average normal of all the rectangles connected to edge $((i, j),(i, j+1))$. With the assumption that the multi-section rods will move in this average plane, the position of the multi-section rods can be obtained. Thus the position of the second-level nodes is determined.

4.3 Grouping of Nodes

By detaching the latitudinal and longitudinal connection, we can simplify the calculation

extensively. Accordingly, the nodes are grouped into two levels. The motion of the first-level nodes is mostly determined by the longitudinal connections, while the motion of the second-level nodes is determined by latitudinal connections.

5 Collision Detection

Our method can directly solve the equations because of the simplicity of the model. One of its drawbacks is that it cannot directly implement collision detection as collision constraints into the partial differential equation and solve them directly in the differential equation like previously developed methods[23]. We have to deal with collisions explicitly. That is, when any collisions are detected, we have to calculate collision response and adjust the nodes' position and velocity explicitly.

In curtain animation we have to deal with two kinds of collisions. One is self-penetrating, the other is collision with surrounding objects. Our cloth grid is further divided into a triangle mesh. The basic idea in both cases is the same----to find vertex-triangle and edge-edge penetration. However, in self-collision detection, since the cloth is a curved surface with no thickness, there are cases when the velocity of the nodes is very high, though in separate frame, there seems no intersection, but actually there has been penetration between two adjacent frames. This kind of collision is detected by detecting vertex-triangle and edge-edge intersection between previous and current positions in two adjacent frames.

Collision detection is very time-consuming as every vertex-triangle and edge-edge has to be checked. Optimizations can be made to speed up the computation. Here we employ two methods described by Provot's collision detection[24]: bounding box and surface curvature. By taking advantage of the bounding box and curvature of each triangle or group of triangles, we eliminate most vertex-triangle and edge-edge pairs that cannot intersect by simple geometrical tests, leaving very few pairs for further detection.

The collision response is calculated according to conservation of momentum. We also add sliding parameters to get a sliding effect.

A robust collision detection system for curtain animation should guarantee that each collision response will not create new collisions in other parts of the curtain, that is, collision consistency. We adopt the idea of collision zone from Provot[24]. The simulator groups interfering collisions into the same zone, and deal with collision response of a single zone. The method will iterate until no new collisions occur. It is very robust but sometimes it has to iterate a number of times in order to converge. It took two and a half hours to process a collision detection for 3000-polygon pair of trousers worn by a model. We cannot afford this time. Therefore, we only let our system iterate 2 times to get rid of most collisions, leaving some undetected. This still yields quite good result.

6 Longitudinal Connection Initialization

In section 4, we use multi-section rods for the longitudinal connection. How many rods should we use for each adjacent pair of the first level node? How long should each section be? In our current implementation, we create the number of rods randomly by the computer. However, our future work will try to incorporate image based modeling method to initialize the multi-section rods. The initial image of the cloth will decide the number of sections for each rod and the length of each section.

7 Conclusions and Results

We have developed a novel method that can generate curtain or drapery animation in real-time. Table 1 gives a performance summary of assorting animations.

Table 1. CPU time of our simulator
(On SGI Visual Workstation 320 with Pentium II 350MHz CPU)

Figure	no. vertices	Time/frame (CPU msec)
flag	240	<0.5
curtains	1760	10

With the elimination of elasticity theory, we have established a very efficient and straightforward model for curtains, flags, skirts, and draperies. The animation generated by our simulator is quite appealing. The computation time will increase linearly as the number of nodes increases, while in most models, the computation time will increase exponentially. The significant decrease of computation time makes our model useful for a rich variety of applications such as real-time animation in virtual reality, e-commerce and so on. Our future work will try to incorporate our model with human bodies and try to dress them with vivid clothes.

Fig.9-10 shows the result of real-time rendering. Moreover, we can interactively adjust the system parameters, such as the wind direction, the wind power, viewpoint, zoom in, zoom out, etc.

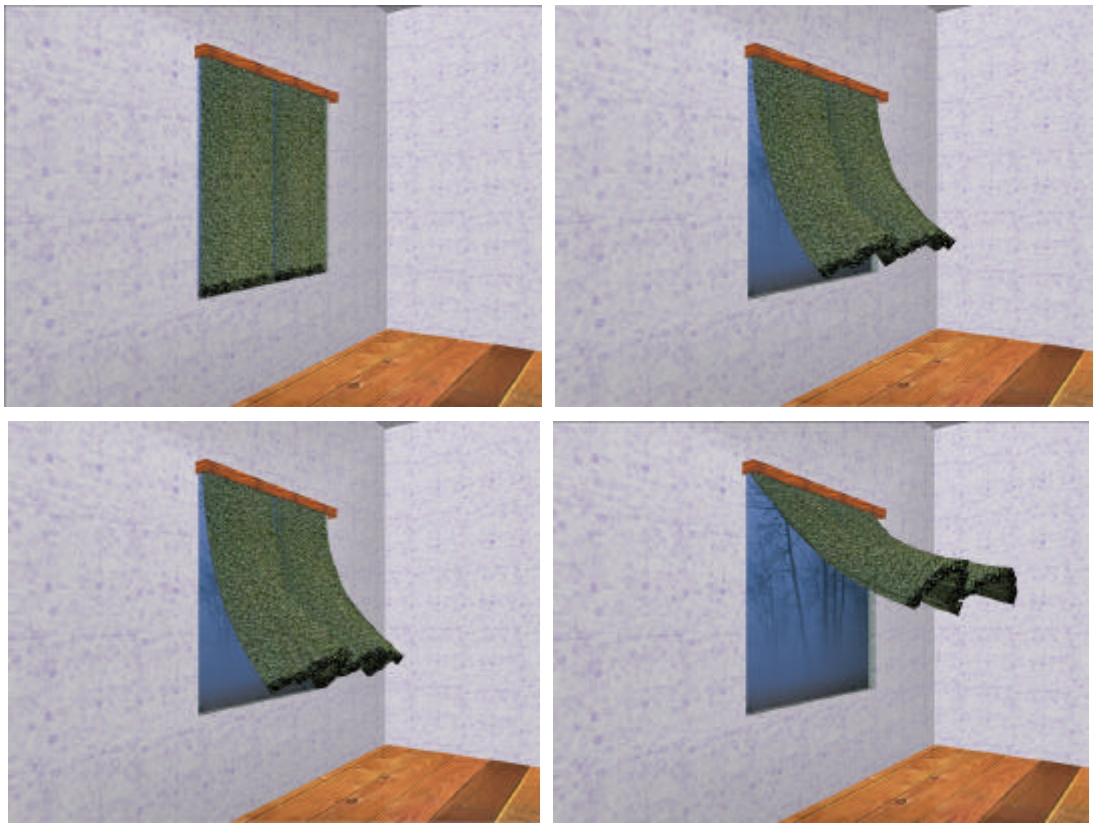


Fig.9 The pictures captured during real-time animation (two -curtains, with 1760 nodes)

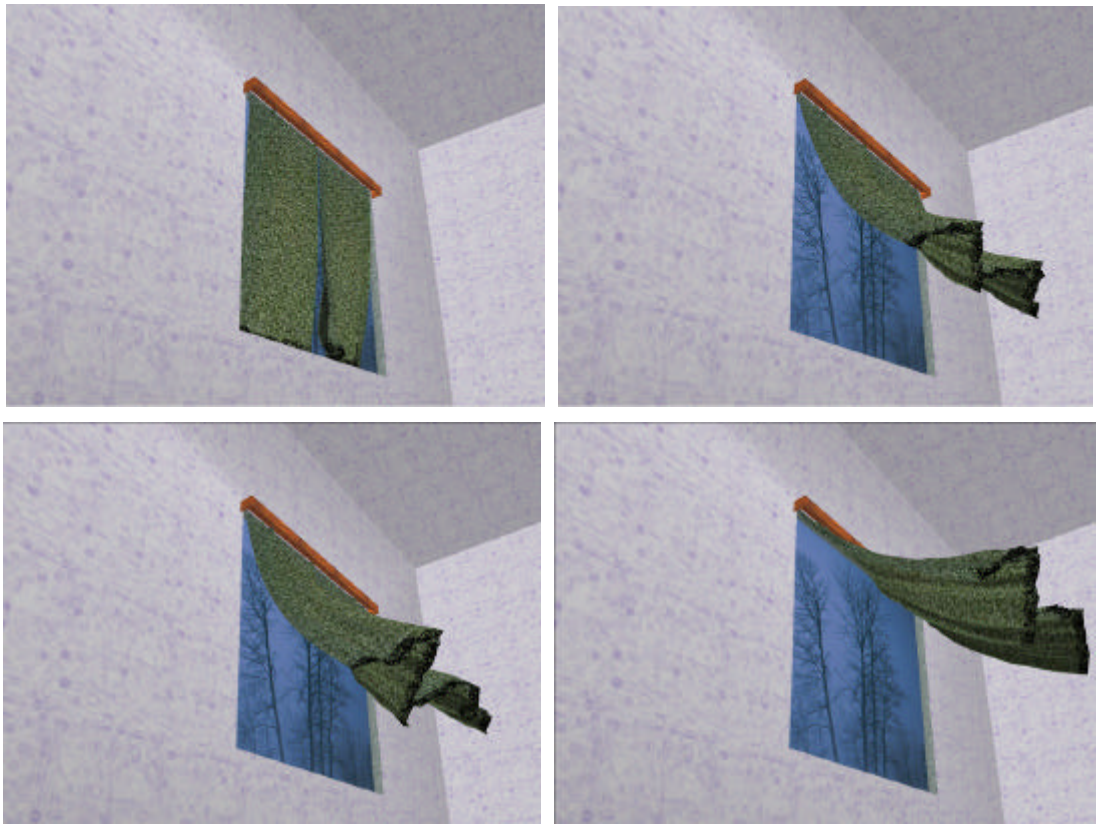


Fig.10 The pictures captured after users adjust the wind' s, view point' s and other parameters

Acknowledgements:

We are grateful to Dr. X. Tong, Dr. B. Guo, Dr. H. Bao, Dr. J. Li, and many other colleagues in CAD&CG China State Key Lab of Zhejiang University and Microsoft Research China for their helpful discussions and suggestions.

Reference:

1. Kawabata S., The Standardization and Analysis of Hand Evaluation, *Textile Machinery Soc.* Of Japan, Osaka, 1980.

2. Terzopoulos D., Platt J. C., and Barr A. H., Elastically Deformable Models, *Computer Graphics(SIGGRAPH'87)*, Vol. 21, pp.205-214, 1987.

3. Terzopoulos D., Fleischer K., Deformable Models, *The Visual Computer*, 4(6), pp.306-331, 1988.

4. Aono M., A Wrinkle Propagation Model for Cloth, *Proc. CG Int'l*, Springer-Verlag, Berlin, pp. 95-155, 1990.

5. Breen D. E., House D. H., and Wozny M. J., Predicting the Drape of Woven Cloth Using Interacting Particles, *Computer Graphics (Proc. SIGGRAPH'94)*, pp.365-372, 1994.

6. Carignan M., Yang Y., Magnenat Thalmann N., Thalmann D., Dressing Animated Synthetic Actors with Complex Deformable Clothes, *Computer Graphics (Proc. SIGGRAPH'92)*, 26(2), pp 99-104, 1992.

7. Volino P., Courchese M., Magnenat-Thalmann N., Versatile and Efficient Techniques for Simulating Cloth and Other Deformable Objects, *Computer Graphics (Proc. Siggraph'95)*, Aug. pp. 137-144, 1995.

8. Gay K. L., Ling L., Damodaran M., A Quasi-Steady Force Model for Animating Cloth Motion, *IFIP Trans. Graphics Design and Visualisation*, North Holland, pp 357-363, 1993.

9. Ling L., Damodaran M., and Gay K. L., A Model for Animating Cloth Motion in Air Flow, *TENCON '94. IEEE Region 10's Ninth Annual International Conference. Theme: 'Frontiers of Computer Technology'*. Proceedings of 1994.

10. Weil J., The Synthesis of Cloth Objects, *Computer Graphics (Proc. SIGGRAPH'86)*, Vol. 20, No. 4, pp.49-54, Aug.1986.

11. Hinds B. K. and McCartney J., Interactive Garment Design, *Visual Computer*, Vol. 6, pp. 53-61, 1990.

12. Hinds B. K., McCartney J. and Woods G., Pattern Developments for 3D Surfaces, *Computer-Aided Design*, Vol. 23, No. 8, pp. 583-592, Aug. 1991.

13. Ng H. N. and Grimsdale R. L., GEOFF —A Geometrical Editor for Fold Formation, Lecture Notes in *Computer Science Vol. 1024: Image Analysis Applications and Computer Graphics*, R. Chin et. Al., eds., Springer-Verlag, Berlin, pp. 124-131, 1995.

14. Hadap S., Bangerter E., Volino P., and Magnenat-Thalmann N., Animating Wrinkles on Clothes, *Proceedings of the conference on Visualization'99: Celebrating ten years*, pp. 175–182, 1999.

15. Okabe H., Imaoka H., Tomiha T., and Niwaya H., Three Dimensional Apparel CAD System, *Computer Graphics (Proc. SIGGRAPH'92)*, Vol. 26, No. 2, pp. 105-110, July 1992.

16. Deformation Constraints in A Mass-spring Model to Describe Rigid Cloth Behavior, *Graphics Interface*, pp. 147-155, 1995.

17. Volino P., Magnenat Thalmann N., Jianhua S., and Thal-mann D.. An Evolving System for Simulating Clothes on Virtual Actors, *IEEE Computer Graphics and Applications*, 16, pp. 42-51, 1996.
18. Yang Y. and Magnenat Thalmann N., An Improved Algorithm for Collision Detection in Cloth Animation with Human Body, *Proc. of Pacific Graphics, World Scientific Press*, Singapore, pp. 237-251, 1993.
19. Volino P. and Magnenat Thalmann N., Efficient Self-collision Detection on Smoothly Discretized Surface Animations Using Geometrical Shape Regularity, *Computer Graphics Forum, (EuroGraphics Proc.)*, Blackwell, UK, vol. 13, pp. 155-166, 1994.
20. Volino P., Magnenat Thalmann N., Jianhua S., and Thal-mann D., Collision and Self-collision Detection: Efficient and Robust Solutions For Highly Deformable Surfaces, In *6th Eurographics Workshop on Animation and Simulation*, pp. 55-65, Maastricht, Sep. 1995.
21. Baraff D. and Witkin A., Global Methods for Simulating Flexible Bodies, *Computer Animation Proc.*, Springer-Verlag, pp 1-12, 1994.
22. Baraff D. and Witkin A., Large Steps in Cloth Simulation, *Computer Graphics (Proc. SIGGRAPH'98)*, pp. 43-54, 1998.
23. Baraff D. and Witkin A., Dynamic Simulation of Non-Penetrating Flexible Bodies, *Computer Graphics (Proc. SIGGRAPH'92)*, 26(2), pp. 303-308, 1992.
24. Provot X., Collision and Self-collision Handling in Cloth Model Dedicated to Design Garments, *Eurographics Workshop on Animation and Simulation*, September 1997.