Adaptive Fracture Simulation of Multi-Layered Thin Plates

Oleksiy Busaryev* The Ohio State University

Tamal K. Dey* The Ohio State University

Huamin Wang* The Ohio State University

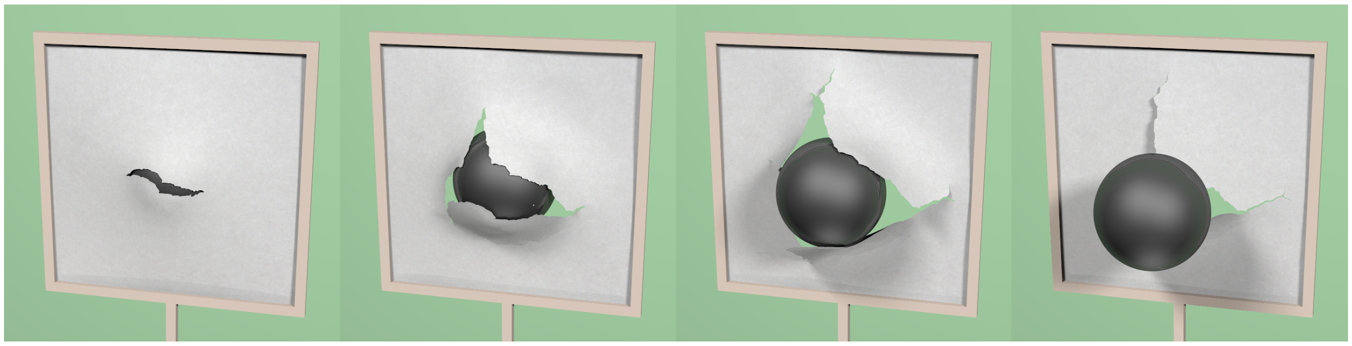

Figure 1: *A metal ball tearing a paper sheet. We use FEM to simulate paper and a 2D constrained Delaunay triangulation to adaptively refine the mesh near the potential crack boundary. This allows us to provide more fracture details with an affordable computational cost.*

Abstract

The fractures of thin plates often exhibit complex physical behaviors in the real world. In particular, fractures caused by tearing are different from fractures caused by in-plane motions. In this paper, we study how to make thin-plate fracture animations more realistic from three perspectives. We propose a stress relaxation method, which is applied to avoid shattering artifacts after generating each fracture cut. We formulate a fracture-aware remeshing scheme based on constrained Delaunay triangulation, to adaptively provide more fracture details. Finally, we use our multi-layered model to simulate complex fracture behaviors across thin layers. Our experiment shows that the system can efficiently and realistically simulate the fractures of multi-layered thin plates.

Keywords: Adaptive remeshing, FEM, thin plates, fracture simulation, layers.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics—Animation.

Links: DL PDF

1 Introduction

The separation of an object into multiple pieces, known as *fracture*, is a common phenomenon in the real world and an important topic in computer animation research. Fracture happens when the stress (or the strain) of an object grows beyond its material strength. To animate 3D fractures, O'Brien and his collaborators [1999; 2002] described the stress on each vertex using a 3D separation tensor, and compared its largest eigenvalue with a fracture threshold. Since a thin-plate object has a small thickness and little deformation in its thickness direction, a straightforward way to animate its fracture is to consider the planar stress only. While this is a reasonable assumption for stiff and homogenous thin plates as Gingold and colleagues [2004] showed, compliant thin plates in the real world often have more complex fracture behaviors, in which the separation effect in the thickness direction cannot be ignored. One such example is shown in Figure 2. When a piece of paper is stretched in its 2D plane, no separation exists in the thickness direction as Figure 2a shows. However, when it is torn, one layer gets peeled from another as Figure 2b shows. This is largely due to the fact that many thin plates are made of multiple layers. The tearing motion produces a large tensile stress in the thickness direction, causing separations in both the 2D plane and the thickness direction. Examples of such compliant thin-plate materials include paper, leather, leaf, plywood, and composite cloth.

An immediate approach to simulate multi-layered thin plates is to define the layers separately and model their interactions using short, adhesive inter-layer springs. However, this can be computationally expensive, not only because of more triangles used to model more layers, but also due to intensive collision handling among the layers. Ideally, layers can share the same triangulation until their separation happens. Since collision tests are usually the bottleneck in collision handling, how to minimize them in fracture simulation is another interesting problem that we would like to study.

The fractures of real-world thin plates often exhibit fine details on their cuts. Most existing techniques form a fracture cut by splitting a triangle into two triangles. As a result, the resolution of the fracture details depends on the resolution of the initial mesh. To generate a highly detailed fracture animation without knowing the fracture path ahead of time, the whole initial mesh needs to be in high resolution. This is a waste of both memory and computational time, since most triangles will not be involved in fracture.

We propose an adaptive approach to efficiently and realistically simulate the fractures of multi-layered thin plates. In this approach, we made the following contributions.

- We propose a stress relaxation method to handle multiple fracture cuts in a single time step. By calculating local stress

*e-mail: {busaryev, tamaldey, whmin}@cse.ohio-state.edu

ACM Reference Format
Busaryev, O., Dey, T., Wang, H. 2013. Adaptive Fracture Simulation of Multi-Layered Thin Plates. ACM Trans. Graph. 32, 4, Article 52 (July 2013), 6 pages. DOI = 10.1145/2461912.2461920 http://doi.acm.org/10.1145/2461912.2461920.

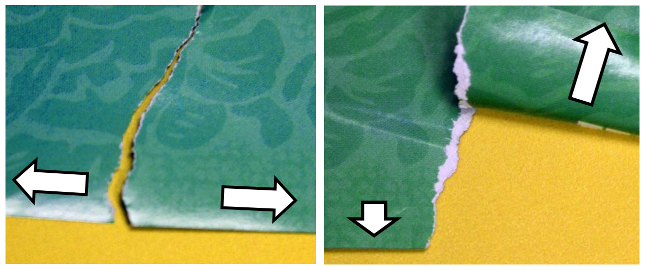
(a) In-plane Stretching (b) Tearing

Figure 2: *Paper fractures. Compared with in-plane stretching in (a), tearing motion is more likely to cause layer separation as shown in (b).*

changes elastostatically after each cut, the method effectively avoids shattering artifacts, which cause an object to break into many small pieces.

- We develop a fracture-aware remeshing scheme based on constrained Delaunay triangulation. We use this scheme to produce fracture details, while keeping the computational cost reasonably small.
- We propose a multi-layered model to handle complex fracture behaviors of layered thin plates, including layer collisions and cut propagation across different layers.

We present several examples to demonstrate the performance of our system. These include simulations of multi-layered paper tearing, tearing of a single foil sheet, and penetration of a paper sheet by a metal ball.

2 Related Work

Thin Plates. Computational approaches for simulating thin plates, such as paper, foil, and cloth, have a long history in computer graphics. While the early work [Provot 1996; Choi and Ko 2002; Bridson et al. 2002] in this field was largely focused on mass-spring systems, the use of the finite element method (FEM) in thin shell simulation [Grinspun et al. 2002; Etzmuß et al. 2003; Volino et al. 2009; Wang et al. 2010; Wang et al. 2011] became an active research topic in recent years. Unlike mass-spring systems, FEM is more suitable for handling viscoelastic characteristics of real-world materials. Alava and Niskanen [2006] described physical properties of paper in details.

Since thin plates have more obvious bending deformation than planar deformation, one important question is: how to robustly and accurately handle bending effects, especially since they are highly nonlinear? Baraff and Witkin derived a hinge-edge bending model using the angle between two neighboring triangles. Bridson and collaborators [2003] improved this model by taking linear momentum preservation into consideration and further developed an implicit method to handle bending damping. Grinspun and colleagues [2003] proposed a discrete shell model to handle bending effects, based on a discrete flexural energy. Assuming that thin plates have no planar deformation, Bergou and colleagues [2006] proposed a quadratic bending energy and developed the corresponding linear bending force formula. Volino and Magnenat-Thalmann [2006] also proposed a linear bending force model. Garg and collaborators [2007] extended this idea to thin shells and developed an implicit bending method under a cubic bending energy. While we use the quadratic bending model proposed by Bergou and colleagues [2006], our system is not limited to any specific elastic solver and it is compatible with other simulators in the future.

Fracture. The seminal work by O'Brien and his collaborators developed FEM-based fracture simulation of both brittle objects [1999] and ductile objects [2002]. Müller and colleagues studied how to simulate fracture animations in real time using a hybrid system. Gingold and colleagues [Gingold et al. 2004] applied FEM to produce the fractures of thin shells. Bao and collaborators [2007] used FEM analysis to animate rigid body fractures. Without changing the geometry, Iben and O'Brien [2006] generated cracking patterns on objects using a simulated stress field. Parker and O'Brien [2009] and Su and colleagues [2009] investigated the efficiency of fracture simulation, especially under game environment. Kaufmann and collaborators [2009] enriched low-resolution fracture cuts with detailed textures. Instead of simulating fractures using meshes, researchers also studied other representations, including particle systems [Boux de Casson and Laugier 2000; Muller 2008], embedded meshes [Sifakis et al. 2007], arbitrary polyhedra [Wicke et al. 2007; Martin et al. 2008], and point clouds [Pauly et al. 2005; Wicke et al. 2005; Guo et al. 2006; Steinemann et al. 2006]. Compared with previous techniques, our system is focused on avoiding artifacts and providing realistic details in the fractures of thin plates. Neither of them has been systematically studied before, as far as we know.

Remeshing. Dynamic remeshing has been extensively used in computer graphics, because of numerous benefits it can provide in many simulation techniques. Researchers have studied the use of dynamic remeshing in simulating a variety of small features, including water droplets [Wicke et al. 2010], cloth wrinkles [Narain et al. 2012] and paper folds [Narain et al. 2013]. In our work, remeshing is employed for modeling detailed fracture patterns.

3 Theoretical Background

Constructing quality meshes of animated objects is of crucial importance for stable FEM simulation. Among meshing algorithms with theoretical quality guarantees, the Delaunay refinement technique is one of the most prominent. See the recent book [Cheng et al. 2012] for more details. The *Delaunay triangulation* of a set of points P in the plane is a triangulation of its convex hull, so that the circumcircle of every triangle is empty of points in P. Delaunay triangulations possess important properties, in particular, they are known to maximize the minimum angle.

A *constrained Delaunay triangulation* guarantees the inclusion of certain edges referred as *constraints*. These edges do not necessarily satisfy Delaunay properties, so the resulting triangulation is not strictly Delaunay.

A *conforming Delaunay triangulation* is a constrained Delaunay triangulation where every constraint is a Delaunay edge. Given the constraints, such triangulations can be computed by iterative refinement of edges and triangles (also known as Delaunay refinement). This is a key component in our algorithm, since we use constrained Delaunay edges to represent fracture lines. More formal definitions and details on the above concepts can be found in [Cheng et al. 2012].

4 Algorithm Overview

Algorithm 1 summarizes the major stages of a single iteration in our system. An iteration starts with a remeshing stage, which ensures that mesh quality in the neighborhood of constrained edges

Algorithm 1 FRACTURESIMULATIONITERATION()

```
Remesh()
for $i = 1 \to N$ do
    ImplicitFEM()
    ApplyBending()
    LimitStrain()
    BreakSprings()
    ApplySpringConstraints()
end for
while a fracturable node $v$ exists do
    Fracture($v$)
    ElastostaticFEM()
end while
LimitStrain()
```

is adequate. Next, we perform N steps of dynamic mesh simulation. (In our experiment, $N = 10$.) During each step, we run an implicit solver first, which takes care of elastic, damping and external forces. Next, we proceed to an explicit solver, which applies quadratic bending forces and handles vertex-triangle collisions. We then use strain limiting on both planar and bending deformation. We break the inter-layer springs, if they are overly stretched. Finally, we apply constraints on the spring lengths, to keep adjacent layers attached.

Once the system finishes dynamic simulation, it examines the mesh for potential cuts. After each cut is done, an elastostatic FEM step is performed locally to relieve internal stress. If there are no more cuts, the system applies strain limiting again and ends this iteration.

5 FEM Simulation

For dynamic simulation of thin plates, we employ the finite element approach similar to the one used in [Etzmuß et al. 2003]. The material surface is discretized as a triangular mesh. The geometry of mesh triangles is stored both in rest and deformed state. At every simulation time step, we extract the deformation gradient for each triangle. Since many thin plates (such as paper and foil) have strong resistance to planar deformation, we assume that the stress linearly depends on the strain. This allows us to use Biot strain and the implicit time integration scheme. We use a linear constitutive model (defined by a Young modulus E and a Poisson ratio ν) to handle the strain-stress relationship. To avoid rotational artifacts, we apply the co-rotational method, by calculating the rotational component $\mathbf{R}$ of the deformation gradient via polar decomposition. Under this setting, linear elastic forces acting on the mesh are computed in the material frame and then rotated back to the world frame:

$$\mathbf{f} = \mathbf{RK}(\mathbf{R}^{-1}\mathbf{p} - \mathbf{x}), \tag{1}$$

where $\mathbf{K}$ is the stiffness matrix, and $\mathbf{p}$ and $\mathbf{x}$ are the world and material coordinates of the vertices respectively. The implicit integration method produces the following linear system with respect to the vertex velocities:

$$(\mathbf{M} - \Delta t^2\hat{\mathbf{K}})\mathbf{v}^{\text{new}} = \mathbf{Mv} + \Delta t(\hat{\mathbf{K}}\mathbf{p} - \hat{\mathbf{f}} + \mathbf{f}), \tag{2}$$

in which $\hat{\mathbf{K}} = \mathbf{RKR}^T$, $\hat{\mathbf{f}} = \mathbf{RKx}$, $\mathbf{v}^{\text{new}}$ is the new velocity at the next time step, $\mathbf{f}$ is the external force, and $\mathbf{M}$ is the mass matrix calculated from material density ρ and thickness d. We solve this system using the preconditioned conjugate gradient method. To avoid in-plane instabilities, we add a Rayleigh damping matrix $\alpha\mathbf{M}+\beta\mathbf{K}$. We also add a small amount of air drag, by multiplying velocities with a damping coefficient γ.

6 Fracture Modeling

In real world, whenever local material stress is sufficiently high, materials may fracture, developing discontinuities in the mesh. After multiple cracks, the quality of the underlying mesh may deteriorate significantly, producing simulation instabilities and artifacts.

To simulate complicated fracture patterns while preserving the quality of the underlying mesh, we use the conforming Delaunay triangulation. Initially, the object is triangulated using a standard Delaunay meshing algorithm. During simulation, whenever a crack is about to appear in the simulated object, we insert the fracture segment as a constraint into the conforming triangulation. This constraint may be subsequently split and the neighborhood of the cut may be automatically retriangulated, keeping the mesh quality in good shape.

We consider several exception cases as did in [O'Brien and Hodgins 1999]. We do not put new vertices into the triangulation that are too close to existing ones. (In our examples, we set the lower bound on the distance between vertices to be 0.001m.) We prevent back-cracking as well, by setting a lower bound on the angle between constraints in the triangulation. (The lower bound is $\pi/16$ in our examples.) We do not enforce the lower bound on the angle between non-constrained edges, since that has been taken care of by the Delaunay triangulation process.

To determine the location and direction of cracks in the material, we use the approach proposed by O'Brien and Hodgins [1999]. At every mesh node, we compute the separation tensor based on the stress values of incident elements. Perpendiculars to the separation tensor eigenvectors represent potential cut directions. Whenever a corresponding eigenvalue is larger than the material threshold, the cut is to be made. However, if all candidate nodes are cut simultaneously during the same timestep, a lot of small pieces can separate from the object, creating glass-like behavior not typical for thin plate materials like paper or cardboard.

Stress Relaxation. We propose a relaxation method to prevent objects from fracturing into small pieces. At each time step, we choose the candidate node with the largest separation tensor eigenvalue first. We compute the crack line perpendicular to the corresponding eigenvector and split the node, putting the intersection of the crack line with the 1-ring neighborhood of the node as a constraint into the triangulation. We note that this causes local remeshing, such as splitting inserted constraints.

Then, we relieve stresses in the elements incident to the inserted crack lines by running a single iteration of elastostatic FEM. We mark all the nodes that belong to the inserted constraints, fix the remaining nodes, and solve for positions of the marked nodes when the mesh becomes static:

$$\mathbf{RK}(\mathbf{R}^{-1}\mathbf{p} - \mathbf{x}) = \mathbf{0}. \tag{3}$$

We solve this sparse linear system using the preconditioned conjugate gradient method. Adjusting positions of the nodes on the cut boundary effectively relieves the stresses in elements, and prevent them from developing oscillations and separating from the mesh. After the stress relaxation step, we proceed with choosing the next best candidate node.

Heterogeneity and anisotropy. To model local variations in cut patterns, we introduce heterogeneity and anisotropy to the simulated materials as described by O'Brien [2003]. Specifically, we apply direction-aware rotation and scaling matrices to the computed stress tensor σ:

$$\sigma^{\text{new}} = (\mathbf{R}^T\mathbf{SR})\sigma(\mathbf{R}^T\mathbf{SR}). \tag{4}$$

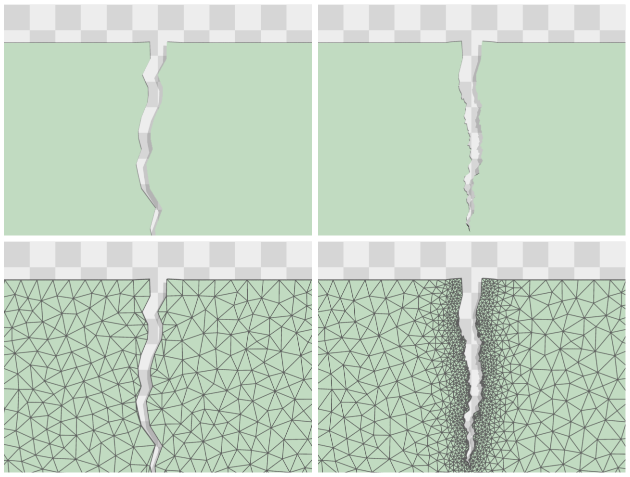

Figure 3: *Refinement comparison. Compared with fracture simulation without refinement (left), fracture simulation with adaptive refinement (right) can produce more fracture details around potential crack regions.*

We define **R** and **S** as two spatially varying matrices that are used to adjust the material strength at each element. The user can specify them using two high-resolution textures. During simulation, **R** and **S** of a specific element are linearly interpolated at the element mass center. This allows us to simulate small-scale changes in fracture directions. (Please watch the accompanying video for an example.)

7 Adaptive Remeshing

Real-world paper materials, unlike glass or ceramics, rarely fracture along straight lines. Its fiber composition results in complicated and highly detailed cut patterns. In physically based simulation, achieving small details on the cut boundary requires meshes to be in sufficiently high resolution. To obtain such result without sacrificing simulation performance, we implement adaptive remeshing with respect to potential crack paths.

Our idea is to perform local refinement where a mesh fracture is about to be initiated. We define a scalar field on the mesh in the material space to control the preferred triangle size. In the neighborhood of potential constrained mesh edges, the scalar field values are linearly proportional to the distance to the closest constraint (subject to a minimum value). This scalar field is passed as a parameter to the standard Delaunay refinement algorithm, which can be found in Chapter 6 of [Cheng et al. 2012]. During Delaunay refinement, extra points (known as *Steiner* points) are inserted into the mesh until certain termination criteria are met. In our case, these criteria involve locally preferred triangle size and element quality.

We note that refining the mesh may involve changes in the existing triangulation constraints. So constrained edges are refined as well. However, the overall geometry of fracture lines does not change. Figure 3 shows a simple refinement example using our method. This example contains from 2K (initial) to 10K triangles (refined), and each frame took 0.025 to 0.1 seconds to simulate. Without using refinement, it takes 1 to 2 seconds to simulate each frame when using a similar quality dense mesh containing 50-52K triangles. Figure 4 compares the simulation time of our method with the simulation times without doing any refinement over 100 frames.

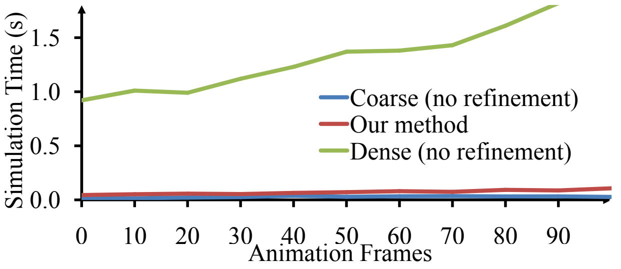

Figure 4: *Simulation times per frame. Compared with the simulations without any refinement, our refinement method can provide fracture details with a small computational overhead.*

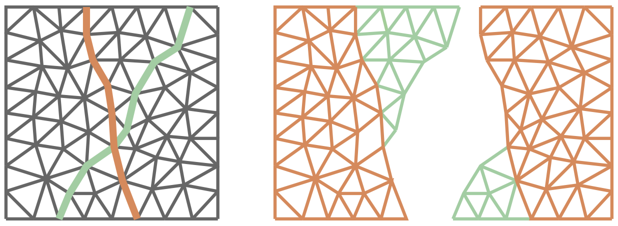

Figure 5: *A two-layered example. The left picture shows the triangulation in the material space. The cuts in the two layers are highlighted. The right picture illustrates a possible look of two fractured pieces in the world space without deformation.*

8 Layer Separation

Thin plates are often made of several layers, which add distinctive behaviors to their fractures. Adjacent layers may develop different fracture patterns, and internal layers may reveal their fiber texture during tearing. This is typically noticed when tearing papers in the real world. We incorporate such multi-layered material composition into our simulation framework.

For a thin plate consisting of multiple layers, we maintain the same triangulation for all layers in the material space. However, since layers may separate and move differently, world space positions for mesh element nodes are defined on a per-layer basis. The problem comes with the fracture: different layers may have different cuts. We found that a simple solution is to put the cuts from all layers as constraints into the sole rest frame triangulation, and assign lists of marks to the constrained edges, representing the layers that were cut along these edges. Hence, multiple layers in the material space can share the same triangulation, even though they are different in the world space as Figure 5 shows.

Using the same triangulation, we can conveniently model interaction among layers. Our interaction scheme is based on the spring model. Initially, we connect all nodes in adjacent layers by short adhesive springs with length Δ and stiffness k. We simulate spring dynamics using the implicit method proposed by Choi and Ko [2002]. Large spring deformation results in its breakage, which models layer separation. To prevent layer collisions and unrealistic gaps between layers, we enforce strain limiting on all of the springs. Specifically, we restrict the stretching ratio to be within $[80\%, 120\%]$. When the stretching ratio of a spring reaches 120%, we break it. In addition, we enforce correct visual ordering between layers. For every spring, we compare its direction to the direction of average normal of its endpoints, and swap the endpoint nodes if the directions do not match. This allows us to avoid self collision issues, without doing actual collision tests.

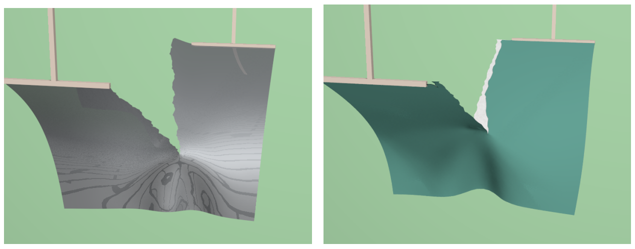
(a) Foil tearing (b) Paper tearing

Figure 6: *Tearing of different thin plates.*

9 Results

(Please see the accompanying video for our animation results.) We implemented our algorithm in C++ with the use of CGAL library [CGAL 2013]. The code runs on an Intel Core i5-2500K desktop with 8GB main memory. The ranges of parameter values used in our examples are summarized in Table 1. We use the same triangle quality criteria to generate our examples. Specifically, the upper bound on the ratio r/l is $\sqrt{2}$. Here r is the triangle circumradius and l is the longest edge length.

Single-layer tearing. Our first example (in Figure 6a) demonstrates tearing a single-layer object, i.e., a thin $1\text{m}\times 1\text{m}$ sheet of foil. The simulation mesh contains 10K triangles initially.

Multi-layer tearing. The second example (Figure 6b) shows multi-layered behaviors. Tearing of a colored $1\text{m}\times 1\text{m}$ paper sheet results in layer separation, with the inner layer opened up, revealing its fiber texture. Each layer contains 5K triangles initially.

Penetration. Our final example is tearing of a paper sheet by a flying metal ball (in Figure 7). The paper sheet has size $1\text{m}\times 1\text{m}$, and the ball has a radius of 0.2m. The final mesh contains approximately 4K triangles, and the triangle size varies from 7.5 to 60mm. Several animation frames are shown in Figure 1, and a simulated mesh example is given in Figure 7.

10 Limitations

Our fracture simulation approach has several limitations. Simulation of materials with high bending resistance such as stiff papers requires an appropriate bending model. The explicit quadratic bending scheme that we are currently using imposes restrictions on the simulation time step. Simulating multi-layered structures also presents certain challenges. Using the same triangulation for all of the layers simplifies the layer modeling problem, but it may cause an unnecessarily dense triangulation when handling multiple layers and cuts. Finally, additional efforts are required to model thin fibers on paper cut boundaries, due to real-world composition of paper.

11 Conclusions and Future Work

We presented a method to simulate tearing of multi-layered thin plate objects, such as paper or cardboard. Many avenues for potential future improvements exist. We plan to investigate methods to add fibers to paper cut boundaries, using either example-based or physically based approaches. Our current remeshing algorithm does not simulate paper folding, and we are interested in incorporating the method proposed by Narain and colleagues [2013] into our system. To reduce the computational cost, we applied the elastostatic relaxation step to the 1-ring neighborhood of the vertices around the cut boundary only. Ideally, it should be applied to a fixed-radius neighborhood to prevent potential artifacts on dense meshes. Finally, we plan to extend our system to handle layered structures with non-trivial thickness, such as plywood.

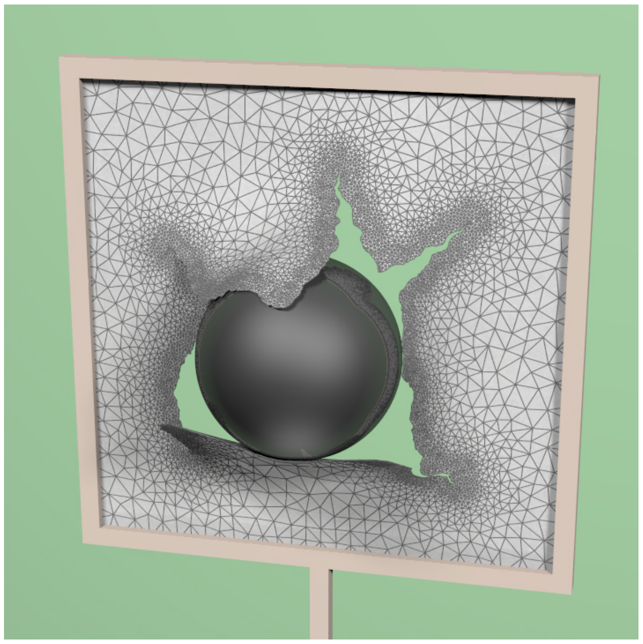
Figure 7: *A metal ball penetrating a paper sheet.*

Acknowledgments

We thank James F. O'Brien and the computer graphics group at UC Berkeley for suggestions and comments. We also thank NVIDIA for supporting our work through equipment and funding.

References

ALAVA, M., AND NISKANEN, K. 2006. The physics of paper. *Reports on Progress in Physics 69*, 3, 669.

BAO, Z., HONG, J.-M., TERAN, J., AND FEDKIW, R. 2007. Fracturing rigid materials. *IEEE Transactions on Visualization and Computer Graphics 13*, 2 (Mar.), 370–378.

BERGOU, M., WARDETZKY, M., HARMON, D., ZORIN, D., AND GRINSPUN, E. 2006. A quadratic bending model for inextensible surfaces. In *Proc. of SGP*, 227–230.

BOUX DE CASSON, F., AND LAUGIER, C. 2000. Simulating 2d tearing phenomena for interactive medical surgery simulators. In *Proc. of the Computer Animation*, 9–.

BRIDSON, R., FEDKIW, R., AND ANDERSON, J. 2002. Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph. (SIGGRAPH) 21*, 3 (July), 594–603.

BRIDSON, R., MARINO, S., AND FEDKIW, R. 2003. Simulation of clothing with folds and wrinkles. In *Proc. of SCA*, 28–36.

CGAL, 2013. Computational Geometry Algorithms Library. http://www.cgal.org.

Notation	Parameter name	Unit	Range
ρ	Material density (Section 5)	kg/m^3	$[0.7 \times 10^3, 1.2 \times 10^3]$
E	Young modulus (Section 5)	N/m^2	$[1.5 \times 10^6, 2 \times 10^6]$
ν	Poisson ratio (Section 5)	1	$[0.15, 0.2]$
d	Material thickness (Section 5)	m	$[0.2 \times 10^{-3}, 0.3 \times 10^{-3}]$
Δ	Gap between layers (Section 8)	m	$[0.1 \times 10^{-3}, 0.2 \times 10^{-3}]$
k	Stiffness of inter-layer springs (Section 8)	N/m	$[1 \times 10^4, 5 \times 10^4]$
α	Rayleigh mass matrix damping coefficient (Section 5)	1	$[0.4, 0.5]$
β	Rayleigh stiffness matrix damping coefficient (Section 5)	1	$[0.05, 0.1]$
γ	Air drag coefficient (Section 5)	1	$[0.9, 0.9995]$

Table 1: *Parameters used in our system.*

CHENG, S.-W., DEY, T. K., AND SHEWCHUK, J. R. 2012. *Delaunay Mesh Generation*. CRC Press, Boca Raton, Florida.

CHOI, K.-J., AND KO, H.-S. 2002. Stable but responsive cloth. *ACM Trans. Graph. (SIGGRAPH) 21*, 3 (July), 604–611.

ETZMUß, O., KECKEISEN, M., AND STRAßER, W. 2003. A fast finite element solution for cloth modelling. In *Proc. of Pacific Graphics*, 244–.

GARG, A., GRINSPUN, E., WARDETZKY, M., AND ZORIN, D. 2007. Cubic shells. In *Proc. of SCA*, 91–98.

GINGOLD, Y., SECORD, A., HAN, J. Y., GRINSPUN, E., AND ZORIN, D. 2004. A discrete model for inelastic deformation of thin shells. Tech. rep., Aug.

GRINSPUN, E., KRYSL, P., AND SCHRÖDER, P. 2002. Charms: a simple framework for adaptive simulation. *ACM Trans. Graph. (SIGGRAPH) 21*, 3 (July), 281–290.

GRINSPUN, E., HIRANI, A. N., DESBRUN, M., AND SCHRÖDER, P. 2003. Discrete shells. In *Proc. of SCA*, 62–67.

GUO, X., LI, X., BAO, Y., GU, X., AND QIN, H. 2006. Meshless thin-shell simulation based on global conformal parameterization. *IEEE Transactions on Visualization and Computer Graphics 12*, 3 (May), 375–385.

IBEN, H. N., AND O'BRIEN, J. F. 2006. Generating surface crack patterns. In *Proc. of SCA*, 177–185.

KAUFMANN, P., MARTIN, S., BOTSCH, M., GRINSPUN, E., AND GROSS, M. 2009. Enrichment textures for detailed cutting of shells. *ACM Trans. Graph. (SIGGRAPH) 28*, 3 (July), 50:1–50:10.

MARTIN, S., KAUFMANN, P., BOTSCH, M., WICKE, M., AND GROSS, M. 2008. Polyhedral finite elements using harmonic basis functions. In *Proc. of SGP*, 1521–1529.

MULLER, M. 2008. Hierarchical position based dynamics. In *VRIPHYS*, 1–10.

NARAIN, R., SAMII, A., AND O'BRIEN, J. F. 2012. Adaptive anisotropic remeshing for cloth simulation. *ACM Trans. Graph. (SIGGRAPH Asia) 31*, 6 (Nov.), 152:1–152:10.

NARAIN, R., PFAFF, T., AND O'BRIEN, J. F. 2013. Folding and crumpling adaptive sheets. *ACM Trans. Graph. (SIGGRAPH)*.

O'BRIEN, J. F., AND HODGINS, J. K. 1999. Graphical modeling and animation of brittle fracture. In *Proc. of SIGGRAPH 98*, Annual Conference Series, 137–146.

O'BRIEN, J. F., BARGTEIL, A. W., AND HODGINS, J. K. 2002. Graphical modeling and animation of ductile fracture. *ACM Trans. Graph. (SIGGRAPH) 21*, 3 (July), 291–294.

O'BRIEN, J. F. 2003. *Graphical Modeling and Animation of Brittle Fracture*. PhD thesis, Georgia Institute of Technology, Atlanta, GA.

PARKER, E. G., AND O'BRIEN, J. F. 2009. Real-time deformation and fracture in a game environment. In *Proc. of SCA*, 165–175.

PAULY, M., KEISER, R., ADAMS, B., DUTRÉ, P., GROSS, M., AND GUIBAS, L. J. 2005. Meshless animation of fracturing solids. *ACM Trans. Graph. (SIGGRAPH) 24*, 3 (July), 957–964.

PROVOT, X. 1996. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *Proc. of Graphics Interface*, 147–154.

SIFAKIS, E., DER, K. G., AND FEDKIW, R. 2007. Arbitrary cutting of deformable tetrahedralized objects. In *Proc. of SCA*, 73–80.

STEINEMANN, D., OTADUY, M. A., AND GROSS, M. 2006. Fast arbitrary splitting of deforming objects. In *Proc. of SCA*, 63–72.

SU, J., SCHROEDER, C., AND FEDKIW, R. 2009. Energy stability and fracture for frame rate rigid body simulations. In *Proc. of SCA*, 155–164.

VOLINO, P., AND MAGNENAT-THALMANN, N. 2006. Simple linear bending stiffness in particle systems. In *Proc. of SCA*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 101–105.

VOLINO, P., MAGNENAT-THALMANN, N., AND FAURE, F. 2009. A simple approach to nonlinear tensile stiffness for accurate cloth simulation. *ACM Trans. Graph. 28*, 4 (Sept.), 105:1–105:16.

WANG, H., O'BRIEN, J., AND RAMAMOORTHI, R. 2010. Multiresolution isotropic strain limiting. *ACM Trans. Graph. (SIGGRAPH Asia) 29*, 6 (Dec.), 156:1–156:10.

WANG, H., O'BRIEN, J. F., AND RAMAMOORTHI, R. 2011. Data-driven elastic models for cloth: Modeling and measurement. *ACM Trans. Graph. (SIGGRAPH) 30*, 4 (July), 71:1–71:12.

WICKE, M., STEINEMANN, D., AND GROSS, M. H. 2005. Efficient animation of point-sampled thin shells. *Computer Graphics Forum (Eurographics) 24*, 3, 667C–676.

WICKE, M., BOTSCH, M., AND GROSS, M. 2007. A finite element method on convex polyhedra. *Computer Graphics Forum (Eurographics) 26*, 3, 355C–364.

WICKE, M., RITCHIE, D., KLINGNER, B. M., BURKE, S., SHEWCHUK, J. R., AND O'BRIEN, J. F. 2010. Dynamic local remeshing for elastoplastic simulation. *ACM Trans. Graph. 29* (July), 49:1–49:11.