Pacific Graphics 2009
S. Lee, D. Lischinski, and Y. Yu
(Guest Editors)

Volume 28 (2009), Number 7

Simulation of Tearing Cloth with Frayed Edges

Napaporn Metaaphanon[†] Yosuke Bando[†,‡] Bing-Yu Chen[§] Tomoyuki Nishita[†]

[†]The University of Tokyo [‡]TOSHIBA Corporation [§]National Taiwan University

Abstract

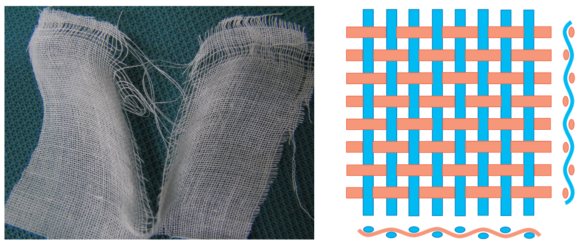
Woven cloth can commonly be seen in daily life and also in animation. Unless prevented in some way, woven cloth usually frays at the edges. However, in computer graphics, woven cloth is typically modeled as a continuum sheet, which is not suitable for representing frays. This paper proposes a model that allows yarn movement and slippage during cloth tearing. Drawing upon techniques from textile and mechanical engineering fields, we model cloth as woven yarn crossings where each yarn can be independently torn when the strain limit is reached. To make the model practical for graphics applications, we simulate only tearing part of cloth with a yarn-level model using a simple constrained mass-spring system for computational efficiency. We designed conditions for switching from a standard continuum sheet model to our yarn-level model, so that frays can be initiated and propagated along the torn lines. Results show that our method can achieve plausible tearing cloth animation with frayed edges.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; I.6.8 [Simulation and Modeling]: Types of Simulation—Animation

1. Introduction

In the field of computer graphics, researches about cloth are mainly focused on the deformation of cloth as a continuum. Typical models represent cloth as a sheet of elastic material, which means that only macroscopic geometric features of cloth could be modeled. Since these kinds of models allow compressing, stretching, out-of-plane and also in-plane bending, they are suitable for representing woven cloth in general cases. However, in some cases, treating cloth as a whole is not sufficient; the case of tearing cloth is one of them.

Torn edges of woven cloth usually fray as shown in Figure 1(a). The weave becomes ruined, and yarns can easily move out. This behavior is caused by weaving, which is a special characteristic of woven cloth. Much work in computer graphics assumes that slippage does not occur between yarns at their crossings including the edges. Hence, frayed edges of woven cloth are normally ignored, and only smooth torn edges are produced, resulting in rubber-like appearance of tear.



(a) Frays at torn edges (b) Plain weave

Figure 1: *(a) A real example of woven cloth and how it looks when torn. (b) Schematics of the plain weave.*

In this paper, we propose a cloth model that can capture the dynamic behavior of fraying at edges when woven cloth is being torn. In order to allow yarn movement and slippage at frayed edges, we adapt techniques from textile and mechanical engineering fields, and we model cloth as woven yarn crossings so that each yarn can be independently torn when the strain limit is reached. The primary contribution of this paper is that, as opposed to the typical yarn-level models in the textile industry with complicated formulation for pursuing mechanical accuracy, we use a simple constrained mass-spring system, and simulate only tearing part of cloth

† e-mail: {noimeta, ybando, nis}@nis-lab.is.s.u-tokyo.ac.jp
‡ e-mail: yosuke1.bando@toshiba.co.jp
§ e-mail: robin@ntu.edu.tw

with a yarn-level model in order to make the model computationally efficient and practical for computer graphics applications. We designed conditions for switching from a standard continuum sheet model to our yarn-level model, so that yarns around the torn area become able to slip over each other, which in turn induces slippage of other neighboring yarns as tearing proceeds. That is, fraying can be initiated and propagated through cloth in our model. We demonstrate that our method can produce visually pleasing tearing cloth animation by showing simulation results of cloth under various types of load which leads to different dynamic behaviors of fraying.

2. Related Work

Since cloth tearing simulation is related with both of cloth simulation and fracture simulation, they are introduced separately in this section.

2.1. Cloth Simulation

Traditionally, cloth models in computer graphics treat woven cloth as a sheet of material. Early work by Terzopoulos *et al.* [TPBF87, TF88a, TF88b] simulated deformable objects including cloth based on finite difference schemes. Since then, a number of researchers have addressed different aspects of cloth simulation, including computational efficiency [VCMT95, VMT00], numerical stability [BW98, CK02], collision handling [VMT00, BFA02, BWK03], cloth bending and buckling [GHDS03, BMF03, CK02, ZJW08], cloth inextensibility [Pro95, BFA02, MHHR07, GHF*07], and dressing synthetic actors [CYMTT92, VMTJT96]. While all of the above continuum approaches are efficient for representing most characteristics of cloth, they are not suitable for modeling frays or raveling edges of torn cloth. Although cloth tearing simulation using a continuum model is possible, only smooth torn edges can be generated.

In textile and mechanical engineering fields, however, yarn-level models are commonplace. The pioneering work of Peirce [Pei37] modeled geometric relationships among circular yarns at yarn crossings. A number of modified forms of Peirce's model have been developed: a non-circular cross section model by Kemp [Kem58], a three-dimensional yarn-crossing structure by Kawabata *et al.* [KNK73], and a curve model of yarn crossings by Warren [War90], for instance. These models are designed to investigate the elastic properties of fabrics at equilibrium, while we use a yarn-crossing model similar to these to simulate dynamic fraying behavior of tearing cloth.

Some researchers simulated dynamic cloth tearing in the context of analyzing ballistic properties of bulletproof woven fabric armors. The work of Zeng *et al.* [ZTS06] bears many similarities to ours, which models each yarn as a chain of linear elements whose nodal points are placed at yarn crossings. Frays can be seen in the simulated results around the bullet hole. Zhang *et al.* [ZBZ08] used a similar but more complicated model in which each yarn is discretized into hexahedral finite elements. The primary differences of our work are the simplicity and practicality of the proposed model for graphics applications: we simulate only tearing part of cloth with a yarn-level model using a simple constrained mass-spring system for computational efficiency, with tailored conditions for switching from a standard continuum sheet model to the yarn-level model.

Some work from the computer graphics community conducted yarn- or strand-level simulation. Breen *et al.* [BHW94] used a particle-based model that represents yarn crossings to simulate static shapes of draped woven cloth. Kaldor *et al.* [KJM08] presented yarn-level simulation of knitted cloth that produced detailed and visually intriguing deformation of knitwear. Unfortunately, as mentioned in their paper, the mechanical structures and behaviors of knits are totally different from woven cloth, and therefore we need to devise a separate model for woven cloth. Selle *et al.* [SLF08] proposed a hair model that can efficiently and stably simulate every one of a hundred thousand strands on a human head. None of the above methods treat tearing and fraying of yarns.

2.2. Fracture Simulation

In computer graphics literature, fracture simulation has been researched on several kinds of materials. Terzopoulos and Fleischer [TF88a, TF88b] demonstrated a technique to tear sheets of paper and cloth-like material. O'Brien and Hodgins [OH99] simulated fracture of brittle objects based on finite element methods, which was extended to include ductile fracture [OBH02]. Smith *et al.* [SWB01] also simulated shattering of brittle objects efficiently by constraint forces. Norton *et al.* [NTB*91] used a mass-spring system to model solid objects that broke under large strains, whereas Hirota *et al.* [HTK98] modeled the drying process of mud with cracks. Unfortunately, none of the above methods modeled strand-like material, and they are not directly applicable to simulating dynamic fraying behavior of tearing cloth.

3. Woven Cloth Modeling

As described in Section 1, a two-level model is used for simulating tearing cloth: the *base cloth model*, which is a continuum sheet model for simulating untorn part of cloth; and the *yarn-level model* for simulating raveling edges. In this section, we start by reviewing the basic structure of woven cloth, and then explain about each level of our cloth model and how and when to switch from the base cloth model to the yarn-level model.

3.1. Structure of Woven Cloth

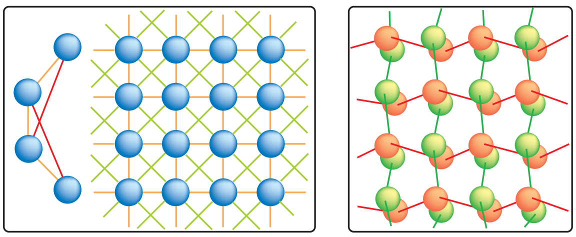
Woven cloth is made up of two sets of yarns called *warps* and *wefts*. Warps are lengthwise yarns that run vertically in woven fabrics, while wefts are crosswise yarns that run horizontally under and over parallel warps. Various mechanical properties of cloth are determined by the characteristics of yarns themselves and how they are woven or interlaced [HB00]. The simplest fundamental weave structure is *plain weave*, in which each weft simply goes over one warp and under the next repetitively as shown in Figure 1(b).

For simplicity of explanation, we will assume plain weave in what follows; applying our method to other types of weave is straightforward, though. We will show an example of twill weave in Section 6.

3.2. Base Cloth Model

As continuum sheet models often suffice for simulating undamaged cloth, we model untorn part of cloth using a standard mass-spring system as shown in Figure 2(a). Three types of springs are attached to each mass: *structural springs*, *bend springs* and *shear springs*. Structural springs connect immediate neighbors, while bend springs connect every other particles. Shear springs connect diagonal particles, and are only active under compression [ZJW08]. Table 1 summarizes the spring constants we used, along with other parameters which will be used in this paper. For numerical stability, the magnitude of the structural spring constant k_{struct} is kept moderate, and the tensile stiffness of cloth is enforced by strain limiting, as will be described in Section 4.1. For consistency with the yarn-level model, we use quadrilateral meshes of yarn-level resolution. Using general triangular meshes and coarser resolution for the base cloth model is left as future work.

For tearing simulation, we assign strain threshold θ_{tear} randomly between 5% and 10% to each structural spring, and we cut springs whose strain exceeds their thresholds.



(a) Base cloth model (b) Yarn-level model

Figure 2: *The two models used in our method. (a) The base cloth model with mass particles (blue), structural springs (orange), shear springs (green) and bend springs (red). (b) The yarn-level model with mass couples. The green balls and springs represent warps, while the red balls and springs represent wefts.*

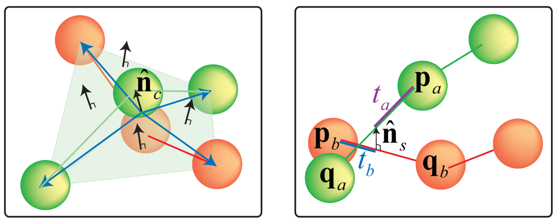
Table 1: *Parameters used in our model.*

Cloth thickness T	0.01 cm
Cloth areal density ρ	0.25 g/cm^2
Number of yarns per unit length σ	10-15 cm^{-1}
Structural spring constant k_{struct}	6000 g/s^2
Bend spring constant k_{bend}	5.0 g/s^2
Shear spring constant k_{shear}	2.0 g/s^2
Strain threshold for tearing θ_{tear}	5-10%
Strain threshold for model transition θ_{trans}	4.5-9%
Distance threshold for mass couples θ_{dist}	0.02 cm
Number of constraint projections N_{proj}	30-50
Time step Δt	1/240 s

3.3. Yarn-Level Model

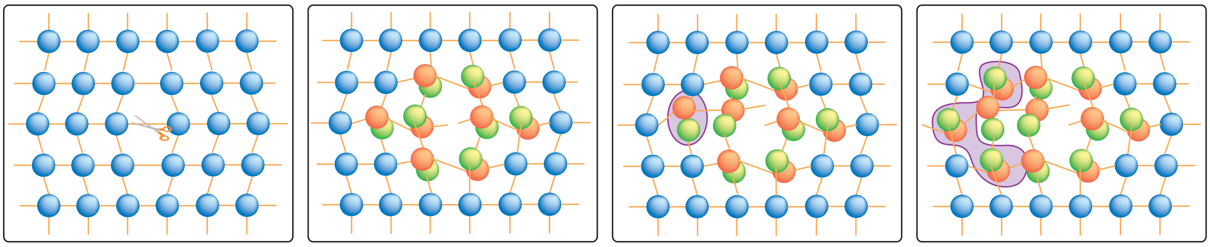
At the yarn level, individual yarns are modeled explicitly as warps or wefts crossing over the others (see Figure 2(b)). Each yarn is composed of a series of mass particles and springs. At the rest state, one particle from a warp and another particle from a weft are located at their crossover point (we collectively call them *a mass couple*). Initially, the two particles in a mass couple are separated by the cloth thickness T in the direction of the cloth normal vector $\hat{\mathbf{n}}_c$ at the crossing. The normal vector $\hat{\mathbf{n}}_c$ is calculated as the average of the normal vectors of the four triangles around the crossover point as shown in Figure 3(a). In our algorithm, the normal vector $\hat{\mathbf{n}}_c$ is calculated only at most once per crossover point when a particle in the base cloth model is split into a mass couple in the yarn-level model.

As we use the yarn-level model to represent frays, we do not strictly constrain a mass couple to stay close to each other, and we allow two states for each mass couple, referred to as *loosely connected* and *disconnected*. In the loosely connected state, we apply constraints so that the warp and weft springs attached to a mass couple can slide over each other, which will be explained below in detail. When a mass couple draws apart beyond the distance threshold θ_{dist}, the state of that mass couple is changed to disconnected. In the disconnected state, no constraints are applied.



(a) Cloth normal (b) Spring pair's normal

Figure 3: *Two types of normal vectors used in our method. (a) The normal vector $\hat{\mathbf{n}}_c$ of cloth. (b) The normal vector $\hat{\mathbf{n}}_s$ for the closest pair of a warp spring and a weft spring.*



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

Figure 4: *Transition from the base cloth model to the yarn-level model. (a) The base cloth model is stretched and one of the structural springs is cut. (b) The two particles at both ends of the cut spring, plus their adjacent particles, are split. (c) One of the mass couples becomes disconnected as highlighted in purple. (d) The neighboring particles are also split.*

Constraint applying process for a loosely connected mass couple starts from determining the closest pair of a warp spring and a weft spring attached to the mass couple as shown in Figure 3(b). Since there are at most two warp and two weft springs per mass couple, a maximum of four warp-weft spring pairs need to be checked so as to identify the closest one. Afterwards, the normal vector $\hat{\mathbf{n}}_s$ is obtained by normalizing the vector between the closest points. The constraints are applied in the direction of normal vector $\hat{\mathbf{n}}_s$ to one spring and in the opposite direction $-\hat{\mathbf{n}}_s$ to the other, so that the closest distance will be kept constant (i.e., cloth thickness T). However, instead of applying constraints directly to the closest points, they are applied to the spring endpoints, weighted by the barycentric coordinates [BFA02]. Let a and b be the closest warp-weft spring pair, and $\mathbf{p}_a$, $\mathbf{q}_a$, $\mathbf{p}_b$, and $\mathbf{q}_b$ be their endpoints. If the closest points are at the fraction t_a and t_b along the edges $\overline{\mathbf{p}_a\mathbf{q}_a}$ and $\overline{\mathbf{p}_b\mathbf{q}_b}$ as shown in Figure 3(b), the new positions of the endpoints can be calculated as:

$$\begin{aligned}
\mathbf{p}_a^{new} &= \mathbf{p}_a + (1-t_a)(d-T)\hat{\mathbf{n}}_s, \\
\mathbf{q}_a^{new} &= \mathbf{q}_a + (t_a)(d-T)\hat{\mathbf{n}}_s, \\
\mathbf{p}_b^{new} &= \mathbf{p}_b - (1-t_b)(d-T)\hat{\mathbf{n}}_s, \\
\mathbf{q}_b^{new} &= \mathbf{q}_b - (t_b)(d-T)\hat{\mathbf{n}}_s,
\end{aligned} \tag{1}$$

where d is the closest distance between the two springs.

3.4. Model Transition

The transition from the base cloth model to the yarn-level model occurs in over-strained regions, where weave becomes loose. The mass particles at both ends of a structural spring whose strain exceeds the threshold θ_{trans}, as well as their immediate neighbor particles, are split into mass couples as shown in Figures 4(a)(b). Although Figures 4(a)(b) show for simplicity the case where the model transition occurs at the same time when a spring is cut, we actually set the threshold θ_{trans} slightly less than the spring cutting threshold θ_{tear} as listed in Table 1, so that weave loosening can occur before cloth begins to tear.

When a mass couple becomes disconnected as described in Section 3.3, the mass splitting propagates to its neighbors as shown in Figures 4(c)(d). The propagation stops at particles with a different weaving pattern. In other words, if the warp is over the weft at a disconnected mass couple, we split masses until reaching points where a weft is over a warp.

Let m_{base}, $\mathbf{x}_{base}$, and $\mathbf{v}_{base}$ denote the mass, position, and velocity of a mass particle in the base model. We also use subscripts *warp* and *weft* to indicate those of a mass couple in the yarn-level model. As described in Section 3.3, we split a mass particle in the cloth normal direction $\hat{\mathbf{n}}_c$ as:

$$\begin{aligned}
\mathbf{x}_{warp} &= \mathbf{x}_{base} + \frac{T}{2}\hat{\mathbf{n}}_c, \\
\mathbf{x}_{weft} &= \mathbf{x}_{base} - \frac{T}{2}\hat{\mathbf{n}}_c.
\end{aligned} \tag{2}$$

Note that the signs (addition or subtraction) in Eq. (2) may be opposite depending on the weave pattern of a fabric. We set $m_{warp} = m_{weft} = m_{base}/2$ for conservation of mass, and $\mathbf{v}_{warp} = \mathbf{v}_{weft} = \mathbf{v}_{base}$ for conservation of linear and angular momenta, as shown in Figure 5(a). In fact, letting $\mathbf{P}$ and $\mathbf{L}$ denote linear and angular momenta, we have:

$$\begin{aligned}
\mathbf{P}_{warp} + \mathbf{P}_{weft} &= m_{warp}\mathbf{v}_{warp} + m_{weft}\mathbf{v}_{weft} \\
&= \frac{m_{base}}{2}\mathbf{v}_{base} + \frac{m_{base}}{2}\mathbf{v}_{base} \\
&= m_{base}\mathbf{v}_{base} = \mathbf{P}_{base},
\end{aligned} \tag{3}$$

and also:

$$\begin{aligned}
\mathbf{L}_{warp} + \mathbf{L}_{weft} &= \mathbf{x}_{warp} \times m_{warp}\mathbf{v}_{warp} + \mathbf{x}_{weft} \times m_{weft}\mathbf{v}_{weft} \\
&= (\mathbf{x}_{base} + \frac{T}{2}\hat{\mathbf{n}}_c) \times \frac{m_{base}}{2}\mathbf{v}_{base} \\
&\quad + (\mathbf{x}_{base} - \frac{T}{2}\hat{\mathbf{n}}_c) \times \frac{m_{base}}{2}\mathbf{v}_{base} \\
&= \mathbf{x}_{base} \times m_{base}\mathbf{v}_{base} = \mathbf{L}_{base}.
\end{aligned} \tag{4}$$

When splitting a mass particle, we reattach the springs to a mass couple as follows. As shown in Figure 5(a), structural springs are separated into warp and weft springs, and attached to the corresponding particles. Reattachment of bend springs goes in a similar manner to that of structural springs.

If a bend spring is on warp, we attach it to the warp particle; otherwise we attach it to the weft particle as shown in Figure 5(b). Shear springs need duplication before reattachment. Two duplicated shear springs connect particles in the same plane (upper or lower one), or connect to the same unsplit particle as shown in Figure 5(c). Because the number of springs is doubled, the shear spring constant becomes a half of the original one (i.e., k_{shear}) to preserve the same stiffness.

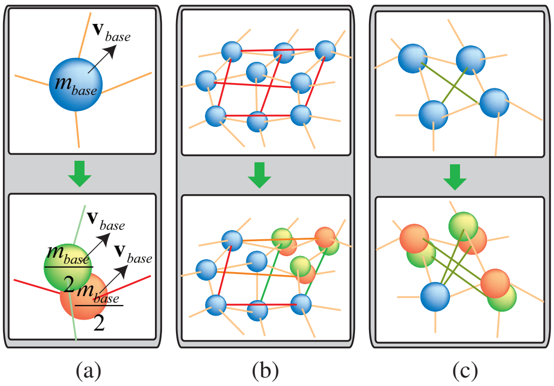


Figure 5: *Splitting mass particles (from top to bottom). (a) Structural spring reattachment. (b) Bend spring reattachment. (c) Shear spring duplication and reattachment.*

4. Cloth Simulation

In each simulation loop, our algorithm performs as follows:

1. Apply forces to cloth particles.
2. Update velocities.
3. Damp velocities.
4. Update positions.
5. Apply constraints (Eq. (1) and Section 4.1).
6. Process collisions (Section 4.2).
7. Apply model transition and cut springs according to the strain values of the springs (Section 3.4).
8. Post-manipulate velocities.

For time integration of cloth dynamics, we use 8 explicit Euler steps with time step Δt = 1/240 s to produce 30 fps animation. Stability is achieved by the constraint projection method for strain limiting and the post velocity manipulation of Müller *et al.* [MHHR07] (Steps 3, 5, and 8).

4.1. Strain Limiting

We impose constraints on structural springs to prevent them from stretching excessively. Each constraint is solved independently one after the other according to [MHHR07]. Since correcting the length of one spring can cause the length of other springs to change, multiple "sweeps" of the constraint projection are necessary, and results can depend on the order in which springs are processed. We first perform test-run of the simulation without tearing forces, and determine the number of sweeps N_{proj} needed to limit the extension of all springs in the system below the threshold θ_{tear}. We use this number N_{proj} for simulating tearing cloth, and we cut overstrained springs beyond θ_{tear} even after N_{proj} sweeps of the constraint projection are performed. Similar to the idea of strain limiting ordering presented by Selle *et al.* for hair simulation [SLF08], we apply strain limiting in increasing order of distance to fixed mass particles (e.g., those attached to the balls and bars in Figures 6 and 7), as springs around fixed particles are most likely to be elongated.

4.2. Collision Handling

Since this paper is mainly concerned with cloth tearing and fraying, we assume objects other than cloth consist of rigid bodies, and we only handle collisions between cloth particles and rigid body faces. Although this allows penetration of springs into rigid bodies, no visual artifacts were seen in our experiments.

The self-collision type in our model is edge-edge collision. To detect a collision between two moving edges $\overline{\mathbf{x}_1\mathbf{x}_2}$ and $\overline{\mathbf{x}_3\mathbf{x}_4}$, we check if there is time t between two consecutive frames (at t = 0 and 1) at which the both edges are touching. When two edges are in contact, the four endpoints are coplanar. Therefore, assuming the endpoints move with constant velocities $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ and $\mathbf{v}_4$ between the frames, the collision time t can be obtained by solving the following equation:

$$(\mathbf{x}_{31} + t\mathbf{v}_{31}) \cdot (\mathbf{x}_{21} + t\mathbf{v}_{21}) \times (\mathbf{x}_{43} + t\mathbf{v}_{43}) = 0, \qquad (5)$$

where $\mathbf{x}_{ij}$ and $\mathbf{v}_{ij}$ are short for $\mathbf{x}_j - \mathbf{x}_i$ and $\mathbf{v}_j - \mathbf{v}_i$. To compute the roots t, instead of floating-point arithmetic, we use interval arithmetic for robust calculation [Eri04]. All roots outside of [0, 1] are discarded. If multiple roots are left, the least one is chosen as the colliding time. If there is no root, no collision has occurred.

5. Yarn Rendering

To capture the details of weaving and to prevent possible visual seams at the border between the two-level models, the whole cloth is rendered as a number of interleaved yarns. Warp and weft positions for base-model masses are calculated using Eq. (2). Each yarn is drawn as a Catmull-Rom spline that interpolates warp/weft particles. Although interpolated splines are not guaranteed to be free from mutual penetration, no noticeable artifacts were seen in our rendering results.

For yarn shading, we use the method presented in [HDK*06] that blends between Kajiya-Kay [KK89] and Lambertian shading. This shader requires the tangent vector of yarns, which we can calculate as the derivative of the spline equation. We applied alpha blending to the tips of torn yarns in order to create the stretched appearance.

6. Results

Figures 6-11 show our results. Simulation parameters are listed in Table 1, and some performance statistics are summarized in Table 2. All simulations were performed on a desktop PC with an Intel Core 2 Duo 2GHz CPU.

In Figure 6, a piece of cloth is grabbed by two balls at the corners and torn apart. Frays can be seen along the torn line. Another example is shown as an animation sequence in Figure 9. Figure 7 shows the case where a piece of cloth which is firmly attached to two bars on the left and right sides is torn by a thrown ball. Variation of weaving pattern is applied to the same scene, and the cases of plain weave and twill weave are shown. Figure 8 shows cloth that is pulled and torn by a cube attached to the bottom right corner. Some more examples are shown in Figure 11, along with the corresponding wire-frame models illustrating how the base model and the yarn-level model are combined.

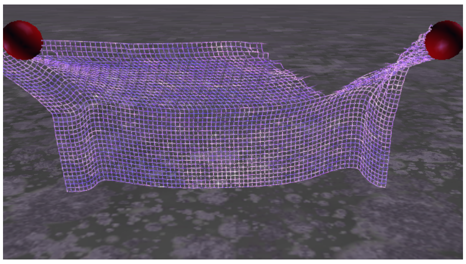


Figure 6: *A piece of cloth grabbed by two balls at its corners is torn apart.*

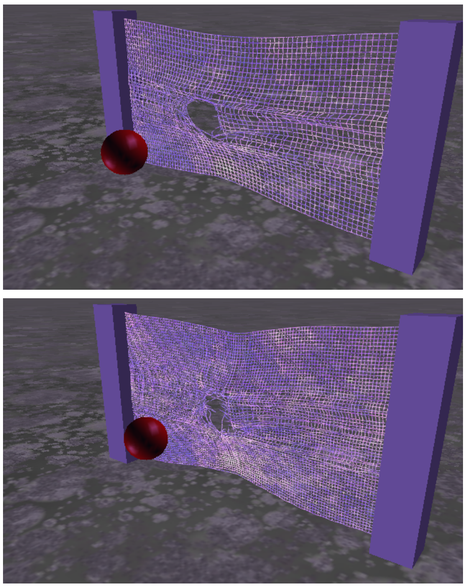


Figure 7: *Two types of woven cloth are torn apart by a thrown ball. Top: plain weave. Bottom: twill weave.*

Table 2: *Performance statistics for some of our results. The averaged times per frame for the whole animation sequence are shown. The yarn counts are given as "warps × wefts".*

	Number of yarns	Simulation time
Fig. 6	80 × 40	10.3 sec/frame
Fig. 8	160 × 80	67.7 sec/frame
Fig. 9	64 × 64	13.1 sec/frame

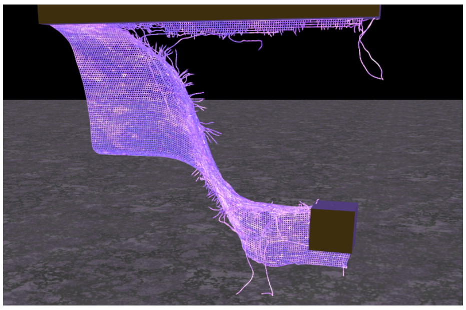


Figure 8: *A piece of cloth is pulled and torn by a cube.*

Figure 10(a) shows a plot of the average simulation times per frame against the number of yarn crossings used to simulate tearing cloth as in Figure 6. An approximately linear relationship is observed, suggesting that our method is scalable. Figure 10(b) shows the time development of the performance of our method when 120 × 60 yarns are used, along with the performance obtained by simulating the whole cloth solely with the yarn-level model. As our yarn-level model has only the loosely connected and disconnected states for mass couples, we added a *connected* state to this "all-yarn" model, in which we constrain each mass couple to stay close in order to keep cloth from raveling easily. As shown in the plots, we observed over 2.5 times performance gain in the beginning of the simulation because the number of particles is halved, and constraints on mass couples are not necessary. While the gain gradually decreases as the base model partially switches to the yarn-level model, our two-level model still runs faster thanks to the remaining base model part.

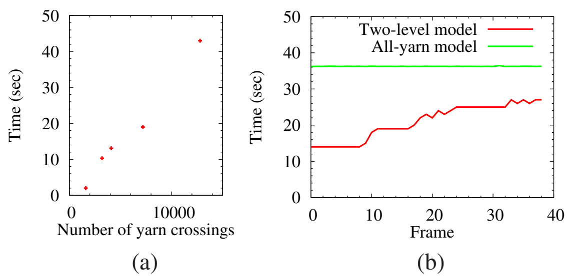


Figure 10: *(a) Plot of averaged simulation times against the number of yarn crossings. (b) Plots of the time development of simulation times.*

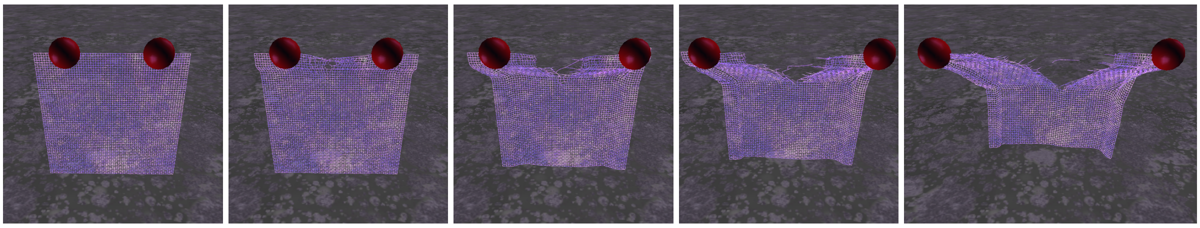
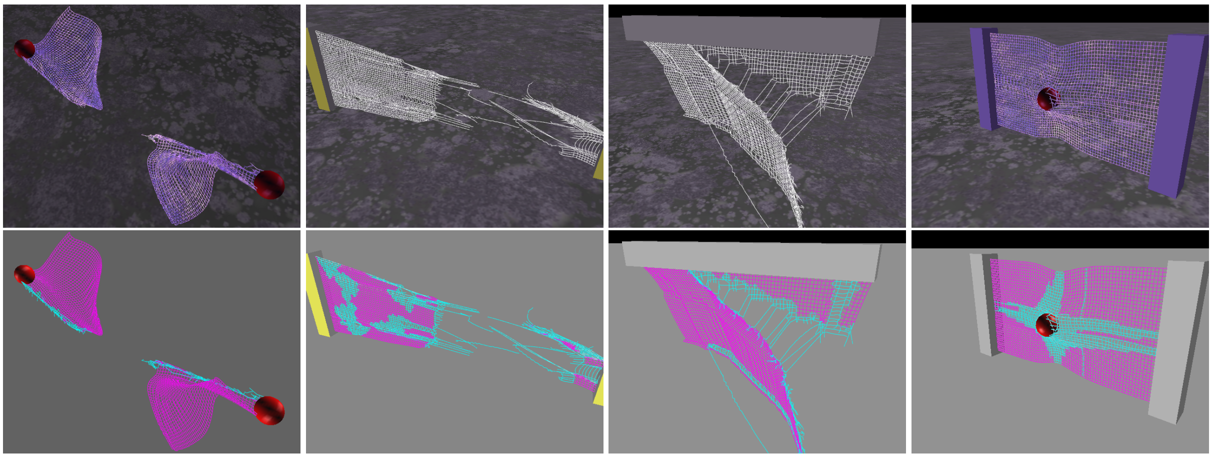


Figure 9: *Frames from an animation of tearing cloth grabbed by two balls (from left to right).*



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

Figure 11: *Rendering results (top) and the corresponding wire-frame models (bottom). The base model area is shown in magenta, while the yarn-level model area is shown in cyan. (a) Tearing cloth in the direction perpendicular to the cloth plane. (b) Cloth is torn away by being attached to two rigid bars that are moving apart. (c) Cloth is torn open under load at the bottom right corner. (d) Cloth is torn by a thrown ball.*

7. Conclusions and Future Work

We have presented a method for simulating tearing cloth with frayed edges. Our model does not treat cloth as a whole sheet of material, but as individual warps and wefts that can cut independently. As a result, frays, the most commonly seen characteristic of torn woven cloth, are visible. Our primary contribution lies in the combination of two kinds of models: the base cloth model which represents the whole cloth at the initial state, and the yarn-level model which represents eventually ruined part. Compared to modeling all yarns directly, the number of nodes at the initial state is halved, and then the model is gradually refined only around torn part. As a consequence, the resources needed for simulation can be reduced. We have presented a set of model refinement criteria with which fraying can be initiated and propagated through cloth.

While we have demonstrated the benefit of our two-level model, the run times of our prototype implementation are still rather long as compared to state-of-the-art cloth simulation methods for graphics applications. We attribute this mainly to a large number of constraint projections and also to our unoptimized code. We would like to further analyze this issue and to speed up the simulation.

One of the limitations of our method is that the whole cloth has to be modeled at the yarn-level resolution. A large number of masses and springs would be required for dense cloth representation, which can lead to high computational cost and slow performance. To resolve this limitation, we plan to develop a hierarchical refinement method that can use a coarser mesh for undamaged part of cloth. We would also like to adapt our method to more commonly-used, general triangular meshes instead of regular quadrilateral meshes.

References

[BFA02] Bridson R., Fedkiw R., Anderson J.: Robust treatment of collisions, contact and friction for cloth animation. *ACM Transactions on Graphics 21*, 3 (2002), 594–603.

[BHW94] Breen D. E., House D. H., Wozny M. J.: Predicting the drape of woven cloth using interacting particles. In *ACM SIGGRAPH 1994 Conference Proceedings* (1994), pp. 365–372.

[BMF03] Bridson R., Marino S., Fedkiw R.: Simulation of clothing with folds and wrinkles. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), pp. 28–36.

[BW98] BARAFF D., WITKIN A.: Large steps in cloth simulation. In *ACM SIGGRAPH 1998 Conference Proceedings* (1998), pp. 43–54.

[BWK03] BARAFF D., WITKIN A., KASS M.: Untangling cloth. *ACM Transactions on Graphics 22*, 3 (2003), 862–870.

[CK02] CHOI K.-J., KO H.-S.: Stable but responsive cloth. *ACM Transactions on Graphics 21*, 3 (2002), 604–611.

[CYMTT92] CARIGNAN M., YANG Y., MAGNENAT-THALMANN N., THALMANN D.: Dressing animated synthetic actors with complex deformable clothes. *ACM SIGGRAPH Computer Graphics 26*, 2 (1992), 99–104.

[Eri04] ERICSON C.: *Real-Time Collision Detection*. Morgan Kaufmann, 2004.

[GHDS03] GRINSPUN E., HIRANI A. N., DESBRUN M., SCHRÖDER P.: Discrete shells. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), pp. 62–67.

[GHF*07] GOLDENTHAL R., HARMON D., FATTAL R., BERCOVIER M., GRINSPUN E.: Efficient simulation of inextensible cloth. *ACM Transactions on Graphics 26*, 3 (2007), 49:1–49:7.

[HB00] HOUSE D. H., BREEN D. E. (Eds.): *Cloth modeling and animation*. A. K. Peters, Ltd., 2000.

[HDK*06] HIEBERT B., DAVE J., KIM T.-Y., NEULANDER I., RIJPKEMA H., TELFORD W.: The chronicles of Narnia: the lion, the crowds and rhythm and hues. *ACM SIGGRAPH 2006 Course Notes*, Course 11, 2006.

[HTK98] HIROTA K., TANOUE Y., KANEKO T.: Generation of crack patterns with a physical model. *The Visual Computer 14*, 3 (1998), 126–137.

[Kem58] KEMP A.: An extension of Peirce's cloth geometry to the treatment on non-circular threads. *Journal of the Textile Institute 49*, 1 (1958), 44–48.

[KJM08] KALDOR J., JAMES D. L., MARSCHNER S.: Simulating knitted cloth at the yarn level. *ACM Transactions on Graphics 27*, 3 (2008), 65:1–65:9.

[KK89] KAJIYA J. T., KAY T. L.: Rendering fur with three dimensional textures. *ACM SIGGRAPH Computer Graphics 23*, 3 (1989), 271–280.

[KNK73] KAWABATA S., NIWA M., KAWAI H.: The finite-deformation theory of plain-weave fabrics part I: the biaxial-deformation theory. *Journal of the Textile Institute 64*, 1 (1973), 21–46.

[MHHR07] MÜLLER M., HEIDELBERGER B., HENNIX M., RATCLIFF J.: Position based dynamics. *Journal of Visual Communication and Image Representation 18*, 2 (2007), 109–118.

[NTB*91] NORTON A., TURK G., BACON R., GERTH J., SWEENEY P.: Animation of fracture by physical modeling. *The Visual Computer 7*, 4 (1991), 210–219.

[OBH02] O'BRIEN J. F., BARGTEIL A. W., HODGINS J. K.: Graphical modeling and animation of ductile fracture. *ACM Transactions on Graphics 21*, 3 (2002), 291–294.

[OH99] O'BRIEN J. F., HODGINS J. K.: Graphical modeling and animation of brittle fracture. In *ACM SIGGRAPH 1999 Conference Proceedings* (1999), pp. 137–146.

[Pei37] PEIRCE F. T.: The geometry of cloth structure. *Journal of the Textile Institute 28*, 3 (1937), 45–96.

[Pro95] PROVOT X.: Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *Proceedings of the 1995 Conference on Graphics Interface* (1995), pp. 147–154.

[SLF08] SELLE A., LENTINE M., FEDKIW R.: A mass spring model for hair simulation. *ACM Transactions on Graphics 27*, 3 (2008), 64:1–64:11.

[SWB01] SMITH J., WITKIN A., BARAFF D.: Fast and controllable simulation of the shattering of brittle objects. *Computer Graphics Forum 20*, 2 (2001), 81–91.

[TF88a] TERZOPOULOS D., FLEISCHER K.: Modeling inelastic deformation: viscoelasticity, plasticity, fracture. *ACM SIGGRAPH Computer Graphics 22*, 3 (1988), 269–278.

[TF88b] TERZOPOULOS D., FLEISCHER K. W.: Deformable models. *The Visual Computer 4*, 6 (1988), 306–331.

[TPBF87] TERZOPOULOS D., PLATT J., BARR A., FLEISCHER K.: Elastically deformable models. *ACM SIGGRAPH Computer Graphics 21*, 3 (1987), 205–214.

[VCMT95] VOLINO P., COURCHESNE M., MAGNENAT-THALMANN N.: Versatile and efficient techniques for simulating cloth and other deformable objects. In *ACM SIGGRAPH 1995 Conference Proceedings* (1995), pp. 137–144.

[VMT00] VOLINO P., MAGNENAT-THALMANN N.: Implementing fast cloth simulation with collision response. In *Computer Graphics International 2000 Conference Proceedings* (2000), pp. 257–266.

[VMTJT96] VOLINO P., MAGNENAT-THALMANN N., JIANHUA S., THALMANN D.: An evolving system for simulating clothes on virtual actors. *IEEE Computer Graphics & Applications 16*, 5 (1996), 42–51.

[War90] WARREN W. E.: The elastic properties of woven polymeric fabric. *Polymer Engineering and Science 30*, 20 (1990), 1309–1313.

[ZBZ08] ZHANG G. M., BATRA R. C., ZHENG J.: Effect of frame size, frame type, and clamping pressure on the ballistic performance of soft body armor. *Composites 39*, 3 (2008), 476–489.

[ZJW08] ZHOU C., JIN X., WANG C. C. L.: Shear buckling and dynamic bending in cloth simulation. *Computer Animation and Virtual Worlds 19*, 3–4 (2008), 493–503.

[ZTS06] ZENG X. S., TAN V. B. C., SHIM V. P. W.: Modelling inter-yarn friction in woven fabric armour. *International Journal for Numerical Methods in Engineering 66*, 8 (2006), 1309–1330.