The Visual Computer

Collision avoidance in cloth animation

Jen-Duo Liu[1], Ming-Tat Ko[2], Ruei-Chuan Chang[1]

[1] Institute of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan, Republic of China
[2] 128 Yen-Chiu-Yuan Road, Sec. II Institute of Information Science, Academic Sinica, Taipei, 11529 Taiwan

For cloth modeling and animation, we use a physically based model to simulate the dynamic formation of folds, pleats, and wrinkles and the final static appearance of cloth draped over a rigid object. To simulate the behavior of the cloth and its final static appearance on the model, we propose a new collision and self-collision avoidance method to prevent penetration between the cloth and rigid objects and between parts of the cloth. At each time step, we enforce constraints over those grid points about to penetrate other objects. Our method is easier and more robust than conventional methods at representing interaction between the cloth and various objects.

Key words: Cloth animation – Physically based modeling – Iteration method, Collision detection and avoidance – Self-collision

e-mail: mtko@iis.sinica.edu.tw
Correspondence to: M.-T. Ko

1 Introduction

In recent years, cloth modeling and cloth animation have become an important subject in computer graphics. Cloth modeling includes the modeling of simple objects such as flags, paper sheets, table cloths, carpets, and stage screens, as well as that of garments or fabrics on the human body.

The images created by cloth modeling and cloth animation are closely related to the deformable model used. Several cloth deformable models have been proposed. Terzopoulos et al. (1987) employed elasticity theory to construct differential equations that can model the behavior of nonrigid curves, surfaces, and solids as functions of time. Their model not only represents the folds, pleats, and wrinkles of woven cloth, but also simulates the dynamic motion of the cloth resulting from external forces such as wind, gravity, repulsion, and other external forces. Carignan et al. (1992) also used Terzopoulos et al.'s (1987) modified elastically deformable model to animate clothes on actors in motion. Breen et al. (1992) simulated the draping behavior of cloth falling over a table. Their model represents cloth as particles denoting thread crossings placed on 2D grid points and several energy functions of these thread crossings. The draping and buckling behavior of cloth can be simulated by minimizing the energy functions as the particles interact with solid objects in the environment. By approximating the empirical mechanical cloth data, the model can represent various kinds of cloth, such as cotton or wool, realistically (Breen et al. 1994). However, the major drawback of this method is the lengthy computation it requires.

In animation design, it is crucial to simulate the responses of objects in collision. The collision simulation must be free of penetration. Moreover, it should be realistic in response behavior.

Many methods, ordinarily called collision-avoidance methods or collision-detection methods, have been proposed for collision simulation in rigid and deformable models. Many studies have focused on rigid models (Baraff 1989; Moore and Wilhelms 1988). Current rigid models can detect collisions efficiently and respond naturally. However, auxiliary forces have commonly been used in modeling collision avoidance for deformable materials, particularly cloth (Carignan et al. 1992; Lafleur et al. 1991; Terzopoulos et al. 1987). Lafleur et al. (1991) and Terzopoulos et al. (1987)

The Visual Computer (1996) 12:234–243

equipped each obstacle surface with a very thin repulsive force field. When an object is close enough to an obstacle surface, the auxiliary force is activated by the repulsive force field to prevent collision.

In general, methods using repulsive force fields guarantee penetration free simulation, but the repulsive force are somewhat artificial and cannot produce realistic response behaviors. Moreover, when two objects happen to be very close, the repulsive force can be extremely strong, and the objects will appear to "bounce" out of each other. Carignan et al. (1992) employ the law of conservation of momentum to govern the collision behavior of objects. When collisions are detected, to make the simulated collision behavior obey the law of conservation of momentum, an extra auxiliary force is computed and applied to the colliding point of the objects in the simulation. Kinetic energy is thus "dissipated", and the bouncing effect just mentioned avoided. This approach may yield acceptable results when simulating synthesized actors; however, it is difficult to simulate the static appearance of a cloth draping over an object, say, a table, using auxiliary forces.

When the auxiliary repulsive force is used, the cloth may take on an unrealistic shape and end up floating over, instead of touching, the object's surface. For the same reason, along the boundary of the table top, the cloth does not touch the table and appears rounded, which is different from the sharp right-angle found in real table cloths. When the auxiliary force in Carignan et al. (1992) is used, penetration cannot be avoided along the boundary of the table top. In short, it is hard to tune an appropriate auxiliary force to avoid penetration, as well as to obtain a realistic static appearance of cloth. Clearly, a new collision avoidance method is needed.

In this paper, the cloth model is based on the elastic deformation proposed by Terzopoulos et al. (1987). We use this model to simulate the draping behavior of cloth over objects (a discussion ensues). Natural effects caused by external forces, like wind and gravity, can be incorporated into the model to provide a compact representation of cloth behavior. To solve the penetration problems mentioned, we propose a new collision-avoidance scheme to simulate the collision behavior of cloth. Our collision-avoidance scheme is based on the following observation: the response force of cloth to a collision with an object is negligible. Thus, we enforce constraints over the colliding grid points to make them move along their respective colliding planes rather than bounce up. By using this scheme, not only can the cloth cling to objects of any shape without penetration, it can also be free of self-intersection.

The subsequent sections of this paper are organized as follows. In Sect. 2, the dynamic model of cloth is introduced. In Sect. 3, we describe the collision detection and avoidance algorithm. Experimental results and demonstration images are presented in Sect. 4. Finally, concluding remarks are given in Sect. 5.

2 The dynamic model of the cloth

Our work is based on the deformable model proposed by Terzopoulos et al. (1987) in which elasticity theory is employed. The dynamic motion of a point in the material coordinates of a deformable model is governed by the following *Lagrange equation* (Terzopoulos et al. 1987):

$$\frac{\partial}{\partial t}\left(\rho(\mathbf{a})\frac{\partial \mathbf{r}(\mathbf{a},t)}{\partial t}\right)+\gamma(\mathbf{a})\frac{\partial \mathbf{r}(\mathbf{a},t)}{\partial t}+\frac{\delta \varepsilon(\mathbf{r})}{\delta \mathbf{r}}(\mathbf{a},t)$$

$$=\mathbf{f}(\mathbf{r}(\mathbf{a},t),t). \tag{1}$$

The first term of the Lagrange equation is the force owing to the mass point's intrinsic motion, the second term is the damping force due to dissipation, and the third term is the elastic force due to the deformation of the body. The external force remains dynamically in balance with these internal forces.

By taking the variational derivative of the elastic potential energy, $\varepsilon(\mathbf{r})$ in Eq. 1, in terms of the shape of the cloth, we can approximate the elastic force by:

$$\mathbf{e}(\mathbf{r})=\frac{\delta \varepsilon(\mathbf{r})}{\delta \mathbf{r}}\approx \sum_{i,j=1}^{2}\left[-\frac{\partial}{\partial a_i}\left(\alpha_{ij}\frac{\partial \mathbf{r}}{\partial a_j}\right)\right.$$

$$\left.+\frac{\partial^2}{\partial a_i \partial a_j}\left(\beta_{ij}\frac{\partial^2 \mathbf{r}}{\partial a_i \partial a_j}\right)\right], \tag{2}$$

where

$$\alpha_{ij}(\mathbf{a},\mathbf{r}) = \eta_{ij}(\mathbf{a})(\mathbf{G}_{ij} - \mathbf{G}^0_{ij}) \quad 1 \le i,j \le 2 \tag{3}$$

and

$$\beta_{ij}(\mathbf{a},\mathbf{r}) = \xi_{ij}(\mathbf{a})(\mathbf{B}_{ij} - \mathbf{B}^0_{ij}) \quad 1 \le i,j \le 2. \tag{4}$$

In Eqs. 3 and 4, **G** and **B** are the metric and curvature tensors, respectively, and their definitions are given in Terzopoulos et al. (1987). η And ξ are two weight matrices (Terzopoulos and Fleischer 1988) used to model the elastic properties of the simulated object. The weight $\eta_{ij}(\mathbf{a})$ controls the stretch and growth deformation of the surface, and the weight $\xi_{ij}(\mathbf{a})$ controls the bend and curvature deformation of the surface.

The cloth model is discretized as a regular grid of $M \times N$ nodes spread on a unit square. Using the finite difference approximation method (Atkinson 1988), we can obtain the following linear system:

$$\mathscr{A}_t \boldsymbol{r}_{t+\Delta t} = \boldsymbol{g}_t, \tag{5}$$

where the $MN \times MN$ matrix $\mathscr{A}_t$ is known as the *stiffness matrix* and the elements of the $MN \times 1$ vector $\boldsymbol{r}_{t+\Delta t}$ are the positions of the cloth grid points at time $t + \Delta t$.

Notice that Eq. 5 actually consists of three equations, one each for the x, y, and z coordinates:

$$\mathscr{A}_t \boldsymbol{r}_{x,t+\Delta t} = \boldsymbol{g}_{x,t}, \tag{6}$$

$$\mathscr{A}_t \boldsymbol{r}_{y,t+\Delta t} = \boldsymbol{g}_{y,t}, \tag{7}$$

$$\mathscr{A}_t \boldsymbol{r}_{z,t+\Delta t} = \boldsymbol{g}_{z,t}. \tag{8}$$

The matrix $\mathscr{A}_t$ in Eq. 5 is a sparse, banded, diagonal matrix, This is because each cloth grid variable is related only to twelve other neighboring grid variables. By carefully selecting the physical parameters of the cloth, such as the mass and the damping factor, we can make $\mathscr{A}_t$ *diagonally dominant* (Atkinson 1988). Since $\mathscr{A}_t$ is diagonally dominant, efficient iteration methods, such as the Gauss-Seidel method, can be used to solve the linear system (Eq. 5). Using the Gauss-Seidel method, convergence of the iterative solution is guaranteed, and the error decreases quickly with each iteration. In our experiments, ten iterations were adequate to achieve single-precision accuracy. Together with the sparsity of $\mathscr{A}_t$, the time complexity of each step in our simulation is almost linear to the size of the cloth.

3 Dynamic motion with collision and self-collision avoidance

Our collision avoidance method for preventing penetration is quite intuitive. In each simulation iteration, we first detect whether any collisions occur and then set constraints on the colliding grid points so that they will slide along their respective colliding triangles. Each iteration of the simulation has two serial phases: the detection phase and the constraint phase. Assume that the simulation time is t.

- *Detection phase.* In the detection phase, we first simulate the behavior of the cloth, assuming that no obstacles are present; more precisely, we solve Eqs. 6–8 at time t for the x, y, and z coordinates independently to obtain intermediate positions for the grid points. Then, by comparing the intermediate grid point positions and those of the obstacles, all possible collisions in the iteration can be detected.
- *Constraint phase.* In the constraint phase, we constrain each colliding grid point's x, y, and z-coordinates according to the respective colliding triangle of each. We then solve the linear system at time t again, but this time with constraints on the colliding grid points.

In the following, we demonstrate how collisions are detected and penetration is avoided.

3.1 Collision detection

Two triangular patches may collide in one of the following ways: (1) a vertex of one triangle may touch another triangle as depicted in Fig. 1 (vertex/triangle type collision), and (2) a side of one triangle may touch the side of another triangle as depicted in Fig. 2 (side/side type collision). Several papers have discussed similar collision phenomena (Baraff 1989; Boyse 1979).

Almost all collisions between cloth and stationary rigid obstacles can be discovered by detecting vertex/triangle-type collisions. Exceptions occur

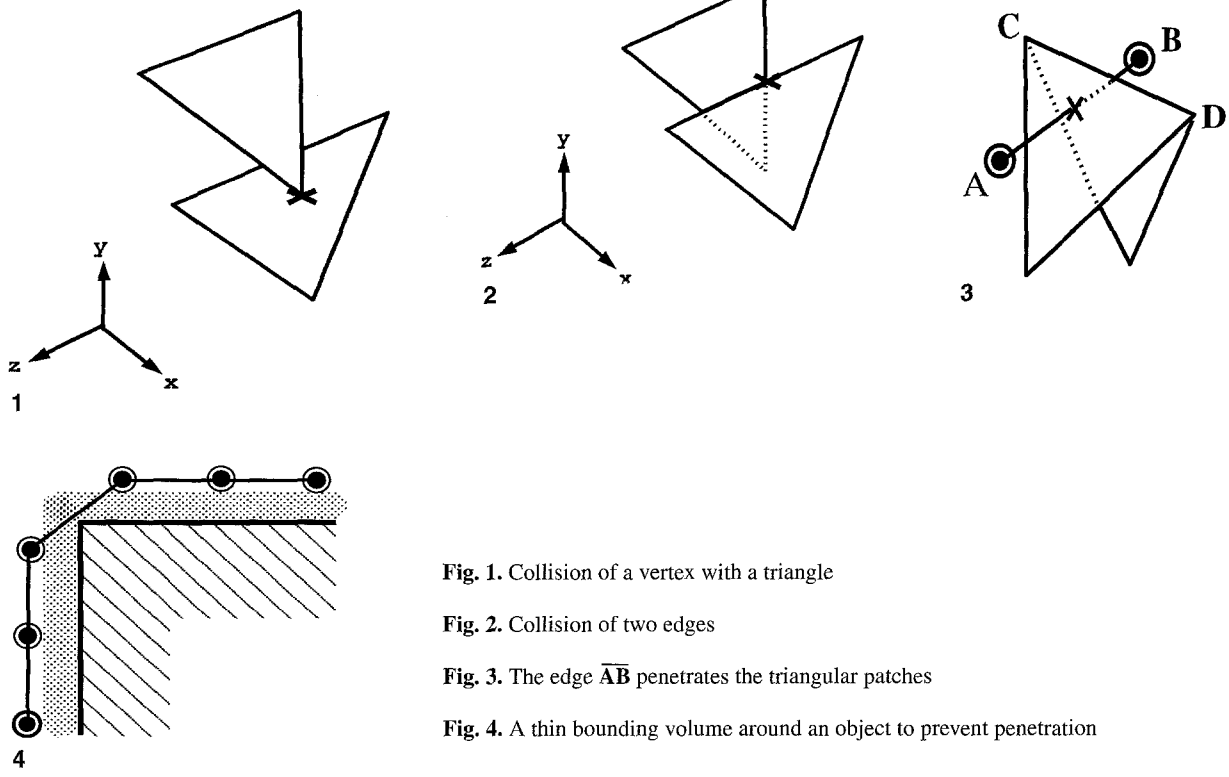


Fig. 1. Collision of a vertex with a triangle

Fig. 2. Collision of two edges

Fig. 3. The edge $\overline{AB}$ penetrates the triangular patches

Fig. 4. A thin bounding volume around an object to prevent penetration

when two triangular patches sharing an edge have a sharp angle relative to the resolution mesh. Such two triangular patches are called a sharp patch pair.

As shown in Fig. 3, edge $\overline{CD}$ is a shared edge of a sharp patch pair. Points **A** and **B** are close to, but do not penetrate, the triangular patches, but edge $\overline{AB}$ penetrates deeply across edge $\overline{CD}$. For shared edges of sharp patch pairs, we must use side/side collision detection. Since shared edges of sharp patch pairs in obstacles are rare, few side/side collision detections are needed. For shared edges of relatively obtuse patch pairs, slight penetration may occur. As shown in Fig. 4, we can use a very thin bounding volume around the stationary rigid obstacle and then constrain the colliding grid point over the newly created surface of the volume to avoid the slight penetration.

The self-collision detection test is more complicated because the cloth and the colliding object are moving. Thorough vertex/triangle collision detection and side/side collision detection are needed to detect all possible collisions.

Vertex/triangle-type collisions can be discovered by testing whether moving grid points pass through any moving triangles. Let $\mathbf{V}_1$, $\mathbf{V}_2$, and $\mathbf{V}_3$ be the vertices of a triangle and $\mathbf{P}$ be a grid point. These points move to $\mathbf{V}_1'$, $\mathbf{V}_2'$, $\mathbf{V}_3'$ and $\mathbf{P}'$ after Δt time, respectively. At time t, the points are at $\mathbf{V}_i(t) = \mathbf{V}_i + t(\mathbf{V}_i' - \mathbf{V}_i)$, $i = 1, 2, 3$ and $\mathbf{P}(t) = \mathbf{P} + t(\mathbf{P}' - \mathbf{P})$. Whether $\mathbf{P}$ penetrates the triangle $\Delta\mathbf{V}_1\mathbf{V}_2\mathbf{V}_3$ and if so, *when* it does, can be determined by solving:

$$(\mathbf{P}(t) - \mathbf{V}_1(t)) \cdot ((\mathbf{V}_2(t) - \mathbf{V}_1(t)) \times (\mathbf{V}_3(t) - \mathbf{V}_1(t))) = 0. \quad (9)$$

For the side/side type of collision, we test whether a collision occurs between any two moving sides of the triangles. Let $\mathbf{P}_i$, $\mathbf{Q}_i$, $i = 1, 2$, be the end points of two sides that move to $\mathbf{P}_i'$, $\mathbf{Q}_i'$, $i = 1, 2$, respectively. Whether these two sides collide

and if so, *when* they do, can be determined by solving:

$$(\mathbf{Q}_2(t) - \mathbf{Q}_1(t)) \cdot ((\mathbf{P}_1(t) - \mathbf{Q}_1(t)) \times (\mathbf{P}_2(t) - \mathbf{Q}_1(t))) = 0. \quad (10)$$

This collision detection test is time consuming since a polynomial equation of degree 3 must be solved for each vertex/triangle pair and each side/side pair. A simple heuristic method is to run several simplified iterations at each time step until no more collisions are detected. The method includes simplified detection and constraint phases. The simplified detection phase only tests whether any side intersects any triangle at time $t + \Delta t$. The simplified constraint phase sets a constraint on each grid point on detected colliding triangles or sides, then solves the constrained linear system to find intermediate grid-point positions for the detection phase of the next simplified iteration. As shown in Fig. 5, one simplified detection phase cannot detect colliding triangles that have passed completely through each other at time $t + \Delta t$. However, as long as penetration has occurred, at least one edge that passes through a triangle can be detected. Taking Fig. 5 as an example, although the edge $\overline{\mathbf{P}_i\mathbf{P}_j}$ is moved completely through the triangle at time $t + \Delta t$, the penetration by $\overline{\mathbf{P}_k\mathbf{P}_i}$ can be discovered. By applying the simplified iteration several times, undiscovered collisions are finally found. Since the displacement of the grid points in a single time quantum is very small, in practice, no collisions are detected after two runs, and usually one run is enough.

The collision-detection processes reduce the efficiency of the whole cloth-animation process. If the cloth has n grid points, and the obstacle is composed of m triangles, then $O(nm)$ time is needed to detect collisions with the obstacle, and $O(n^2)$ time is needed to detect collisions between parts of the cloth. Several methods, such as the bounding-box and bounding-volume methods, have been developed to improve the efficiency of the calculations (Moore and Wilhelms 1988). The bounding-volume method is implemented in our cloth simulation because the cloth is evenly spread and most vertex/triangle pairs need not be tested. Although the asymptotic time complexity is still as great as before, the efficiency is improved.

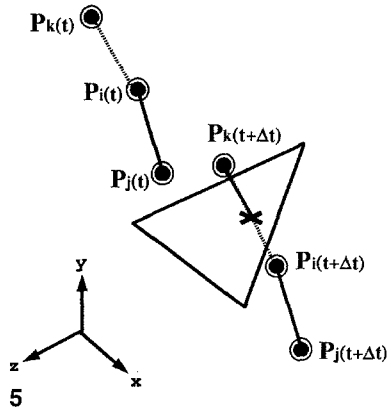
3.2 Collision avoidance

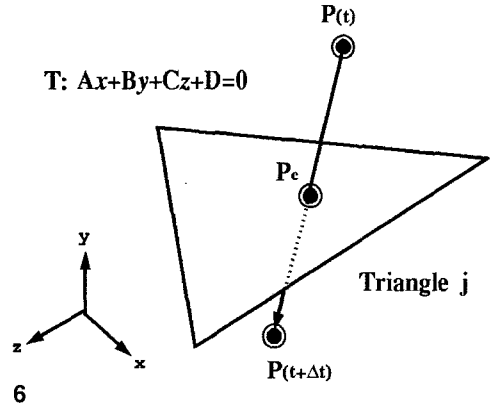
There are various ways in which objects collide, and the behaviors of cloth colliding with a rigid obstacle and with parts of itself are quite different. For this reason, we prevent these two types of collision with different avoidance schemes.

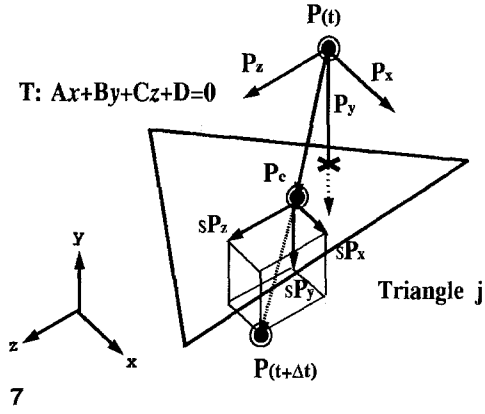
We now describe the collision avoidance scheme for cloth and rigid obstacles. Let $\mathbf{P}_t$ and $\mathbf{P}_{t+\Delta t}$ be the positions at time t and $t + \Delta t$ of a colliding grid point $\mathbf{P}$ obtained in step 1, and let the triangular patch $\mathbf{T}$ of the obstacle penetrated by $\mathbf{P}$ be contained in the plane defined by $\mathbf{A}x + \mathbf{B}y + \mathbf{C}z + \mathbf{D} = 0$, as shown in Fig. 6. Let $\mathbf{P}_c$ be the penetrating point of $\overrightarrow{\mathbf{P}_t\mathbf{P}_{t+\Delta t}}$. By intuition, in the simulation the grid point $\mathbf{P}$ will collide with triangular patch $\mathbf{T}$ at $\mathbf{P}_c$ and then slide along the triangular patch in the direction of projection of $\overrightarrow{\mathbf{P}_t\mathbf{P}_{t+\Delta t}}$ onto $\mathbf{T}$ and $\mathbf{P}$ at time $t + \Delta t$ is close to the projection of $\mathbf{P}_{t+\Delta t}$, denoted as $\mathbf{P}'_{t+\Delta t}$. Our collision avoidance method follows the intuition. To ensure that the position of $\mathbf{P}$ at time $t + \Delta t$ is on the plane of colliding triangular patch, two of the coordinates of $\mathbf{P}$ are chosen to be calculated independently while solving Eq. 5 by the Gauss-Seidel method, and the remaining one depends on the chosen two by the defining equation of the colliding plane. In the following, we describe the method for choosing the two coordinates to simulate independently so that $\mathbf{P}$ at time $t + \Delta t$ is close to $\mathbf{P}'_{t+\Delta t}$.

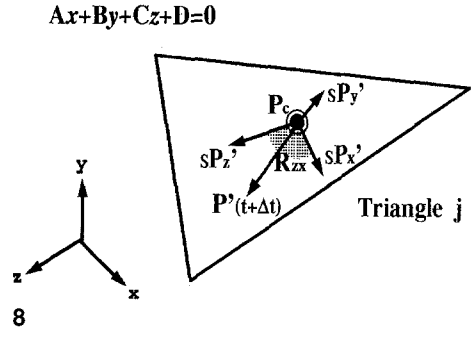
Let $\vec{\mathbf{P}_x}$, $\vec{\mathbf{P}_y}$, and $\vec{\mathbf{P}_z}$, be the x, y, and z components of $\overrightarrow{\mathbf{P}_t\mathbf{P}_{t+\Delta t}}$ (Fig. 7). The projection of $s\vec{\mathbf{P}_x}$, $s\vec{\mathbf{P}_y}$, and $s\vec{\mathbf{P}_z}$, $s \geq 0$ onto the plane containing the colliding triangular patch with $\mathbf{P}_c$ as its origin partitions the plane into three portions in general as shown in Fig. 8. The projection of $\overrightarrow{\mathbf{P}_t\mathbf{P}_{t+\Delta t}}$ is onto one of the three portions.

In the simulation phase, we first determine onto which portion $\overrightarrow{\mathbf{P}_t\mathbf{P}_{t+\Delta t}}$ projects. Consider the case that the portion is $\mathbf{R}_{zx}$, the portion bounded by projection of $s\mathbf{P}_z$ and $s\mathbf{P}_x$. The cases for $\mathbf{R}_{xy}$ and $\mathbf{R}_{yz}$ are symmetric. To guarantee that the position of $\mathbf{P}$ at time $t + \Delta t$ is on the same portion of the projection of $\overrightarrow{\mathbf{P}_t\mathbf{P}_{t+\Delta t}}$, the portion $\mathbf{R}_{zx}$, we use the









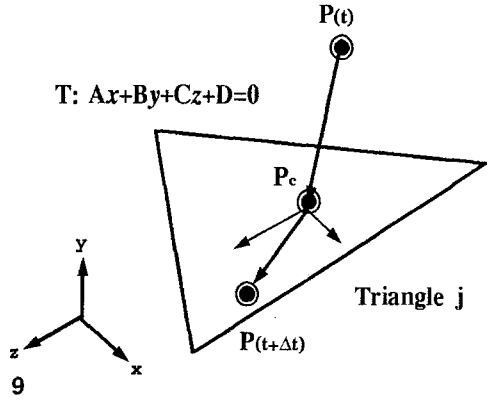


Fig. 5. A collision may be missed by the simple heuristic method

Fig. 6. Collision between a grid point and an obstacle

Fig. 7. Decomposing the moving vector for setting constraints

Fig. 8. The projection of vectors onto the plane $\mathbf{A}x+\mathbf{B}y+\mathbf{C}z+\mathbf{D}=0$

Fig. 9. The colliding point lies on the triangle without penetration

following method to constrain the moving direction of point **P**. That is, while solving Eq. 5 by the Gauss-Seidel method, the z and x coordinates of **P** are calculated independently, but the y coordinate is constrained to satisfy $\mathbf{A}x + \mathbf{B}y + \mathbf{C}z + \mathbf{D} = 0$. The resulting z and x coordinates of **P** are almost equal to those of $\mathbf{P}_{t+\Delta t}$, since they are calculated independently. Thus, the position of **P** at time $t + \Delta t$ is close to $\mathbf{P}'_{t+\Delta t}$, the projection of $\mathbf{P}_{t+\Delta t}$, and on the portion $\mathbf{R}_{zx}$.

To deal with some special cases, the initial guess of the Gauss-Seidel method is set to $\mathbf{P}_c$ for colliding grid point **P** and to $\mathbf{P}_t$ for the other points. We rewrite the iteration steps of the Gauss-Seidel

method as follows:

$$
\begin{aligned}
x_i^{(m+1)} &= \begin{cases} \frac{1}{a_{ii}}\{g_{xi} - \sum_{j=1}^{i-1} a_{ij}x_j^{(m+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(m)}\} & \text{if } x_i \text{ has not been constrained} \\ -\frac{1}{A_j}(\mathbf{B}_j y_i^{(m)} + \mathbf{C}_j z_i^{(m)} + \mathbf{D}_j) & \text{if } x_i \text{ has been constrained} \end{cases} \\
y_i^{(m+1)} &= \begin{cases} \frac{1}{a_{ii}}\{g_{yi} - \sum_{j=1}^{i-1} a_{ij}y_j^{(m+1)} - \sum_{j=i+1}^{n} a_{ij}y_j^{(m)}\} & \text{if } y_i \text{ has not been constrained} \\ -\frac{1}{B_j}(\mathbf{A}_j x_i^{(m+1)} + \mathbf{C}_j z_i^{(m)} + \mathbf{D}_j) & \text{if } y_i \text{ has been constrained} \end{cases} \\
z_i^{(m+1)} &= \begin{cases} \frac{1}{a_{ii}}\{g_{zi} - \sum_{j=1}^{i-1} a_{ij}z_j^{(m+1)} - \sum_{j=i+1}^{n} a_{ij}z_j^{(m)}\} & \text{if } z_i \text{ has not been constrained} \\ -\frac{1}{C_j}(\mathbf{A}_j x_i^{(m+1)} + \mathbf{B}_j y_i^{(m+1)} + \mathbf{D}_j) & \text{if } z_i \text{ has been constrained.} \end{cases}
\end{aligned} \tag{11}
$$

Consider Fig. 7, for example. Since the y component of point **P** was constrained by triangle j, while the x and z components were not, the iterated values of $x^{(m+1)}$ and $z^{(m+1)}$ was computed as usual, but the iterated value of $y^{(m+1)}$ was computed as:

$$
y^{(m+1)} = -\frac{1}{\mathbf{B}_j}(\mathbf{A}_j x^{(m+1)} + \mathbf{C}_j z^{(m)} + \mathbf{D}_j),
$$

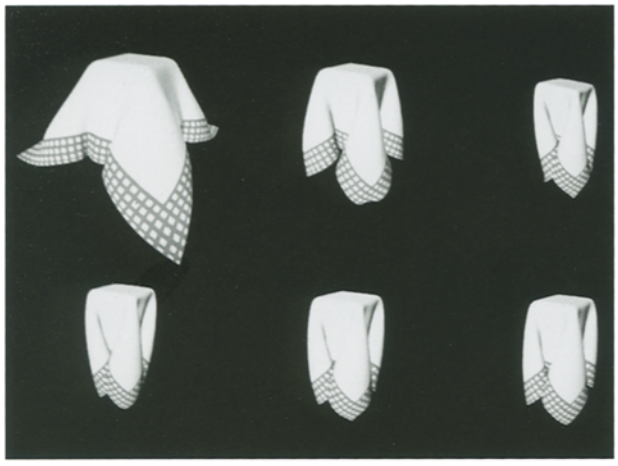
where $\mathbf{A}_j, \mathbf{B}_j, \mathbf{C}_j$, and $\mathbf{D}_j$ are precalculated and can be indexed with key j. After all the iteration steps, the position of point **P** is updated and constrained over the triangle without penetration (Fig. 9).

The method for preventing self-collisions between parts of the cloth employs the same scheme as that for preventing collisions between the cloth and rigid obstacles, but with modification of the collision detection test and constraint-setting technique. There are various situations in which two moving objects may collide and thus many different responses. For simplicity, we apply less sophisticated heuristic constraints that just fix the vertices of colliding triangles and sides as their respective positions at time t. After all the constraints have been set, we go back to time $t + \Delta t$ and simulate again by evaluating the linear system. We now rewrite the iteration steps of the Gauss-Seidel method as follows (these steps are somewhat different from those for the case of a collision between the cloth and a rigid obstacle):

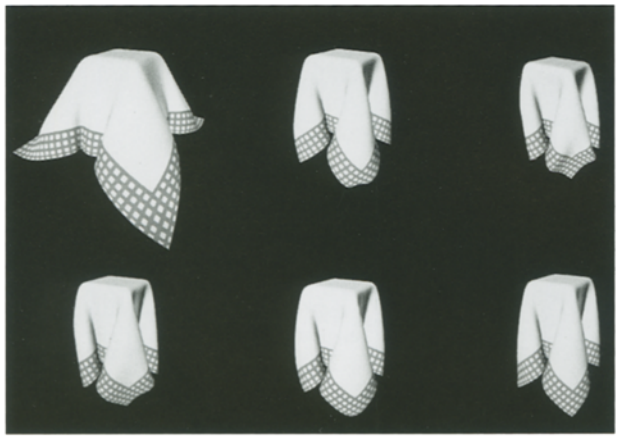
$$
\begin{aligned}
x_i^{(m+1)} &= \begin{cases} \frac{1}{a_{ii}}\{g_{xi} - \sum_{j=1}^{i-1} a_{ij}x_j^{(m+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(m)}\} & \text{if } x_i \text{ has not been constrained} \\ x_i^{(m)} & \text{if } x_i \text{ has been constrained} \end{cases} \\
y_i^{(m+1)} &= \begin{cases} \frac{1}{a_{ii}}\{g_{yi} - \sum_{j=1}^{i-1} a_{ij}y_j^{(m+1)} - \sum_{j=i+1}^{n} a_{ij}y_j^{(m)}\} & \text{if } y_i \text{ has not been constrained} \\ y_i^{(m+1)} & \text{if } y_i \text{ has been constrained} \end{cases} \\
z_i^{(m+1)} &= \begin{cases} \frac{1}{a_{ii}}\{g_{zi} - \sum_{j=1}^{i-1} a_{ij}z_j^{(m+1)} - \sum_{j=i+1}^{n} a_{ij}z_j^{(m)}\} & \text{if } z_i \text{ has not been constrained} \\ z_i^{(m)} & \text{if } z_i \text{ has been constrained} \end{cases}
\end{aligned} \tag{12}
$$

The method for preventing self-collisions between parts of the cloth employs the same scheme as that for preventing collisions between the cloth and rigid obstacles, but with modification of the collision detection test and constraint-setting technique. There are various situations in which two moving objects may collide and thus many different responses. For simplicity, we apply less sophisticated heuristic constraints that just fix the vertices of colliding triangles and sides as their respective positions at time t. After all the constraints have been set, we go back to time $t + \Delta t$ and simulate again by evaluating the linear system. We now rewrite the iteration steps of the Gauss-Seidel method as follows (these steps are somewhat different from those for the case of a collision between the cloth and a rigid obstacle):

After the Gauss-Seidel method iterations, the self-collision is successfully prevented at time $t + \Delta t$. Our collision avoidance method has several advantages. First, we need not apply any forces to avoid penetration. As a result, instead of exhibiting the strange cloth behavior discussed in Sect. 1, the cloth covers an object smoothly when



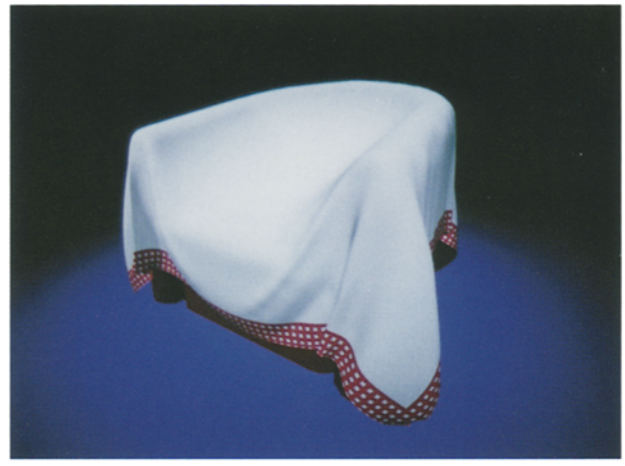
10



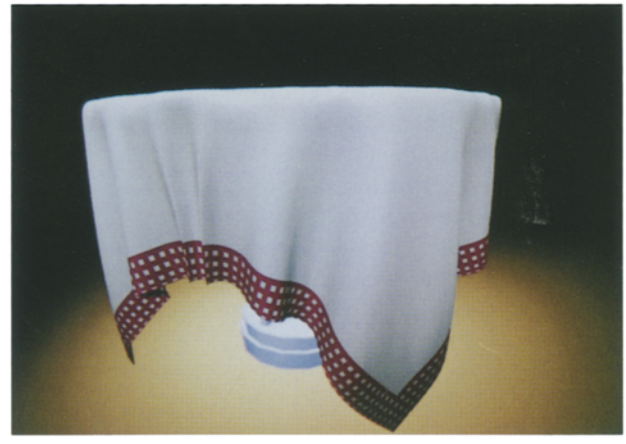
11

Fig. 10. Cloth falling over a table without self-collision detection

Fig. 11. Cloth falling over a table with self-collision detection



12



13

Fig. 12. 51×51 grid points of cloth falling over a sofa

Fig. 13. 51×51 grid points of cloth falling over a round table

collisions occur. Second, we do not need to worry about the shapes of stationary obstacles the cloth falls onto; most obstacles can easily be handled with our cloth-animation technique. Third, external forces, such as ruffling and lifting of the cloth by the wind, can still be used to create realistic images of cloth animation. Although producing such images may require us to recalculate the paths of many points to prevent collisions, the computing time needed depends mainly on the size of the cloth and not on the roll-back and recalculation processes.

Experimental results show that our collision and self-collision avoidance methods are easy to implement and produce realistic cloth images. In those methods that use extra response forces, it is hard work to tune the amounts of response force needed to produce realistic images, and often, inappropriate amounts of force cause divergence in the governing linear system. Since no response forces are applied to avoid penetration in our

Table 1. The experimental results for various detection strategies

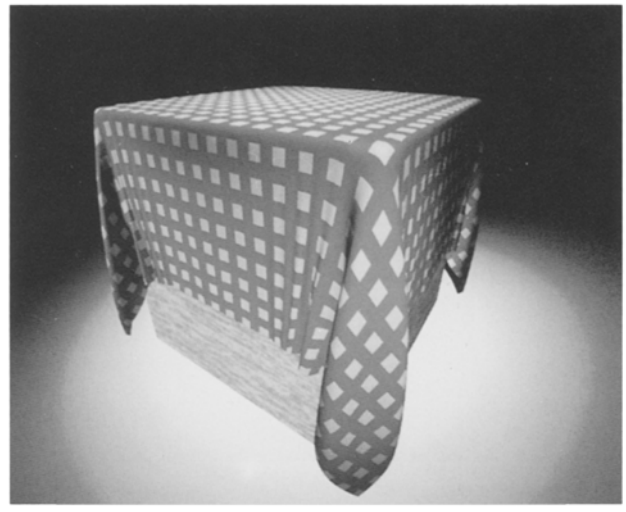
Cloth size	21 × 21	51 × 51	101 × 101
No collision detection	529 s	2519 s	11602 s
With collision detection	627 s	2756 s	11766 s
With self-collision detection	877 s	13823 s	81620 s

methods, fewer parameters need tuning. Thus the cloth behavior produced by our method is more stable and easier to control.

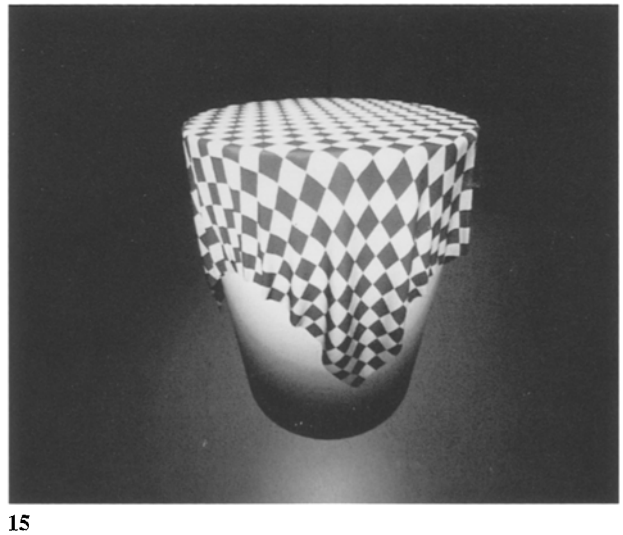
4 Experimental results

In our experiments, we simulated a cloth falling freely in the air and finally covering a table. We used three grid sizes for the cloth: 21×21, 51×51, and 101×101. The table was static and composed of ten triangles. We used an SGI Indy workstation with an R4000 CPU and 32 MB of memory for our experimental environment. Five hundred time steps were run in all simulations to achieve the final static appearance. The computation times for our cloth animation with three collision detection strategies are listed in Table 1. Note that, in this table, simulations with collision detection and self-collision detection required much more time than those with no self-collision detection, since too many triangles comprising the cloth were involved in the self-intersection checking needed for self-collision detection.

Figures 10–15 illustrate the images of our results. Figures 10 and 11 show the cloth falling over a table without and with self-collision detection. Figures 12 and 13 show the 51×51 grid-point cloth falling over a sofa and a round table. The sofa is composed of about 476 triangles, and the round table is composed of about 370 triangles. Figures 14 and 15 show the 101×101 grid points of cloth falling over a cube and a cylinder.



14



15

Fig. 14. 101×101 grid points of cloth falling over a cube

Fig. 15. 101×101 grid points of cloth falling over a cylinder

5 Concluding remarks

We have proposed a method for modeling deformable cloth. Our goal was to simulate efficiently and realistically the dynamic formation and the final static appearance of the draping and folding of cloth falling over a table. Our model was based on a physically based model proposed by Terzopoulos et al. (1987). On top of the physically based model, we proposed a new collision and self-collision avoidance method that does not use response forces. Using response forces causes the cloth to float over the table face. However, our method allows the cloth to cling to the table face instead, and it obtains a realistic final static appearance.

Further work remains to be done. It is desirable to enhance the proposed cloth model so it can be used to represent various kinds of materials, such as silk, flax, and so on. Although realistic cloth animation can be created, the overhead required by collision detection, especially self-collision detection, should be reduced to improve the efficiency of the simulation.

Acknowledgement. This research was supported by the National Science Council of the Republic of China under grant No. NSC 81-0408-E009-20.

References

Atkinson KE (1988) An introduction to numerical analysis. Wiley, New York

Baraff D (1989) Analytical methods for dynamic simulation of non-penetrating rigid bodies. Comput Graph 23:223–231

Boyse JW (1979) Interference detection among solids and surfaces. Commun ACM 22:3–9

Breen DE, House DH, Getto PH (1992) A physically based particle model of woven cloth. Visual Comput 8:264–277

Breen DE, House DH, Wozny MJ (1994) Predicting the drape of woven cloth using interacting particles. Comput Graph 24: 365–372

Carignan M, Yang Y, Thalmann NM, Thalmann D (1992) Dressing animated synthetic actors with complex deformable clothes. Comput Graph 26:99–104

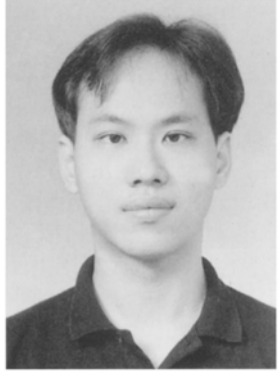
Lafleur B, Thalmann NM, Thalmann D (1991) Cloth animation with self-collision detection. Proceedings of the IFIP Conference on Modeling in Computer Graphics. Tokyo, Japan, Springer, Tokyo, pp 179–187

Moore M, Wilhelms J (1988) Collision detection and response for computer animation. Comput Graph 22:289–298

Terzopoulos D, Fleischer K (1988) Deformable models. Visual Comput 4:306–331

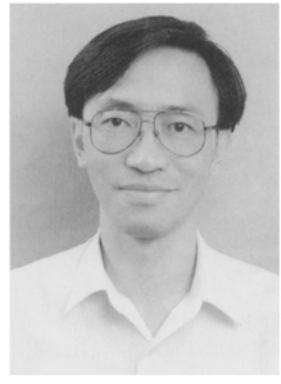
Terzopoulos D, Platt J, Fleischer K, Barr AH (1987) Elastically deformable models. Comput Graph 21:205–214

Weil J (1986) The synthesis of cloth objects. Comput Graph 20:49–54



JEN-DUO LIU was born on February 12, 1969, in Taichung, Taiwan, Republic of China. He received his BS degree in Applied Mathematics from National Chiao Tung University in 1991, and his MS degree in Computer Science from National Chiao Tung University in 1993. He is a PhD candidate in the Department of Computer and Information Science at the National Chiao Tung University. His current research interests are in the area of computer graphics and animation.

MING-TAT KO was born on September 1, 1957 in Taipei, Taiwan. He received his BS and MS degrees in Mathematics from National Taiwan University in 1979 and 1982, respectively. He received his PhD degree in Computer Science from National Tsing Hua University, Taiwan, in 1988. He then joined the Institute of Information Science, Academia Sinica, Taiwan, as an Asociate Research Fellow. His current research interests include the design and analysis of algorithms, computational geometry, graph algorithms, computer graphics, and real-time systems.



RUEI-CHUAN CHANG was born on January 30, 1958, in Keelung, Taiwan, Republic of China. He received his BS degree (1979), MS degree (1981), and PhD degree (1984), all in Computer Science from the National Chiao Tung University. Currently, he is a Professor in the Department of Computer and Information Science at the National Chiao Tung University in Hsinchu. He is also an Associate Research Fellow at the Institute of Information Science, Academia Sinica, Taipei. His current research interests include design and analysis of algorithms, computer graphics, and system software.