Adaptive Tearing and Cracking of Thin Sheets

Tobias Pfaff Rahul Narain Juan Miguel de Joya James F. O'Brien

University of California, Berkeley

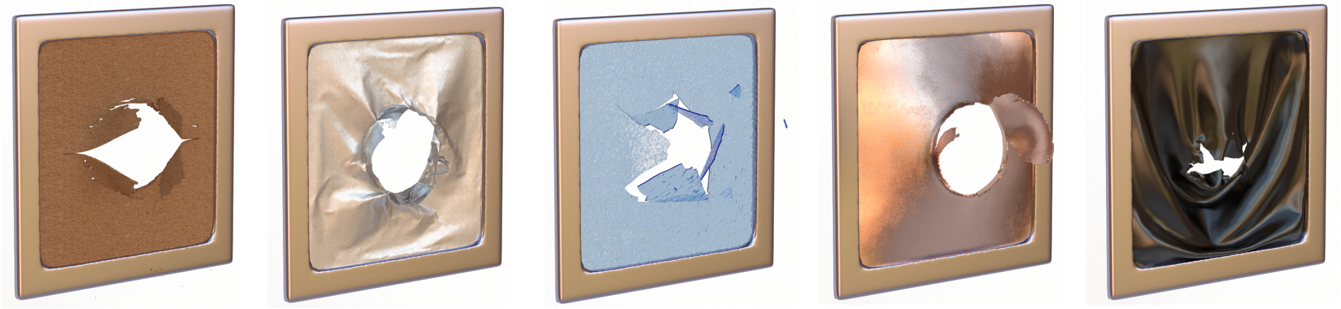


Figure 1: *Our method produces realistic tearing and cracking phenomena for thin sheets made from a wide variety of materials such as cork, foil, plastic, metal, or vinyl. These results are achieved using simulation on adaptive meshes that resolve fracture behavior at very high resolution.*

Abstract

This paper presents a method for adaptive fracture propagation in thin sheets. A high-quality triangle mesh is dynamically restructured to adaptively maintain detail wherever it is required by the simulation. These requirements include refining where cracks are likely to either start or advance. Refinement ensures that the stress distribution around the crack tip is well resolved, which is vital for creating highly detailed, realistic crack paths. The dynamic meshing framework allows subsequent coarsening once areas are no longer likely to produce cracking. This coarsening allows efficient simulation by reducing the total number of active nodes and by preventing the formation of thin slivers around the crack path. A local reprojection scheme and a substepping fracture process help to ensure stability and prevent a loss of plasticity during remeshing. By including bending and stretching plasticity models, the method is able to simulate a large range of materials with very different fracture behaviors.

Keywords: Fracture simulation, cracking, tearing, thin sheets, shells, plastic deformation, bending.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics—Animation; I.6.8 [Simulation and Modeling]: Types of Simulation—Animation. A.3.1 [Animals]: Types of Birds—Ducks.

Links: DL PDF VIDEO WEB CODE

From the conference proceedings of ACM SIGGRAPH 2014.
Appearing in ACM Transaction on Graphics Vol. 33, No. 4.

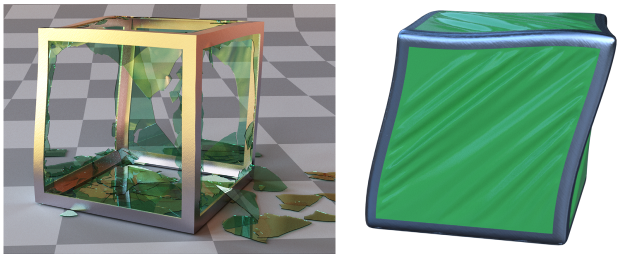
1 Introduction

While the simulation of fracture dynamics has a long history in computer graphics, reproducing the variety and detail of fracture patterns observed in real-world materials remains a difficult problem. Many of the objects exhibiting interesting breaking behavior can be well approximated using thin-shell models represented as triangle meshes. Triangle meshes are computationally less expensive than equivalent tetrahedralized volumes, and they allow for simpler, more flexible algorithms for adaptive remeshing. In this work, we therefore focus on techniques for high-fidelity simulation of cracks and tears in thin sheets with adaptive triangle meshes.

The distinctive crack patterns observed in many materials arise due to small-scale interactions between elastic strain, plastic yielding, and material failure. Stress gradients can be very large near the crack tip where the stress field often approaches singularity. Using adaptive meshing, the resolution around potential fracture regions can be increased to accurately resolve these interactions and produce realistic behavior. Subsequent coarsening, done once stresses have been reduced, avoids continually increasing computational cost and keeps overall simulation times reasonable. Adaptive mesh refinement also permits nearly arbitrary crack paths by removing the ill-shaped triangles generated when existing elements are split along the crack's path.

The use of aggressive adaptive remeshing in conjunction with fracture simulation introduces some challenges. Most of them are related to non-physical stresses and diffusion that may be created during remeshing and that can lead to spurious crack initiation, unrealistic crack boundaries, and loss of shape. Below, we summarize our solutions to these issues which makes the powerful combination of adaptivity and fracture simulation usable in practice.

Remeshing Many interesting fracture patterns occur in materials with high stiffness, such as glass or iron. In these materials, even small distortions commonly introduced by vertex repositioning during remeshing can cause high stresses. If the simulation does not include fracture, then these distortions only lead to minor popping artifacts as small transient disturbances are smoothed away by the material's natural dynamics. However, when fracture is enabled they lead to spurious breaking. The problem is exacerbated when refinement and coarsening happen aggressively around the crack tip. In §5, we present a reprojection scheme which stabilizes stresses



Images copyright Tobias Pfaff, Rahul Narain, Juan Miguel de Joya, James F. O'Brien.

Figure 2: *The left image shows a multi-material simulation of glass panes in a aluminum frame. The plastic bending of the frame induces shearing fracture of the glass, similar to when a bear bounces on top of one's car to break the windows. On the right, a quasi-static state is shown, with the glass panes replaced by rubber sheets. Using a single global embedding for plasticity would impose the plastic deformation from the frame on the glass, forcing it into an artificial plastic shape similar to the quasi-static state shown right.*

during remeshing. We also discuss how to coarsen and refine curved surfaces without loss of shape.

Plasticity Interpolating plasticity tensors during remeshing causes diffusion and artificial recovery from past plastic deformation. However, techniques to prevent this plasticity loss, such as plastic embedding [Wicke et al. 2010a; Narain et al. 2013], don't work well for combined stretching and bending plasticity. They also may produce artificial plasticity in connected regions with different plastic material parameters (Fig. 2). In §5.2, we propose a novel way of redistributing the plasticity tensors during remeshing, which avoids both plastic diffusion and artificial plasticity buildup. In contrast to existing methods, the new method also deals with refining plasticity on curved surfaces in a principled way.

Fracture model When a crack forms, the stress perpendicular to crack's path is relieved. Further deformation then concentrates stress at the new crack tip, which in turn fails, allowing the crack to propagate. This stress redistribution happens as the node positions are updated in a physics or relaxation step. As a result, only one crack increment can be accurately computed per simulation step. In §4.1, we introduce an algorithm which allows multiple propagating cracks while obeying physical constraints. Additionally, we discuss an improved fracture criterion, which unlike the separation tensor proposed by O'Brien and Hodgins [1999] does not depend on triangle size and is better suited for adaptive simulation.

Directibility Finally, we explore possibilities for art direction for adaptive fracture simulation in §6. The ability to direct cracks is an important tool, as crack paths are very hard to predict from the scene setup and material parameters alone. We show that our adaptive fracture algorithm facilitates the use of weakened zones and defect lines, which can be specified independent of the initial mesh and then conformingly meshed for fracture boundaries only as needed.

2 Related Work

Dynamics simulation There is a large body of literature on the simulation of cloth and thin shells in computer graphics. Many of these methods deal with the simulation of elastic sheets [Grinspun et al. 2003; Bridson et al. 2003; Narain et al. 2012]. Volume-preserving multiplicative plasticity models have been explored for volumetric meshes [Bargteil et al. 2007; Wojtan and Turk 2008], but are seldom applied to sheet models. Wicke et al. [2010b] introduced a plastic embedding to prevent diffusion of plasticity in volumetric materials, which was adapted to bending plasticity in sheets by Narain et al. [2013]. However, for sheets with both bending and stretching plasticity their embedding approach encounters problems. For example, local plastic bending of a fixed sheet will induce artificial stretching plasticity in the embedding. For the fourth example in Fig. 1 for example, on average over 50% of the plasticity could not be embedded and must be stored as a residual, which leads to instabilities during remeshing.

Adaptive remeshing In recent years, there has been a growing interest in adaptive remeshing techniques for simulation. In particular, dynamic remeshing has been used for fluid simulation [Klingner et al. 2006; Chentanez et al. 2007; Ando et al. 2013; Clausen et al. 2013], viscoelastic bodies [Bargteil et al. 2007; Wojtan and Turk 2008; Wicke et al. 2010b] and thin sheet simulation [Hutchinson et al. 1996; Simnett et al. 2009; Narain et al. 2012; Busaryev et al. 2013]. For stiff materials, however, these methods exhibit abnormal stresses due to vertex jittering during remeshing. This issue can be mitigated by a post-remeshing reprojection step [Narain et al. 2013]. For adaptive breaking, that approach becomes infeasibly expensive, as it would need to be executed many times per timestep. In addition, none of the above methods deal with coarsening and refinement of curved rest shapes.

Fracture Early work on fracture simulation in computer graphics dates back to the papers by Terzopoulos et al. [1988] and Norton et al. [1991], which use a simple stretching limit as the breaking criterion. Many newer fracture methods in computer graphics make use of the separation tensor [O'Brien and Hodgins 1999], which provides a per-vertex maximum stress criterion. However, that criterion is dependent on the size and shape of neighboring elements. For uniform-resolution meshes, such a dependence is tolerable, but in the context of adaptive refinement it becomes problematic.

While the above methods represent fracture directly by splitting apart elements, others researchers have investigated alternative representations. Virtual node methods [Molino et al. 2004; Sifakis et al. 2007] embed cracks in virtual elements to prevent stability issues. However, embedding also limits the accuracy of the crack propagation computations, which made prescribed crack paths necessary for many of their examples. XFEM embedding [Kaufmann et al. 2009] on the other hand can accurately resolve dynamics along the crack path, but imposes limits on the fracture geometry and can be costly for complicated fracture patterns. Recently, many methods make use of Voronoi diagrams or prescribed crack paths to generate crack surfaces [Bao et al. 2007; Su et al. 2009; Müller et al. 2013]. Some materials, like unreinforced concrete, break in ways that are somewhat reminiscent of Voronoi regions, but for most materials these Voronoi-based boundaries are unrealistic. Fig. 10 (right) shows an example of such non-Voronoi shapes seen in real fracture.

To allow the opening of more then one crack per timestep, residual propagation [O'Brien 2000] can be used to propagate the momentum of the crack forward. Busaryev et al. [Busaryev et al. 2013] perform a relaxation step after each crack opening to approximate the change in the stress distribution. Their approach however does not take the crack momentum and collision constraints into account.

Many existing methods for dynamic fracture use volumetric representations, such as the above tetrahedral methods or so-called meshless methods (e.g. [Pauly et al. 2005]). There are however a few existing examples of surface-based methods. Gingold et al. [2004] present a unified bending and stretching strain model for fracturing thin shells, which we use as a starting point for our model. Their static meshing and simplistic fracture criterion can however lead to unrealistic fracture patterns. Busaryev et al. [2013] use a layering model and dynamic remeshing to achieve high-resolution tearing of multi-layered sheets. However, the range of fracture behavior that method can realize is limited by the lack of plasticity and their fracture criterion being independent of bending. Further, they lose efficiency by using only isotropic Delaunay meshes that do not coarsen once excess triangles are no longer required. It would be interesting to combine their layering algorithm with our method.

Other related methods include Iben et al. [2006] who compute static fracture patterns for texturing, and Glondu et al. [2012] who use Bayesian optimization to generate example-based crack patterns.

3 Overview

The dynamics of our model are simulated by solving the elasto-plastic equations on a triangulated finite element mesh. We build on the publicly available ARCSim[1] code [Narain et al. 2012; Narain et al. 2013], which implements elasticity using Green strains and discrete hinges [Bridson et al. 2003; Grinspun et al. 2003], and resolves collisions using repulsive forces and impact zones [Bridson et al. 2002; Harmon et al. 2008]. We use this model to compute the basic sheet dynamics, and modify it so that plasticity and fracture may be modeled. The simulation loop and underlying elasticity model is summarized in this section; our additions to the model are described in §4–§6.

3.1 Simulation Loop

We perform our simulation on a triangulated mesh. Each vertex has a position in undeformed reference space $\mathbf{u}$ and in world space $\mathbf{x}$, and a velocity $\mathbf{v}$. Each face stores a thickness h, and the plasticity tensors $\mathbf{F}_p$ and $\mathbf{S}_p$ which respectively describe stretching and bending plasticity. The thickness remains constant in elastic simulations, but can change due to plastic stretching as explained in §4.2.

At each timestep, we compute a conventional physics update as a first step. That is, the elasto-dynamic equations (§3.2) are integrated to obtain an updated velocity and position for each vertex, and collisions are handled. Next, we account for plastic yielding by updating the plasticity tensors as described in §4.2. Based on the updated configuration, we can now evaluate the fracture criterion, and split the mesh where the material fails. During each timestep, more than one crack can open, therefore we perform this evaluation and splitting multiple times per timestep. To ensure good resolution during the fracture process, the mesh is also remeshed locally around the opened cracks in each substep. This fracture substepping algorithm is described in §4.1 and summarized in Algorithm 1. After the fracture substepping is complete, a global remeshing step is performed for the whole mesh. This step coarsens around newly formed cracks once the local stress is relieved, and regulates triangle size in regions of the mesh where no fracture is active. Finally, the simulation time is incremented and the next timestep begins.

3.2 Elasticity Model

Our elasticity model is very similar to the one of Narain et al. [2012; 2013]. However, where they use a two-dimensional parametrization space, we instead embed the reference shape in three dimensions so that we can represent arbitrary shapes, such as spheres, which cannot be flattened into the plane. The deformation gradient $\mathbf{F} = \nabla_{\mathbf{u}}\mathbf{x}$ is thus a 3×3 matrix. Using the deformation gradient, Green's strain for in-plane stretching is given by

$$\mathbf{G} = \frac{1}{2}(\mathbf{F}^T\mathbf{F} - \mathbf{I})\,. \tag{1}$$

The in-plane stress can be calculated using the material model

$$\boldsymbol{\sigma} = k_s\left((1-\nu)\,\mathbf{G} + \nu\,\mathrm{tr}(\mathbf{G})\,\mathbf{I}\right)\,, \tag{2}$$

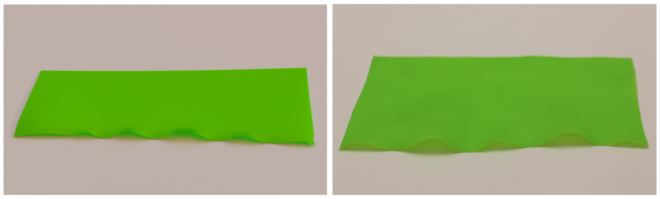


Figure 3: *We simulate the tearing of a rubber sheet (left image) and compare it to a photograph of a real-world tearing experiment (right image). In both simulation and photograph, undulations on the fractured edge due to plasticity around the crack tip are visible.*

where $k_s = Y\,h/(1-\nu^2)$ is the stretching stiffness, Y is Young's modulus, h the material thickness and ν Poisson's ratio. These quantities are computed per triangle.

Similarly, the discrete bending strain of a triangle [Gingold et al. 2004] can be obtained via the dihedral angle θ_i at its three edges,

$$\mathbf{S} = \frac{1}{2A}\sum_{i=1}^{3}(\theta_i - \theta_i^R)\,l_i\,\mathbf{t}_i\,\mathbf{t}_i^T \quad . \tag{3}$$

Here, A denotes the triangle area, l_i the edge lengths and $\mathbf{t_i}$ the unit-length edge tangents. The dihedral angle is measured with respect to the rest pose angle θ_i^R.

The potential energy of the simulated sheet is now given as

$$E = \sum_{i}^{\text{faces}}\frac{A_i}{2}\mathbf{G}_i : \boldsymbol{\sigma}_i + \sum_{i}^{\text{edges}} k_b\frac{l_i^2}{8\,A_i}(\theta_i - \theta_i^R)^2 \tag{4}$$

with the bending stiffness $k_b = \alpha\,k_s\,h^2/12$. The factor α allows modeling materials with anisotropic stiffness along the out-of-plane direction (e.g. rolled steel). The equations of motion can be derived from this potential. To include the effects of plasticity, the dihedral angle is measured against the sum of rest pose angle θ^R and plastic angle θ^p, and the deformation gradient $\mathbf{F}$ is replaced by the purely elastic deformation gradient $\mathbf{F}_{el}$. In §4.2 and §5.2 we discuss these plastic quantities.

4 Fracture and Plasticity

The characteristic appearance of many materials as they crack or tear occurs due to complex emergent behavior driven by interactions between the fracture process, elastic deformation, plastic yielding, and the material's constitutive model [O'Brien et al. 2002]. As shown by the example in Fig. 3, even apparently simple fracture effects may involve subtle, yet significant, plasticity. In the case of thin sheets, the combination of in-plane properties and out-of-plane bending properties further enriches the range of behaviors that may be encountered.

4.1 Fracture model

In order to initiate and propagate cracks, we first need to determine where the material fails. Fracture occurs due to concentrations of stress, so at each timestep we evaluate stress at each node of the mesh to determine if it is high enough to cause material failure. In accordance with O'Brien and Hodgins [1999], we take the approach of splitting at vertices, as this allows for natural extension of existing cracks.

Unfortunately, the naïve approach of simply averaging the stress around a vertex leads to vertex stresses that may not be maximal

[1] http://graphics.berkeley.edu/resources/ARCSim

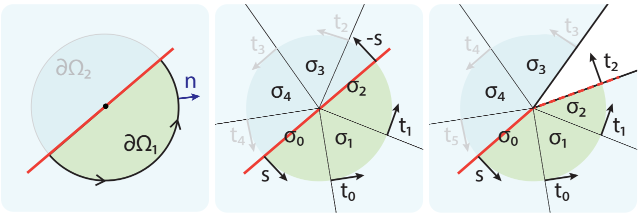


Figure 4: *A splitting plane, shown in red, with normal* $\mathbf{s}$ *separates the surrounding triangles of a vertex into two segments. The traction across the potential split can be computed from the surrounding stress field via a path integral in the continuous case (left), or using the tangent vectors* $\mathbf{t}$ *in the discrete case (middle). If the splitting plane is close to an open crack (right), the crack edge is used as the second hand of the split plane to divide the segments.*

Algorithm 1 Fracture substepping

```
Compute separation strength $s$ for all vertices (Eq. 7)
Add nodes with $s > \tau$ to fracture zone $Z$
Store vertex positions in $x'$

while $Z \neq \emptyset$ do
    $Z' \leftarrow Z$ and $n$-ring neighbors
    Locally remesh $Z'$
    Compute physics update step in $Z'$
    Recompute $s$ for vertices in $Z'$
    Rewind vertex positions in $Z'$ to $x'$
    if any $s > \tau$ then
        Break vertex with max. $s$
    $Z \leftarrow$ all vertices $\in Z'$ with $s > \tau$
```

at the crack tip. Consequently, materials will not break cleanly but instead produce spurious cracks and jagged edges. This problem motivated the use of the separation tensor [O'Brien and Hodgins 1999] as a fracture criterion. However, their separation tensor has the drawback of being too strongly influenced by element size and shape. While that dependence is somewhat tolerable for isotropic meshes with uniformly sized elements, it creates problems in the context of adaptive meshes. Accordingly, we use a different vertex-centric fracture criterion that captures the direction of maximum stress relief independent of element size and shape.

For a given vertex, our approach computes the separation strength based on the amount of stress that would be relieved by splitting the vertex. Accordingly, we first need to find the optimal splitting plane which would maximally relieve the stress around that vertex. By formulating a computable measure of that relief, we can solve a small optimization problem to find the appropriate splitting plane.

Our measure of separation evaluates the stress in a small disc centered at the vertex. Given a potential splitting plane, we can calculate the traction across the plane using the path integral along the disc boundary

$$\mathbf{q}_1 = \int_{\partial\Omega_1} \boldsymbol{\sigma}^+ \cdot \hat{\mathbf{n}}\, dS \quad . \tag{5}$$

where $\hat{\mathbf{n}}$ is the outwards unit normal. In the above equation, we clamp the negative eigenvalues of the stress tensor $\boldsymbol{\sigma}$ to zero to prevent fracture due to compression. As we use a linear FEM discretization, the stress $\boldsymbol{\sigma}$ is piecewise constant on sectors defined by the surrounding triangles (Fig. 4), which gives

$$\mathbf{q}_1 = \boldsymbol{\sigma}_0^+(\mathbf{t}_0 - \mathbf{s}) + \sum_{i=0}^{N-1} \boldsymbol{\sigma}_i^+(\mathbf{t}_{i+1} - \mathbf{t}_i) + \boldsymbol{\sigma}_N^+(-\mathbf{s} - \mathbf{t}_N) \tag{6}$$

where $\mathbf{t}_i$ are the tangent vectors between the triangles and $\mathbf{s}$ is the split plane normal. $\mathbf{q}_2$ is computed similarly by integrating over the segment on the opposite side of the potential split plane. When propagating existing cracks, we divide the segments using one hand of the split plane, and the existing crack edge on the other side, to accurately compute curved crack paths and prevent back-cracking (Fig. 4).

For balanced loads, the difference of $\mathbf{q}_1$ and $\mathbf{q}_2$ of a given vertex and split plane can be used to predict material failure. To prevent the fracture criterion from responding to unbalanced loads, which would produce rigid motion, we instead use the smallest common component

$$s = \min(\mathbf{q_1} \cdot \hat{\mathbf{q}}, -\mathbf{q_2} \cdot \hat{\mathbf{q}}) \tag{7}$$

as the separation strength. $\hat{\mathbf{q}}$ is the normalized difference of the traction vectors $\mathbf{q_1}$,$\mathbf{q_2}$. If s is above the material's area toughness τ, the material fails at this vertex, and the neighboring triangles are split along the split plane. The area toughness is derived from the volume toughness by multiplying with the face thickness $\tau = \tau_V \cdot h$.

There is no easy-to-compute analytical solution to finding the split plane with maximum separation strength. However, because $\boldsymbol{\sigma}$ and the products in (6) can be precomputed for all tested separation planes at a given vertex, we can easily sample this equation (we use 200 samples in practice). Additionally, we avoid computing the separation strength in low-stress regions. The magnitude of the separation strength is upper-bounded by $s_{\text{bnd}} = 2 \max \lambda_i$, where λ_i is the largest eigenvalue of $\boldsymbol{\sigma}_i^+$, so we can ignore vertices where s_{bnd} is below the area toughness.

Bending fracture So far, we have only considered fracture due to in-plane stress. However, for many materials, such as sheets of glass or ceramics, the predominant fracture mechanism is bending strain. We include bending fracture into our model in a similar way as Gingold et al. [2004]. Bending of a thin sheet introduces additional in-plane strain. It is maximal at the outer layer, where an additional deformation of $\Delta l = \frac{1}{2} h\kappa$, with the sheet thickness h and curvature κ, is induced. In three dimensions, we can express this effect in our material model by multiplying the deformation gradient $\mathbf{F}$ with a bending factor

$$\mathbf{F}_B = \mathbf{F}\Big(1 + \frac{1}{2}\alpha h \mathbf{II}^\star\Big) \quad . \tag{8}$$

The second fundamental form $\mathbf{II}$ is calculated using the finite element derivative of the vertex normals. Negative curvature also results in additional tensile strain, just on the opposite side of the sheet. Therefore, we reverse negative eigenvalues in $\mathbf{II}$ to obtain an all-positive tensor $\mathbf{II}^\star$. As the stress tensor used in (6) is computed using the Green strain, including the effects of bending in the fracture model now reduces to replacing $\mathbf{F}$ with $\mathbf{F}_B$ in the calculation of the Green strain (1).

Fracture substepping If the separation strength exceeds the material toughness at a vertex, the material fractures at this point, and its adjacent triangles are split by the separation plane. As we intend to resolve the crack propagation process with fine geometry, many triangle splits will occur per simulation step. To accurately predict new crack openings, the stress relief from the previous crack needs to be taken into account. On the other hand, we note that the stress redistribution on these short timescales is strongest around the newly formed crack tip. We therefore propose a local virtual physics update around the newly formed crack.

First, all potential crack tips (vertices with $s > \tau$) and their n-ring neighborhood are collected into a fracture zone. To relieve bending stress, we need at least a two-ring neighborhood. We use

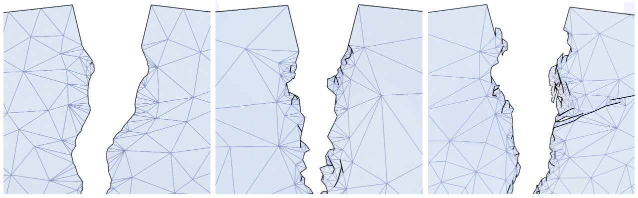


Figure 5: *In this example, we tear a sheet with our algorithm (left) and the separation tensor of O'Brien and Hodgins [1999] with fracture substepping (middle) and without substepping (right). Our algorithm produces significantly fewer artifacts in simulations with varying triangle size.*

$n = 3$ for most of our examples. Next, a local remeshing of this zone is performed (see §5), to make sure stress redistribution and crack opening happens at a high resolution. A local virtual physics update is performed for the vertices in the zone, and the stress and separation strength is updated based on the updated positions for all zone nodes. We select the node with the largest s, and open or propagate a crack at this vertex with the appropriate spliting plane. Then, the node positions are rewound to their original position, and the process is repeated until no vertices with $s > \tau$ are found.

This process is summarized in Algorithm 1. Note that at the end of the process, the node positions remain unchanged, only the topology has changed. The actual node displacement and stress redistribution will happen within the next full physics step. The substepping only locally predicts the stress values to compute crack opening. This restriction is important to correctly handle collisions, and potential long-range effects not handled by the local prediction.

4.2 Plasticity

To be able to simulate materials such as sheet metal, we need to represent plasticity of both tensile and bending stress. For in-plane tensile stresses we use a multiplicative model [Bargteil et al. 2007] adapted to sheets. In this model, the measured total deformation gradient $\mathbf{F}$ of a triangle is multiplied with a per-triangle plasticity matrix, yielding the elastic deformation gradient $\mathbf{F}_{el} = \mathbf{F} \cdot \mathbf{F}_p$ which is then used in the physics update computation. $\mathbf{F_p}$ is initially an identity matrix, and is updated in each timestep using the flow rule

$$\mathbf{F}_p \leftarrow \mathbf{F}_p \cdot \mathbf{V} \left(\det(\mathbf{\Sigma})^{-\frac{1}{3}} \mathbf{\Sigma} \right)^{\gamma} \mathbf{V}^T \tag{9}$$

where $\mathbf{U\Sigma V}^T$ is the singular value decomposition of $\mathbf{F}_{el}$. The exponent $\gamma = \operatorname{clamp}(\Delta t\, \mu \cdot (\sigma_M - \sigma_0), 0 \ldots 1)$ is defined by the flow rate μ, the von Mises effective stress $\sigma_M = \sqrt{3} ||\boldsymbol{\sigma} - \frac{1}{3}\operatorname{tr}(\boldsymbol{\sigma})\mathbf{I}||$ and the yield stress σ_0. To represent the thinning of plastically stretched material, the per-triangle thickness is updated in the same way using $h \leftarrow h \det(\mathbf{\Sigma})^{-\frac{1}{3}\gamma}$. This thickness is then used to derive the area toughness τ from the volume toughness, and similarly the bending stiffness; thinned parts of the mesh therefore bend and break more easily.

Bending plasticity is modeled as in Narain et al. [2013]. Again, the bending plasticity is encoded as a 3×3 tensor, with the update rule

$$\mathbf{S}_p \leftarrow \mathbf{S}_p + \frac{\mathbf{S}_{el}}{||\mathbf{S}_{el}||} (||\mathbf{S}_{el}|| - \kappa_0) \tag{10}$$

for the elastic bending strain $\mathbf{S}_{el}$ from (3) and the yield curvature κ_0.

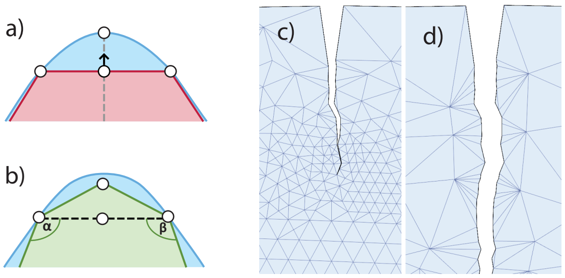


Figure 6: *A vertex is added during an edge-split remeshing operation. Its rest shape position is computed by projection onto a reference mesh shown in blue (a). Its plasticity is computed by minimizing the plastic bending and stretching energy of its surrounding elements (b). Adaptive remeshing allows to concentrate resolution around the crack tip (c), and coarsening the mesh immediately after the crack has formed (d).*

5 Remeshing

Adaptive remeshing is central to the efficiency and accuracy of our method. Through remeshing, we increase mesh resolution around the crack tip, maintain well-shaped elements as the fracture propagates, and coarsen the mesh in regions where stress is relieved. This allows complex crack patterns to be realistically produced without excessive computation elsewhere.

Unfortunately, modifying the discretization of curved shapes or objects always incurs some amount of discretization error, which will induce extraneous stresses in the material. In a non-fracturing system, subsequent timesteps would redistribute these isolated irregularities in the stress field. Artifacts such as popping would only occur if relatively large imbalances were introduced by remeshing. However, for a system that includes fracture modeling, artificial stresses can disturb the process of crack propagation and introduce spurious initiation sites. For stable results, therefore, it is essential to minimize artificial stresses caused by remeshing. This section presents techniques to address this issue.

5.1 Sizing Prediction

Our method uses anisotropic remeshing using a sizing metric [Narain et al. 2012; Narain et al. 2013] as a basis. This metric determines the desired length of an edge according to $\mathbf{e}^T\mathbf{M}\mathbf{e} \leq 1$ with the sizing tensor $\mathbf{M}$ and the edge vector in rest pose $\mathbf{e}$. The sizing metric encodes the resolution requirements of different aspects of the dynamic simulation, such as world-space curvature, distance to obstacles, and buckling. We include two additional criteria in this metric which are important to fracture simulation.

First, as we allow curved rest poses, we add a rest pose curvature term $\mathbf{M}_{rc} = \mathbf{S}_{rp}/\Delta n^2$ to prevent locking into flat modes. $\mathbf{S}_{rp}$ denotes the rest pose bending strain, and Δn the angular bound also used for the world-space curvature field. Second, a fracture sizing $\mathbf{M}_{\text{frac}} = (2\lambda/(\tau\, x_{\min}))^2\, \mathbf{I}$ is introduced. λ denotes the largest eigenvalue of σ^+ as in §4.1 and $x_{\min}$ the user specified minimum edge length for the remesher. This term ensures that the neighborhood of candidate fracture nodes is resolved with the highest resolution. We use an isotropic term here to provide an even distribution for stress evaluation. We also find that the refinement near the crack tip facilitates the collapse of any sliver triangles created by splitting during fracture propagation.

As the sizing metric is constructed in undeformed reference space,

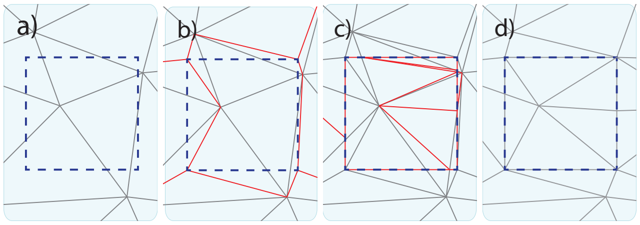


Figure 7: *A target shape (dotted blue square) is meshed into an existing geometry (a). First, triangles are split at the corner points (b). New edges are shown in red. Next, all edges intersecting the target shape are split (c). Finally, remeshing any removes ill-shaped triangles (d).*

regions which have been strongly stretched due to plasticity will show as severely deformed triangles in world space. To avoid these distortions, we measure the sizing tensor relative to the plastically stretched shape of each triangle. The deformation from the reference shape to the local plastic rest shape is characterized by $\mathbf{F}_p$. Therefore, we transform the combined sizing field using $\mathbf{M}' = \mathbf{F}_p^{-T}\mathbf{M}\mathbf{F}_p^{-1}$ and use $\mathbf{M}'$ instead of $\mathbf{M}$ in the remeshing criterion.

5.2 Remeshing Operators

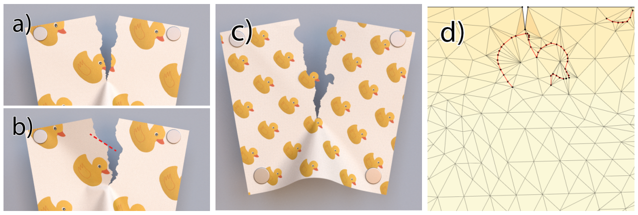
Once the sizing fields are known, the actual remeshing is performed using edge collapses, edge flips and edge splits. When adding a vertex during split, or moving a vertex during collapse, most algorithms interpolate coordinates and plasticity values. This interpolation will lead to spurious strains, especially for stiff materials and curved rest shapes.

One option is to apply a global optimization step [Narain et al. 2013] after remeshing, to reposition vertices in a way that minimizes stress. As reprojection is necessary after each opened crack to prevent distortions of the crack paths, global optimization is not feasible for fracture simulations. Instead, we propose a local reprojection formulation, which can be executed after each remeshing operation or topology change, and also integrates with plasticity remeshing. In contrast to the global optimization, it is automatically regularized by the surrounding fixed geometry and less prone to jittering.

Rest space position We allow the coarsening and refinement of curved rest shapes without an explicit parameterization, to enable the adaptive simulation of arbitrary shapes. If new vertices created by edge-split operations are placed directly on the split edges, the accumulation of resampling error will cause successive coarsening and refinement to diffuse and eventually eliminate shape features in curved regions. Instead, we keep a high-resolution version of the rest shape mesh, and ensure the rest shape positions of all vertices always lie on this reference surface.

When splitting an edge, we can reproject its rest space positition $\mathbf{u}$ onto the reference mesh by intersecting it with the ray $\mathbf{u} + s \cdot \hat{\mathbf{n}}$, where $\hat{\mathbf{n}}$ is the averaged face normal of the edge's adjacent triangles. Both $s > 0$ and $s < 0$ need to be tested to account for concave regions (Fig. 6a).

Plasticity In each remeshing step, the plasticity tensors need to be redistributed in a way that avoids loss of plasticity through diffusion. Previous work [Wicke et al. 2010a; Narain et al. 2013] does so by representing plastic deformation geometrically through a global "plastic embedding" that approximates the deformed rest shape. We instead construct a *local* plastic embedding in the neighborhood of a remeshing operation.



Images copyright Tobias Pfaff, Rahul Narain, Juan Miguel de Joya, James F. O'Brien.

Figure 8: *The natural crack path in a paper tearing simulation (a) is guided around the printed ducks using a localized defect line, shown in red (b). In (c), a weakening path around the ducks' silhouettes creates tear figures during the tearing. The defect lines are dynamically meshed in only when needed for breaking (d).*

We wish to choose new plasticity tensors which are consistent with the underlying rest geometry, and minimize change in the plastic edge length l_i^p and plastic angles θ_i^p at neighboring edges. These quantities can be recovered from the plasticity tensors stored on each triangle,

$$\mathbf{S}_p = \frac{1}{2A}\sum_{i=1}^{3}(\theta_i^p - \theta_i^R)l_i^p\mathbf{t}_i\mathbf{t}_i^T, \quad \mathbf{F}_p = \sum_{i=1}^{3}\frac{||\mathbf{e}_i||}{l_i}\frac{\mathbf{e}_i\,\mathbf{e}_i^T}{||\mathbf{e}_i||^2}. \tag{11}$$

For edges and faces adjacent to the edge to be remeshed, we construct a configuration in a local embedding space such that edge lengths and bending angles are equivalent to the plastic lengths and angles. This compliant plastic surface is the minimum energy state of the plastic energy in the neighborhood, given by evaluating (4) but with plastic angles θ^p and lengths l^p instead of world-space angles and positions:

$$E_p = \sum_{i}^{\text{adj. faces}}\frac{A_i}{2}\mathbf{G}_i^p : \boldsymbol{\sigma}_i^p + \sum_{i}^{\text{adj. edges}} k_B\frac{l_i^2}{8\,A_i}(\theta_i^p - \theta_i^R)^2. \tag{12}$$

Applying the remeshing operation to this embedding gives the new plastic lengths and edges that locally preserve the shape. On an edge split, we find an optimal embedding position of the new vertex which minimizes (12), with all surrounding vertices remaining fixed. This optimization has only three degrees of freedom and a well-shaped energy landscape, and quickly converges in a few Gauss-Newton steps. After the remeshing, the plasticity tensors can be reconstructed from the embedding using (11).

World-space Position Once the plasticity parameters have been found, the world space position of the changed vertex needs to be adjusted. Similar consideration as in the previous paragraph apply; again, we solve an energy minimization problem to find a world space position which avoids the generation of artificial stresses. The energy equation

$$E_s = \sum_{i}^{\text{adj. faces}}\frac{A_i}{2}\mathbf{G}_i : \boldsymbol{\sigma}_i + \sum_{i}^{\text{adj. edges}} k_B\frac{l_i^2}{8\,A_i}(\theta_i - \theta_i^p)^2 + m\,\mathbf{a}\cdot\mathbf{x} \tag{13}$$

is equivalent to (12), with the difference that we are now optimizing for world-space positions, and measure relative to plastic rest pose, in accordance with the physics update.

Minimizing the energy for the world-space position results in a quasi-static equilibrium state. To prevent the loss of momentum, we include the the additional term $m\mathbf{a} \cdot \mathbf{x}$, with the vertex mass m and the acceleration $\mathbf{a}$ from the last physics step interpolated at the new vertex. This term tries to preserves the local acceleration in a similar way to Narain et al.'s [2013] global reprojection. For the world space, no embedding needs to be constructed—the minimization

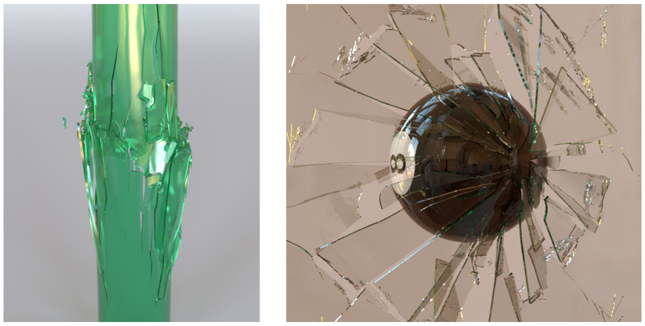


Figure 9: *The image on the left shows a polycarbonate cylinder fractured due to buckling under compression. The right image shows a high-resolution simulation of a glass window being broken by an impacting sphere.*

can be run on the world space mesh, again with the changed vertex position as the only degree of freedom.

6 Directible Fracture

Our method is designed to compute fracture dynamics as physically accurate as possible. However, an animator often may require a specific fracture behavior, such as cracks running in a particular direction or avoiding certain regions. Adaptive meshing facilitates the implementation of artistic direction in our framework.

Anisotropy An easy way to adjust crack paths is anisotropy. To make cracks in direction $\mathbf{a}$ more or less likely, the traction vectors $\mathbf{q}$ in (6) can be multiplied by a weakening matrix $\mathbf{I} + \beta\mathbf{a}\mathbf{a}^T$ with a parameter $\beta < 0$ (reinforcement) or $\beta > 0$ (weakening). The anisotropy tensor can be global, or combined with the defect matrix below. For example, this type of anisotropy is demonstrated by the leftmost image in Fig. 1.

Defect matrix Another way to influence crack behavior is to locally modulate the material's toughness. The physical equivalent are material defects, which often dominate crack initiation and propagation in real materials. Modulation of toughness is easy to achieve in an averaged sense by texturing the material toughness. However, without a very high mesh resolution this approach will not have the desired effect, as small features vanish within big triangles, and even for strong weakening defects, the crack will only roughly follow the desired pattern.

As our algorithm is designed for stable aggressive remeshing, we can provide conforming remeshing for these regions when necessary. We model defect lines or polygons as a vector texture mapped on the sheet, and refine mesh triangles to conform to these defects when they are close to fracture. Recalling from §4.1 that a triangle can fracture only if $2\lambda > \tau$, we consider a triangle to be close to fracture if it satisfies $2\lambda > \delta\tau$, with parameter $\delta < 1$ (we use $\delta = 0.1$ in our examples). If such a triangle contains an endpoint of a defect line, we split the triangle such that the endpoint lies on a new vertex. We perform the same test with lines, which cross defect lines, and split them at the intersection point. This process is visualized in Fig. 7. As the figure shows, the splits create many ill-shaped triangles. Therefore, the newly created lines which correspond to defect lines are now marked to be preserved by the remesher, and a global remeshing step is executed to remove all sliver triangles around the shapes.

If $2\lambda/\tau$ falls below $\frac{1}{2}\delta$, defect lines are remeshed out again to reduce the triangle count. Using this technique, the defect lines or areas are directly tested when computing separation strength, and will produce the desired breaking pattern (Fig. 8).

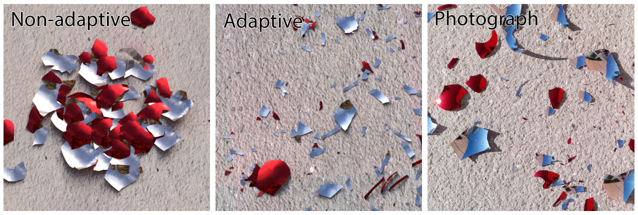


Figure 10: *The shattering of a spherical glass ornament is simulated using our adaptive algorithm (middle), and a statically meshed version with the same triangle count (left). The fracture pattern of the adaptive simulation is similar to an actual experiment (right), while the simulation without remeshing fails to correctly resolve the fracture dynamics and produces an unrealistic distribution of shards.*

7 Results and Discussion

We implemented and tested the described fracture algorithm in a variety of scenarios, which will be discussed in this section. A reference implementation of our method has been incorporated into the ArcSim software package. Performance numbers for our examples are summarized in Table 1. The supplemental material for this paper includes a video with several example animations.

Fracture dynamics To demonstrate the space of fracture patterns we can achieve using our method, we simulated the impact of a rigid sphere on several sheets of varied materials with different mechanical properties. Five of the obtained fracture patterns are depicted in Fig. 1, and more can be seen in the accompanying video. The differing behaviors can be attributed to different plasticity and fracture mechanics. For example, the first example in Fig. 1 is driven by bending fracture, while stretching fracture dominates in the second example. The third image shows a purely elastic material, while the fourth and fifth images are dominated by bending and stretching plasticity, respectively. We can also model fracture due to buckling. In Fig. 9 (left), a polycarbonate plastic cylinder is compressed, and then fractures due to the bending stresses induced by buckling. To obtain these varied dynamics, a full treatment of bending and stretching plasticity and fracture, as presented in our method, is necessary.

In Fig. 10, we simulated a glass ball ornament dropping on the floor. Our adaptive simulation produces shards with a variety of shapes and sizes, similar to the ones observed in a real-world experiment. Many of these shapes are unlike Voronoi regions, and would not be observed in a Voronoi-based fracturing algorithm. We also compare our adaptive simulation to a statically meshed simulation with the same average number of triangles. Not only does the statically meshed simulation yield a smaller number of bigger and more uniform shards, the locally coarser resolution in fracture regions also impacts the fracturing dynamics. While the adaptive simulation shows wide spreading of the shards due to shock-wave propagation in the material, these dynamics are not observed in the statically meshed sphere.

We compared our fracture criterion to the separation tensor of O'Brien and Hodgins [1999]. While their method works well when meshes are largely homogeneous in terms of mesh size, it behaves poorly in our adaptive context. In Fig. 5 we show a tearing simulation of a thin sheet. If we replace our fracture criterion with their separation tensor, the small triangles directly at the crack tip will contribute a smaller force than the surrounding larger triangles.

Scene	# faces	dynamics	geometry	total
Metal and glass Fig. 2	5.72k	28.6	25.8	54.4
Sphere drop Fig. 10	4.52k	23.2	22.5	45.7
Dome Fig. 11	17.12k	34.2	53.5	87.7
Paper tearing Fig. 8	3.01k	16.7	21.8	38.5
Rubber tearing Fig. 3	2.69k	11.3	12.7	24.0
Buckling Fig. 9	3.8k	30.3	44.7	75.0
Material variety Fig. 1 (average value)	4.22k	20.6	43.7	64.3

Table 1: *Performance measurements for our examples. Timing numbers are in seconds per frame (at 25 fps).* Dynamics *includes physics update, collision handling and plasticity, while* geometry *refers to remeshing and breaking. All simulations were run on a workstation with an Intel Xeon X5675 CPU.*

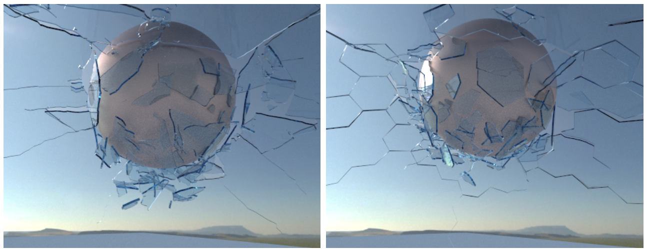


Figure 11: *A glass dome is hit by a fast-moving sphere. The left image shows the natural crack paths, while the right image uses a hexagonal defect line texure to mimic the shattering of a geodesic dome.*

This condition effectively flattens out the separation strength and leads to jagged crack geometry. If fracture substepping, and local re-projection are disabled, then stress will not be relieved in the triangles surrounding the main crack path as it forms. This situation leads to artificial crack junctions and spurious small cracks around the main crack.

Plasticity Plasticity has a strong impact on breaking behavior. One interesting effect is that due to the stress singularity around the crack tip, crack edges are mores strongly subject to stretching plasticity than the rest of the material. When tearing plastic foil or rubber sheets, this plasticity commonly manifests as undulating tear edges. We performed a tearing simulation of a thin rubber sheet, and compared it to a real-world experiment with similar results (Fig. 3).

In Fig. 2 (left), a multi-material simulation consisting of an aluminum box frame and glass panes is depicted. The plastic deformation of the aluminum frame induces shear strains on the glass, which breaks due to shear fracture. If we were to interpolate plasticity tensors, this example would suffer from a loss of plasticity in the frame. That is, the shape would slowly degrade towards its initial position. Simple plastic embedding, on the other hand, would try to represent plasticity as a global plastic rest shape. That approach fails in the given example, as it induces artificial plasticity in the glass panes (Fig. 2 right). Similar issues occur for the metal examples in Fig. 1.

Directibility The specific crack path of a fracturing simulation is hard to predict. In Fig. 8, we demonstrate how defect lines and textures can be used to influence crack paths. This example shows a simulated piece of paper being torn. The fracture path generated with no included defects has a natural look, but it also splits one of the printed ducks. To avoid tearing the duck, a small defect line is introduced to guide the crack around the duck's head, while maintaining a natural look. To make the paper behave as if the ducks were tear-out figures, the outline of each duck is used to specify defect geometry. As the tear progresses, the defect lines are dynamically meshed into high-stress regions, allowing the ducks to tear out cleanly while preserving a low triangle count. Areas away from the defect lines retain the natural appearance of tearing paper.

In Fig. 11, the fast impact of a sphere on a glass dome is simulated. If a hexagonal defect line texture is embedded in the dome, we observe that fracture on the large scale produces hexagonal shapes, while smaller shards are dominated by unrestricted fracture dynamics.

8 Conclusions

Realistic simulation of fracture is challenging to simulate. It requires high resolution around the crack tip to resolve the fracture dynamics for realistic crack path formation, and is prone to introducing ill-shaped geometry in the splitting process. We have shown that adaptive remeshing in conjunction with local reprojection can be used to accurately compute and resolve high-resolution crack paths, while coarsening passive regions of the mesh. We demonstrated that this adaptive evaluation together with plasticity and fracture modeling of both bending and stretching deformation is an effective combination to simulate a wide range of materials with very different fracture behaviors. In addition, we explored ways to leverage this adaptivity to dynamically remesh defect pattern when needed.

Our use of substepping and relaxation generally allows timesteps larger than would normally be expected given a material's crack speed and the minimum refined edge size. However, even with these procedures, some materials will still require small timestep sizes. This limitations arises because the node displacement over a large timestep will not be tolerable for very stiff (incompliant) materials with relatively low toughness. Our relaxation and substepping scheme succeeds in reproducing the behavior of glass-like brittle materials such as acrylic (Fig. 1 middle, Fig. 11) at reasonable timesteps. However, the distinct fracture pattern of plate glass appears because even extremely small deformations result in brittle fracture. As shown in Fig. 9 (right), we can capture this behavior. Doing so however requires extremely small timesteps of up to 2×10^{-6} s, leading to a total simulation time of 20 hours, an order of magnitude larger than our other examples. It would be interesting to investigate how to adapt relaxation to allow such simulations at larger timesteps.

Ductile fracture is a very complex process which is hard to model accurately. While the maximum stress criterion is often a good approximation for elastic fracture, and works nicely for many ductile fracture problems when combined with the thinning described in §4.2, it would be interesting to combine our algorithm with a failure model for plastic damage due to microvoid formation. However, these models are often hard to parametrize and are tailored towards specific materials and loading scenarios. An investigation of data-driven methods to capture plastic and fracture behavior, as has been done for elastic parameters of cloth [Wang et al. 2011], would be an interesting avenue for future work.

Acknowledgments

We thank the other members of the Berkeley Visual Computing Group for their help and support, and the anonymous reviewers for their insightful comments. This work was supported by funding from the Intel Science and Technology Center for Visual Computing, NSF Grant IIS-0915462, and a gift from Pixar Animation Studios.

Our method was implemented using the UC Berkeley ARCSim framework. The images in this paper and accompanying video were

rendered using the Mitsuba rendering software by Wenzel Jakob.[2]

References

ANDO, R., THÜREY, N., AND WOJTAN, C. 2013. Highly adaptive liquid simulations on tetrahedral meshes. *ACM Trans. Graph. (Proc. SIGGRAPH 2013)* (July).

BAO, Z., HONG, J.-M., TERAN, J., AND FEDKIW, R. 2007. Fracturing rigid materials. *IEEE transactions on visualization and computer graphics 13*, 2 (Jan.), 370–8.

BARGTEIL, A. W., WOJTAN, C., HODGINS, J. K., AND TURK, G. 2007. A finite element method for animating large viscoplastic flow. *ACM Transactions on Graphics 26*, 3 (July), 16.

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

BRIDSON, R., MARINO, S., AND FEDKIW, R. 2003. Simulation of clothing with folds and wrinkles. In *Proc. 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '03, 28–36.

BUSARYEV, O., DEY, T. K., AND WANG, H. 2013. Adaptive fracture simulation of multi-layered thin plates. *ACM Transactions on Graphics 32*, 4 (July), 1.

CHENTANEZ, N., FELDMAN, B. E., LABELLE, F., O'BRIEN, J. F., AND SHEWCHUK, J. R. 2007. Liquid simulation on lattice-based tetrahedral meshes. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2007*, 219–228.

CLAUSEN, P., WICKE, M., SHEWCHUK, J. R., AND O'BRIEN, J. F. 2013. Simulating liquids and solid-liquid interactions with lagrangian meshes. *ACM Transactions on Graphics 32*, 2 (Apr.), 1–15.

GINGOLD, Y., SECORD, A., HAN, J. Y., GRINSPUN, E., AND ZORIN, D. 2004. A Discrete Model for Inelastic Deformation of Thin Shells. In *ACM Symposium on Computer Animation*.

GLONDU, L., MUGUERCIA, L., MARCHAL, M., BOSCH, C., RUSHMEIER, H., DUMONT, G., AND DRETTAKIS, G. 2012. Example-Based Fractured Appearance. *Computer Graphics Forum 31*, 4 (June), 1547–1556.

GRINSPUN, E., HIRANI, A. N., DESBRUN, M., AND SCHRÖDER, P. 2003. Discrete shells. In *Proc. 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '03, 62–67.

HARMON, D., VOUGA, E., TAMSTORF, R., AND GRINSPUN, E. 2008. Robust treatment of simultaneous collisions. *ACM Trans. Graph. 27*, 3 (Aug.), 23:1–23:4.

HUTCHINSON, D., PRESTON, M., AND HEWITT, T. 1996. Adaptive refinement for mass/spring simulations. In *7th Eurographics Workshop on Animation and Simulation*, Springer-Verlag, 31–45.

IBEN, H. N., AND O'BRIEN, J. F. 2006. Generating surface crack patterns. *ACM Symposium on Computer Animation* (Sept.), 177–185.

KAUFMANN, P., MARTIN, S., BOTSCH, M., GRINSPUN, E., AND GROSS, M. 2009. Enrichment textures for detailed cutting of shells. *ACM Transactions on Graphics 28*, 3 (July), 1.

KLINGNER, B. M., FELDMAN, B. E., CHENTANEZ, N., AND O'BRIEN, J. F. 2006. Fluid animation with dynamic meshes. *ACM Trans. Graph. 25*, 3 (July), 820–825.

MOLINO, N., BAO, Z., AND FEDKIW, R. 2004. A virtual node algorithm for changing mesh topology during simulation. *ACM Transactions on Graphics*, 23 (July), 385—-392.

MÜLLER, M., CHENTANEZ, N., AND KIM, T.-Y. 2013. Real time dynamic fracture with volumetric approximate convex decompositions. *ACM Transactions on Graphics 32*, 4 (July), 1.

NARAIN, R., SAMII, A., AND O'BRIEN, J. F. 2012. Adaptive anisotropic remeshing for cloth simulation. *ACM Transactions on Graphics 31*, 6 (Nov.), 1.

NARAIN, R., PFAFF, T., AND O'BRIEN, J. F. 2013. Folding and crumpling adaptive sheets. *ACM Trans. Graph. 32*, 4, 51:1–51:8.

NORTON, A., TURK, G., BACON, B., GERTH, J., AND SWEENEY, P. 1991. Animation of fracture by physical modeling. *The Visual Computer 7*, 4, 210–219.

O'BRIEN, J. F., AND HODGINS, J. K. 1999. Graphical modeling and animation of brittle fracture. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques - SIGGRAPH '99*, ACM Press, New York, New York, USA, 137–146.

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

O'BRIEN, J. F. 2000. Graphical Modeling and Animation of Fracture.

PAULY, M., KEISER, R., ADAMS, B., DUTRÉ, P., GROSS, M., AND GUIBAS, L. J. 2005. Meshless Animation of Fracturing Solids. *ACM transactions on graphics 24*, 3 (July), 957–964.

SIFAKIS, E., DER, K. G., AND FEDKIW, R. 2007. Arbitrary cutting of deformable tetrahedralized objects. In *ACM Symposium on Computer Animation*, Eurographics Association, 73–80.

SIMNETT, T. J. R., LAYCOCK, S. D., AND DAY, A. M. 2009. An Edge-based Approach to Adaptively Refining a Mesh for Cloth Deformation. In *Eurographics UK Theory and Practice of Computer Graphics*, 77–84.

SU, J., SCHROEDER, C., AND FEDKIW, R. 2009. Energy stability and fracture for frame rate rigid body simulations. In *ACM Symposium on Computer Animation*, ACM Press, New York, New York, USA, 155.

TERZOPOULOS, D., AND FLEISCHER, K. 1988. Modeling inelastic deformation: viscolelasticity, plasticity, fracture. In *Proc. SIGGRAPH '88*, 269–278.

WANG, H., O'BRIEN, J. F., AND RAMAMOORTHI, R. 2011. Data-driven elastic models for cloth. *ACM Transactions on Graphics 30*, 4 (July), 1.

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.

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 Transactions on Graphics 29*, 4 (July), 1–11.

WOJTAN, C., AND TURK, G. 2008. Fast viscoelastic behavior with thin features. *ACM Trans. Graph. 27*, 3 (Aug.), 47:1–47:8.

[2] http://www.mitsuba-renderer.org