

UC Berkeley

UC Berkeley Previously Published Works

Title

Adaptive anisotropic remeshing for cloth simulation

Permalink

<https://escholarship.org/uc/item/5s1775xd>

Journal

ACM Transactions on Graphics, 31(6)

ISSN

0730-0301

Authors

Narain, Rahul
Samii, Armin
O'Brien, James F

Publication Date

2012-11-01

DOI

10.1145/2366145.2366171

Supplemental Material

<https://escholarship.org/uc/item/5s1775xd#supplemental>

Peer reviewed

Adaptive Anisotropic Remeshing for Cloth Simulation

Rahul Narain

Armin Samii

James F. O'Brien

University of California, Berkeley



Images copyright Rahul Narain, Armin Samii, and James F. O'Brien.

Figure 1: A motion-captured character performs a jumping kick. His clothing is dynamically remeshed to capture detail such as wrinkles, while having larger elements in smooth areas. Here and elsewhere in the paper, large elements are shown in blue, small equilateral elements in red, and anisotropic elements in yellow.

Abstract

We present a technique for cloth simulation that dynamically refines and coarsens triangle meshes so that they automatically conform to the geometric and dynamic detail of the simulated cloth. Our technique produces anisotropic meshes that adapt to surface curvature and velocity gradients, allowing efficient modeling of wrinkles and waves. By anticipating buckling and wrinkle formation, our technique preserves fine-scale dynamic behavior. Our algorithm for adaptive anisotropic remeshing is simple to implement, takes up only a small fraction of the total simulation time, and provides substantial computational speedup without compromising the fidelity of the simulation. We also introduce a novel technique for strain limiting by posing it as a nonlinear optimization problem. This formulation works for arbitrary non-uniform and anisotropic meshes, and converges more rapidly than existing solvers based on Jacobi or Gauss-Seidel iterations.

Keywords: Cloth simulation, dynamic remeshing, anisotropic remeshing, strain limiting, augmented Lagrangian method.

Links: [DL](#) [PDF](#) [VIDEO](#) [WEB](#)

1 Introduction

Cloth in real life simultaneously exhibits both highly detailed wrinkles and folds, and flat or smoothly curving regions. As the cloth moves, the fine wrinkles appear and disappear at different locations.

Contact email: {narain, samii, job}@eecs.berkeley.edu

From the conference proceedings of ACM SIGGRAPH Asia 2012.
Appearing in ACM Transaction on Graphics Vol. 31, No. 6.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACM SIGGRAPH Asia, 2012, Singapore

© Copyright ACM 2012

Wrinkles may also migrate across the surface or travel in a wave-like fashion. With a fixed simulation mesh, adequately resolving these details requires a large number of extremely small elements over the entire mesh. The resulting computation and memory costs for uniformly high-resolution simulations can be burdensome, and it is instead desirable to focus simulation resolution in regions that exhibit complex shape and motion, while representing flat regions with coarser elements.

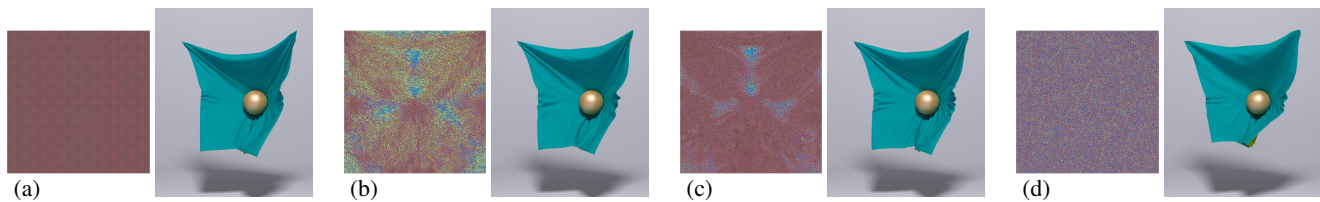
In this paper, we present a technique to dynamically refine and coarsen a finite element mesh used for cloth simulation. Our scheme creates anisotropic elements that follow the curvature of wrinkles and creases in the cloth. It preserves dynamic behaviors by maintaining resolution conforming to high velocity field gradients. It also anticipates the buckling of the material by refining the mesh where it is beginning to become compressed.

The resulting simulation method efficiently produces results that are visually equivalent to those produced with more costly uniformly high-resolution meshes. By adaptively refining and coarsening the mesh, elements are concentrated in detail regions. The anisotropic nature of our remeshing algorithm means that refining near clusters of long parallel wrinkles, which occur commonly in cloth, will require roughly linear instead of quadratic growth in the number of elements. Anisotropic remeshing also tends to align mesh edges with wrinkles so that visually they appear smoother. These features can be seen in Figures 1 and 2. Our adaptive, anisotropic remeshing procedure is computationally inexpensive and simple to implement.

We also introduce a fast technique for performing strain limiting on large meshes. Existing methods based on nonlinear Jacobi or Gauss-Seidel iterations converge slowly for large meshes. By casting strain limiting as a constrained optimization problem, we obtain faster convergence using the augmented Lagrangian method along with nonlinear conjugate gradients.

2 Related Work

Cloth simulation. Cloth simulation has been a major topic of computer graphics research for over two decades. Many successful techniques have been proposed for modeling the dynamics of cloth and for handling collisions [Baraff and Witkin 1998; Bridson et al. 2002; Choi and Ko 2002; Bridson et al. 2003]. Summaries



Images copyright Rahul Narain, Armin Samii, and James F. O'Brien.

Figure 2: A square sheet of cloth interacting with a moving sphere is simulated with, from left to right, (a) a fixed high-resolution mesh with 65.5k faces, (b) adaptive anisotropic remeshing with average 11.3k faces, (c) adaptive isotropic remeshing with average 30.3k faces, (d) a fixed mesh with 11.3k faces. The results of (a), (b), and (c) have comparable wrinkles, while (d) has fewer and exhibits stiffer looking motion. As shown in Table 1, the cost of simulation with adaptive anisotropic remeshing is significantly less than the comparable uniformly or isotropically meshed simulations.

covering much of the work done in graphics for modeling cloth and deformable bodies can be found in the recent survey articles of Nealen et al. [2006] and Thomaszewski et al. [2007].

In this paper, our goal is to represent the fine details of wrinkles and folds in cloth directly in the simulation mesh through adaptive refinement. An alternative approach would be to simulate on either a coarser or reduced model, and then add visual detail during a postprocessing step. A number of recent cloth simulation techniques fall into this category. Strategies for determining the additional detail include learning from high-resolution simulation data [de Aguiar et al. 2010; Feng et al. 2010; Wang et al. 2010a; Kavan et al. 2011], sampling from recordings of real cloth [Popa et al. 2009; Hilsmann and Eisert 2012], and directly applying a simplified physical model [Müller and Chentanez 2010; Rohmer et al. 2010]. We have observed that the dynamics of fine details can be visually significant and we believe there is utility in simulating fine details directly rather than approximating them. However, we also believe that our approach is complementary to these detail enhancement methods. For example, our adaptive simulation could replace the uniform base simulation in the method of Müller and Chentanez [2010], allowing a controlled trade-off between realistic dynamics and computational speed based on the scale at which a transition from our dynamic wrinkles to their quasi-static wrinkles occurs.

We have observed that the constitutive model used for cloth simulation can have a significant impact on the overall appearance and perceived realism of the resulting motion. Recent work has focused on building realistic constitutive models from measurements of real cloth materials. Techniques for measuring this data have been described by Volino et al. [2009], Wang et al. [2011], and Miguel et al. [2012]. In our examples, we have made use of the constitutive model proposed by Wang et al. [2011] with parameters derived from their published measurements of various cloth materials.¹ Other related work includes capturing of volumetric elastic materials [Bickel et al. 2009] and example-based simulation of elastic materials [Martin et al. 2011].

Remeshing. Recently, there has been substantial interest in dynamic remeshing of simulation meshes, especially for simulation of plastic flow and fluid dynamics. Some previous work has used Eulerian tetrahedral meshes for fluid simulation which require remeshing to account for moving boundaries such as obstacles [Klingner et al. 2006] and free surfaces [Chentanez et al. 2007]. For plastic flow, Bargeil et al. [2007] and Wojtan and Turk [2008] used a Lagrangian finite element mesh whose vertices moved with the flow, and performed global remeshing whenever elements became poorly conditioned. Wicke et al. [2010] extended the tetrahedral mesh optimization framework of Klingner and Shewchuk [2007] to

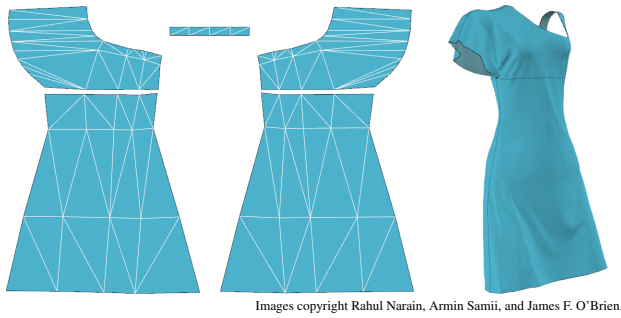
do dynamic anisotropic remeshing for Lagrangian simulations with elasticity and plastic flow. For detailed simulation of elastic bodies, Grinspun et al. [2002] advocated refining the finite element basis functions rather than the mesh elements themselves.

A number of dynamic remeshing techniques have also been proposed for cloth simulation. Early work used mass-spring systems in a square grid refined in a quadtree fashion [Hutchinson et al. 1996; Villard and Borouchaki 2002], or a hierarchical representation of a multiresolution triangle mesh [Li and Volkov 2005]. Those methods used curvature alone as the refinement criterion, which tends to suppress the formation of wrinkles as it is more difficult for an underresolved mesh to buckle. Simnett et al. [2009] used an edge-based approach for refining a triangle mesh, including compression and collisions as additional refinement criteria. All of these techniques use a mesh hierarchy to maintain regularity of element shapes and simplify the coarsening operation; however, this precludes the creation of anisotropic elements, which we find are necessary to efficiently represent the folds and wrinkles of real cloth.

Because we represent cloth geometry as a two-dimensional triangle mesh, our work is also closely related to previous work on remeshing of static surfaces. In general, the body of literature focusing on generating, refining, and optimizing triangle meshes is extensive; a comprehensive review can be found in the survey by Alliez et al. [2007]. Particularly close to our method, techniques for anisotropic meshing in two dimensions have been studied in the computational geometry literature [Bossen and Heckbert 1996; Labelle and Shewchuk 2003]. In computer graphics, efficient techniques have been proposed for simplifying meshes [Hoppe 1996; Garland and Heckbert 1997], while refinement can be performed using subdivision rules such as those of Catmull and Clark [1978] and Loop [1987]. Although splitting, flipping, and collapsing edges have been well established tools for mesh refinement and simplification (for example [Popović and Hoppe 1997; Garland and Heckbert 1998; Jiao et al. 2006]), there is little discussion about using dynamical metrics to achieve a desired spatial and directional distribution of mesh resolution.

Strain limiting. Provot [1995] introduced strain limiting as a technique for stably modeling stiff springs by imposing constraints on the maximum and minimum allowed strain of each link. Subsequently, many extensions of this technique have been developed by different authors. Goldenthal et al. [2007] proposed an efficient constrained Lagrangian method for modeling inextensible spring networks. For triangle meshes, English and Bridson [2008] used non-conforming elements to model inextensible cloth. Thomaszewski et al. [2009] presented a continuum-based technique that independently constrained the three components of the strain tensor. A technique for isotropic strain limiting was proposed by Wang et al. [2010b], who also introduced a multiresolution approach for enforcing these constraints. We found that these existing techniques did not perform

¹<http://graphics.berkeley.edu/papers/Wang-DDE-2011-08>



Images copyright Rahul Narain, Armin Samii, and James F. O'Brien.

Figure 3: A dress is modeled using five flat panels stitched together. We retain the flat shape of the panels in the form of two-dimensional material space coordinates stored for the vertices of each face. Initial meshes may be quite coarse as adaptive meshing will automatically refine them as needed.

well when applied to our highly nonuniform, anisotropic meshes, and in this paper we also propose the use of a numerical method with better convergence properties to enforce strain limits.

3 Simulation Overview

We represent the cloth as a finite element mesh composed of one or more triangulated panels, as shown in Figure 3.² We use a planar reference embedding for the panels where each vertex i is assigned two-dimensional material space coordinates \mathbf{u}_i . Some of the panel boundary edges correspond to seams or darts. The vertices along these edges are associated with corresponding vertices on the boundary edges that are joined by the seam.

Each vertex also has a three-dimensional world-space position \mathbf{x}_i , and velocity \mathbf{v}_i . However, not all vertices have distinct positions and velocities. The vertices associated with a single location on a seam will share a single world-space position and velocity. This representation explicitly fuses separate panel boundaries together along seams by mapping all of a vertex's material locations to a single location in world space. We use lumped masses so that each node has a scalar mass m_i that is set to one third of the sum of material-space areas of its incident faces, multiplied by the material's area-density. Vertex normals \mathbf{n}_i are computed using a weighted average of face normals [Max 1999].

For the constitutive model, we use the piecewise linear stretching and bending model described by Wang et al. [2011] with parameters derived from their measured data. In-plane stretching is treated implicitly using a corotational finite element approach [Müller and Gross 2004], while bending forces are added using a discrete hinge model [Bridson et al. 2003; Grinspun et al. 2003]. The linear system for implicit time integration is solved using the sparse Cholesky-based solver in the TAUCS library.³ We detect collisions using a bounding volume hierarchy [Tang et al. 2010] and compute collision response using non-rigid impact zones [Harmon et al. 2008]. Our remeshing and strain limiting techniques are independent of these choices, and would also work with other triangular finite element cloth simulation methods.

Our simulation loop is essentially the same as the traditional cloth simulation loop, with the addition of our dynamic remeshing step that runs periodically. If continuous collision detection is employed,

²Cloth is typically manufactured as flat sheets and nearly all manufactured garments are assembled from multiple cutout panels that are joined together at seams. As with prior clothing simulation methods (e.g. [Carignan et al. 1992; Decaudin et al. 2006; Umetani et al. 2011]) we model three-dimensional garments as assemblies of flat panels.

³<http://www.tau.ac.il/~stoledo/taucs/>

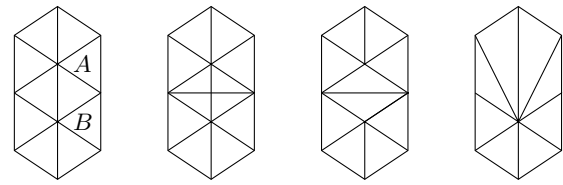


Figure 4: The three basic remeshing operations applied to the mesh at far left. From left to right, the results after the edge AB is: split; flipped; and collapsed in the direction $A \rightarrow B$.

remeshing must be performed between collision detection intervals. The remeshing algorithm is described in the following section. We also introduce an optimization-based technique for strain limiting, which we describe in Section 5.

4 Dynamic Remeshing for Cloth

The goal of remeshing is to refine the simulation mesh enough to resolve visual and dynamical detail, while coarsening the mesh in areas where such detail is absent. In particular, we aim to capture fine wrinkles and folds using small, narrow triangles oriented along these features, while still having large triangles in the flat, smooth areas of the material.⁴

To achieve this goal, we phrase the dynamic remeshing algorithm simply as the task of finding the coarsest mesh whose edges are all short enough to resolve simulation detail. Formally, we define a *sizing field* as a tensor field \mathbf{M} over the embedding space that describes the maximum permitted length of edges at each location and orientation, and we perform local remeshing operations to make the mesh satisfy this sizing field. The use of a tensor field to control the size and shape of the mesh is an established strategy in adaptive meshing (for example, [Bossen and Heckbert 1996; Labelle and Shewchuk 2003; Wicke et al. 2010]).

4.1 The Remeshing Scheme

For now, assume that the sizing field is given as a tensor field \mathbf{M} defined at each of the vertices of the current mesh. Using this sizing field, we may define the *size* of an edge between vertices i and j as

$$s(i, j)^2 = \mathbf{u}_{ij}^T \left(\frac{\mathbf{M}_i + \mathbf{M}_j}{2} \right) \mathbf{u}_{ij}. \quad (1)$$

Here and below, we write \mathbf{u}_{ij} to denote $\mathbf{u}_i - \mathbf{u}_j$ for convenience. If the edge lies on a seam, we apply the above formula both to it and to its corresponding edge across the seam, and take the average. We judge an edge as *valid* if its size is at most 1. Our remeshing algorithm proceeds by refining the mesh until all edges are valid, and then coarsening it as far as possible without introducing any edges that are close to being invalid.

We use three basic remeshing operations: edge splits, edge flips, and edge collapses. These operations are illustrated in Figure 4. Edge splits and collapses apply simultaneously to both sides of a seam. Edges along seams and boundaries cannot be flipped.

An edge may be split unconditionally if its size exceeds 1. The material-space coordinates, the velocity, and the sizing field at the newly created vertex are interpolated halfway between the two endpoints of the edge. We set the new vertex's world-space coordinates

⁴Thin triangles with aspect ratios far from one are often blamed for instability and poor accuracy. However, when the underlying fields are anisotropic and the thin triangles in the discretization are accordingly oriented, they can be as accurate as and no less stable than higher-resolution isotropic elements. We refer the reader to the article by Shewchuk [2002] for further discussion.

by minimizing a generalization of the quadric error metric [Garland and Heckbert 1997], taking into account both surface curvature and strain within adjacent faces. Specifically, we choose its position \mathbf{x}_i to minimize

$$Q(\mathbf{x}_i) = \sum_{\substack{\text{vertex } j \\ \text{adjacent to } i}} \left(\frac{(\mathbf{x}_i - \mathbf{x}_j) \cdot \mathbf{n}_j}{\|\mathbf{u}_i - \mathbf{u}_j\|} \right)^2 + \sum_{\substack{\text{face } f \\ \text{adjacent to } i}} \|\boldsymbol{\varepsilon}_f\|_F^2, \quad (2)$$

where $\boldsymbol{\varepsilon}_f$ is the element's strain tensor and $\|\cdot\|_F$ denotes the Frobenius norm. We linearize $\boldsymbol{\varepsilon}_f$ with respect to \mathbf{x}_i about the world-space midpoint of the original edge, thus reducing Q to a quadratic function which can be minimized by a linear solve.

Edge flips serve the purpose of ensuring that triangles are well shaped with respect to the sizing field. We use the anisotropy-aware criterion of Bossen and Heckbert [1996] to determine whether to flip an edge. If an edge ij has adjacent vertices k and ℓ , then it should be flipped to form edge $k\ell$ if

$$(\mathbf{u}_{jk} \times \mathbf{u}_{ik}) \mathbf{u}_{i\ell}^T \mathbf{M}_{\text{avg}} \mathbf{u}_{j\ell} + \mathbf{u}_{jk}^T \mathbf{M}_{\text{avg}} \mathbf{u}_{ik} (\mathbf{u}_{i\ell} \times \mathbf{u}_{j\ell}) < 0, \quad (3)$$

where \mathbf{M}_{avg} is the average of the sizing fields at all four vertices, and $\mathbf{a} \times \mathbf{b} = a_x b_y - a_y b_x$ is the two-dimensional cross product. This test is equivalent to testing whether the edge is Delaunay in the stretched space of the sizing field [Labelle and Shewchuk 2003].

An edge collapse can be performed on a given edge in two ways, namely by removing either one of its vertices; when collapsing, we try both in an arbitrary order.⁵ Collapsing an edge is forbidden if

- it changes the panel boundaries in material space,
- it produces any face that is inverted, or whose aspect ratio is too small, or
- it creates any edge that is close to invalid.

By “close to invalid”, we mean that the sizes of the new edges must not exceed $1 - h$, where h is a small hysteresis parameter; this asymmetry prevents oscillations between splitting and collapsing over consecutive frames. We use $h = 0.2$ for all the results shown.

Given these local operations, all that remains to define the dynamic remeshing algorithm is the schedule of how they are applied to the input mesh. Our algorithm, shown in pseudocode in Algorithm 1, has two phases. In the first phase, we eliminate all invalid edges by repeatedly splitting edges whose size exceeds 1. In the second phase, once all edges are valid, we attempt to coarsen the mesh as far as possible without introducing invalid edges. Each phase terminates when no more changes to the mesh are possible. In each phase, we perform edge flips to maintain the quality of the mesh.

For efficiency, in the COLLAPSEEDGES and FLIPEDGES subroutines, we maintain a set of active faces which have been recently changed, and only consider the edges of those faces as candidates for remeshing. Faces created (or destroyed) by a remeshing operation are added to (resp. removed from) the active set. Also, in the SPLITEDGES and FLIPEDGES loops, we reduce the number of edge size evaluations by operating on many independent edges in one iteration. Two edges are considered independent if they share no vertices, and so can be split or flipped without modifying each other's adjacent faces. A maximal independent set of edges can be computed greedily in linear time: for each edge, we check if it is valid to split/flip and its vertices are not already selected, and if so, we add it to the set and mark its vertices as selected.

⁵We do not collapse an edge to its midpoint, as doing so would require averaging physical quantities such as world-space positions and velocities. We want to avoid such averaging as much as possible as it causes artificial damping and degrades the cloth's motion.

Algorithm 1 Our adaptive remeshing algorithm.

procedure ADAPTIVEREMESH(mesh M)
 SPLITEDGES(M)
 COLLAPSEEDGES(M)

procedure SPLITEDGES(mesh M)
repeat
 $E \leftarrow$ maximal independent set of splittable edges in M
 for all (i, j) in E **do**
 if edge (i, j) still exists in M **then**
 Split edge (i, j)
 Run FLIPEDGES on faces created by split
until $E = \emptyset$

procedure COLLAPSEEDGES(mesh M)
 $F \leftarrow$ set of faces of M
repeat
 Find collapsible edge (i, j) adjacent to a face in F ,
 removing faces with no collapsible edges
 Collapse edge (i, j)
 Update F with faces changed by collapse
 Run FLIPEDGES on faces changed by collapse
 Update F with faces changed by flips
until $F = \emptyset$

procedure FLIPEDGES(set of active faces F)
repeat
 $E' \leftarrow$ set of edges adjacent to faces in F
 $E \leftarrow$ maximal independent set of flippable edges in E'
 for all (i, j) in E **do**
 Flip edge (i, j)
 Update F with faces changed by flips
until $E = \emptyset$

4.2 The Sizing Field

We want the mesh to be finely resolved where the surface is curved, to capture geometric detail; where the material is compressed, to allow it to buckle; and where vertex velocities are rapidly varying, to retain dynamical detail. Therefore, we define the sizing field in terms of the curvature, compression, and velocities of the material. As these involve material-space derivatives and are most conveniently discretized on faces instead of vertices, we first compute the sizing field on faces, and then perform area-weighted averaging to interpolate the values onto vertices.⁶ One issue is how to define the sizing field at vertices that lie on seams; our solution is discussed at the end of this section.

Similar to Rusinkiewicz [2004], we can estimate the curvature over a face by using the differences in vertex normals. The change in normal between vertices i and j can be expressed as

$$\begin{aligned} \|\mathbf{n}_i - \mathbf{n}_j\|^2 &= \|\nabla \mathbf{n} \cdot \mathbf{u}_{ij}\|^2 \\ &= \mathbf{u}_{ij}^T (\nabla \mathbf{n}^T \nabla \mathbf{n}) \mathbf{u}_{ij}, \end{aligned} \quad (4)$$

where $\nabla = \partial/\partial \mathbf{u}$ denotes the Jacobian with respect to material-space coordinates, discretized on faces using the finite element basis functions. A similar metric can be written for the velocities,

$$\|\mathbf{v}_i - \mathbf{v}_j\|^2 = \mathbf{u}_{ij}^T (\nabla \mathbf{v}^T \nabla \mathbf{v}) \mathbf{u}_{ij}. \quad (5)$$

⁶If we stored the sizing field on faces during the remeshing process, it would be difficult to assign values to new faces after each operation without an expensive recomputation. Interpolating onto vertices does have a side effect of introducing a small amount of smoothing in the sizing field, but we find that this helps to create meshes that are smoothly graded.

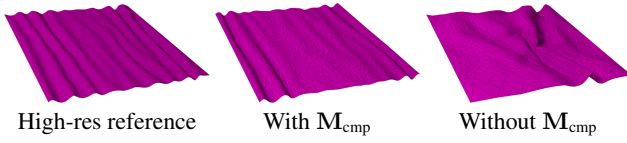


Figure 5: A square sheet of cloth under compression from two sides tends to buckle and form wrinkles.

These gradient terms can be conveniently expressed as positive definite quadratic forms,

$$\mathbf{M}_{\text{crv}} = \nabla \mathbf{n}^T \nabla \mathbf{n}, \quad (6)$$

$$\mathbf{M}_{\text{vel}} = \nabla \mathbf{v}^T \nabla \mathbf{v}. \quad (7)$$

As noted above, curvature alone is not sufficient to resolve wrinkles in adaptively remeshed cloth, as a coarse mesh is less likely to buckle in the first place. Even when it does, it will be forced to do so along the coarse mesh edges before refinement can take place, producing an unrealistic result as shown in Figure 5. In order to allow realistically fine wrinkles to appear, the remeshing scheme must anticipate the buckling of the material before it happens. For this purpose, we define a “compression tensor” \mathbf{M}_{cmp} which describes the expected minimum size of wrinkles in the mesh.

We determine \mathbf{M}_{cmp} through the following heuristic argument. Consider a strip of material of length ℓ with stretching and bending stiffnesses k_s and k_b , clamped at both ends to be under a small compressive strain ε . In the flat state, its elastic energy per unit length is proportional to $k_s \varepsilon^2$. If the strip buckles into a single arc, its curvature is $\kappa \propto \sqrt{\varepsilon}/\ell$, leading to a bending energy on the order of $k_b \kappa^2$, or $k_b \varepsilon / \ell^2$. Thus, the buckled state is energetically favorable to the flat state when $\varepsilon \ell^2 > \text{const}$, with the constant depending on the ratio of stretching and bending stiffnesses. This reasoning suggests that the size of the smallest possible wrinkles in a cloth with compressive strain ε is proportional to $1/\sqrt{\varepsilon}$, and so the sizing field should be proportional to ε . To only consider compression, we take the negative part of the Green strain,

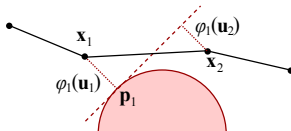
$$\mathbf{M}_{\text{cmp}} = (\mathbf{1} - \nabla \mathbf{x}^T \nabla \mathbf{x})^+. \quad (8)$$

Here $\mathbf{1}$ is the 2×2 identity matrix, and the operation $(\cdot)^+$ denotes replacing each eigenvalue ξ of the argument with $\max(\xi, 0)$.

Finally, we also anticipate the cloth’s interaction with obstacles by refining the mesh in regions close to obstacle geometry. For a given face, consider one of its vertices, i , and find the nearest point \mathbf{p}_i on any obstacle. We want to ensure that the face is not so large that that it crosses \mathbf{p}_i ’s tangent plane. If φ_i is the signed distance of a point on the cloth face to said tangent plane, then for any other vertex j of the face, we want $\varphi_i(\mathbf{u}_j) = \varphi_i(\mathbf{u}_i) + \nabla \varphi_i \cdot \mathbf{u}_{ji} > 0$, for which it is sufficient to require $\mathbf{u}_{ji}^T (\nabla \varphi_i \nabla \varphi_i^T) \mathbf{u}_{ji} < \varphi_i(\mathbf{u}_i)^2$. Therefore, for each face, we define an obstacle-related sizing tensor at its vertices $i = 0, 1, 2$ as

$$\mathbf{M}_{\text{obs}}^i = \frac{\nabla \varphi_i \nabla \varphi_i^T}{\varphi_i(\mathbf{u}_i)^2} \quad (9)$$

and define the tensor for the entire face as the average of the three tensors at the vertices. A two-dimensional analogue of this procedure is illustrated below: $\mathbf{M}_{\text{obs}}^1$ is defined such that the element is split if $\varphi_1(\mathbf{u}_2)$ is negative.



To control the remeshing, we specify global bounds on change in vertex normal Δn_{max} , material compression c_{max} , and velocity difference Δv_{max} across any edge. We refer to these as *dynamical* remeshing criteria, as they relate to the physical behavior of the material, in contrast with the *geometrical* criteria we later introduce regarding the shapes and sizes of the faces themselves. First, we define a preliminary sizing field

$$\hat{\mathbf{M}} = \frac{\mathbf{M}_{\text{crv}}}{\Delta n_{\text{max}}^2} + \frac{\mathbf{M}_{\text{cmp}}}{c_{\text{max}}^2} + \frac{\mathbf{M}_{\text{vel}}}{\Delta v_{\text{max}}^2} + \mathbf{M}_{\text{obs}}. \quad (10)$$

This has the property that for any edge ij , the condition $\mathbf{u}_{ij}^T \hat{\mathbf{M}} \mathbf{u}_{ij} \leq 1$ guarantees that the dynamical criteria are satisfied.

However, simply using $\hat{\mathbf{M}}$ would allow a flat sheet of material at rest to have arbitrarily large faces, and conversely, a sharp crease would result in arbitrarily small faces. Therefore, we also specify geometrical criteria in terms of the minimum and maximum edge lengths ℓ_{min} and ℓ_{max} , and a minimum aspect ratio $\alpha_{\text{min}} \in [0, 1]$. These serve to prevent the faces from becoming undesirably large, small, or skinny. Rather than enforcing these through additional remeshing constraints that may conflict with the effect of the dynamical criteria, we directly modify the sizing field to comply with the geometrical criteria. Let the eigendecomposition of the preliminary sizing field be $\hat{\mathbf{M}} = \mathbf{Q} \hat{\Lambda} \mathbf{Q}^T$, with eigenvalue matrix $\hat{\Lambda} = \text{diag}(\hat{\lambda}_1, \hat{\lambda}_2)$. We clamp the eigenvalues to the range $[\ell_{\text{max}}^{-2}, \ell_{\text{min}}^{-2}]$, and further clamp the smaller one to be at least α_{min}^2 times the larger. In other words, we set

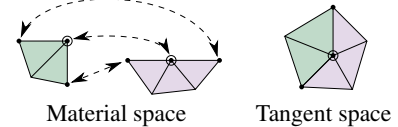
$$\tilde{\lambda}_i = \text{clamp}(\hat{\lambda}_i, [\ell_{\text{max}}^{-2}, \ell_{\text{min}}^{-2}]), \quad (11)$$

$$\tilde{\lambda}_{\text{max}} = \max(\tilde{\lambda}_1, \tilde{\lambda}_2), \quad (12)$$

$$\lambda_i = \max(\tilde{\lambda}_i, \alpha_{\text{min}}^2 \tilde{\lambda}_{\text{max}}), \quad (13)$$

and define the final sizing field as $\mathbf{M} = \mathbf{Q} \Lambda \mathbf{Q}^T$ with eigenvalue matrix $\Lambda = \text{diag}(\lambda_1, \lambda_2)$. If isotropic elements are desired, one can set $\alpha_{\text{min}} = 1$ in this step.

Sizing at seam vertices. At seam vertices, the question arises of how to perform area-weighted averaging of the sizing field between faces that are disconnected in material space. We construct a single tangent space for all the vertices associated with the same world-space position, by transforming all the adjacent faces so that they line up along the seam edges. For vertices created by edge splits, this is accomplished by a rotation. However, seam vertices in the original input may not be flat, meaning that their incident face angles do not add up to 2π . We linearly scale these angles to flatten the vertex, as shown below.



Having constructed a common tangent space, we take the sizing tensors at faces incident to all colocated vertices and rotate them to this tangent space before averaging. When evaluating the size of an edge, we rotate the vertex’s averaged tensor back out of the tangent space into the material-space neighborhood of the vertex to obtain the \mathbf{M} for the edge size formula (1).

Handling interpenetrations. Each of the three remeshing operations discontinuously changes the cloth geometry, and so can cause the cloth to penetrate obstacles or itself. One could use collision detection while remeshing to avoid introducing interpenetrations in the first place [Brochu and Bridson 2009; Brochu et al. 2012], or resolve them at the end of remeshing, say by applying global intersection

analysis [Baraff et al. 2003]. We take the latter strategy, but with an approach described below that is much simpler and cheaper than performing an expensive topological analysis. We found that this simpler approach is adequate for our problem because the changes in geometry produced by remeshing tend to be small and roughly in the normal direction of the surface.

The material-space coordinates give us a one-to-one correspondence between points on the mesh before remeshing, M_{old} , and after, M . Knowing that M_{old} was free of interpenetrations, if a face j in M intersects an obstacle, it can only be because points on the face were moved towards the obstacle by a remeshing operation, so we merely need to move it back to where it “came from”. Specifically, we find the midpoint \mathbf{x} of the line segment common to face j and the face of the obstacle it intersects, determine its displacement $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}_{old}$ relative to its position before remeshing, and move the face along $-\Delta\mathbf{x}$ until the faces no longer intersect. The triangle is moved by moving its vertices weighted by the barycentric coordinates of \mathbf{x} . We can do the same for two interpenetrating cloth faces, choosing $\Delta\mathbf{x}$ as the change in the *relative* positions of the now-coincident points on the two faces. There will in general be multiple pairs of interpenetrations; we treat them simultaneously by treating their required translations along the respective $-\Delta\mathbf{x}$ as constraints and solving them using impact zones [Harmon et al. 2008].

5 Robust Strain Limiting on Irregular Meshes

Existing strain-limiting methods use solvers based on Jacobi or Gauss-Seidel iterations. While these types of solvers work well for small-to-moderate sized systems, they converge slowly for larger ones. The slow convergence for large systems is a well-known limitation of Jacobi and Gauss-Seidel solvers which require many iterations to propagate information from one end of the mesh to another. We have also observed that these solvers become unreliable and difficult to keep stable when applied to our highly irregular and anisotropic meshes. This behavior occurs because a single Jacobi step applied to one face can drastically change the geometry of nearby faces if they are much smaller in size, as illustrated in Figure 6.

Instead, we pose the strain limiting problem as a nonlinearly constrained optimization problem. Given a triangle mesh with vertices at current positions $\hat{\mathbf{x}}_i$, we seek new positions \mathbf{x}_i closest to their current positions, such that the strains of all faces lie within the user-specified strain limits $[\varepsilon_{min}, \varepsilon_{max}]$. In this form, the problem can be solved using standard techniques for constrained optimization. We use the augmented Lagrangian method for its combination of generality and simplicity: in particular, it can handle nonlinear and non-convex inequality constraints, and can be applied without evaluation of higher-order derivatives of the objective and constraint functions.

We perform strain limiting once at every time step before performing collision resolution. After the new vertex positions are computed, we update the velocities using the change in positions,

$$\mathbf{v}_i \leftarrow \mathbf{v}_i + (\mathbf{x}_i - \hat{\mathbf{x}}_i)/\Delta t. \quad (14)$$

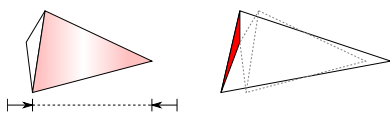


Figure 6: A failure of Jacobi iterations for strain limiting. The large face is initially compressed by a small amount, but enlarging it to its rest size inverts an adjacent face.

Figure 7 compares the behavior of our approach and a Jacobi solver for enforcing strain limits on a heavily deformed high-resolution mesh. Both methods can easily remove compressive strain by introducing high-frequency out-of-plane wrinkles, but tensile strain requires a low-frequency response at which Jacobi iterations perform poorly. The augmented Lagrangian method, on the other hand, rapidly drives both compressive and tensile face strains close to the strain limits, and soon terminates.

5.1 The Augmented Lagrangian Method

For completeness, we briefly review the augmented Lagrangian method in general form here. A detailed analysis can be found in the text by Nocedal and Wright [2006, chapter 17.4].

Consider an inequality-constrained optimization problem of the form

$$\min f(\mathbf{x}) \quad \text{s.t. } \mathbf{g}(\mathbf{x}) \leq 0 \quad (15)$$

where $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is the objective function on an n -dimensional domain, and $\mathbf{g}: \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a vector-valued function representing m constraints. We can write this equivalently as

$$\min f(\mathbf{x}) \quad \text{s.t. } \hat{\mathbf{g}}(\mathbf{x}, \mathbf{s}) = 0, \quad (16)$$

where $\hat{\mathbf{g}}(\mathbf{x}, \mathbf{s}) = \mathbf{g}(\mathbf{x}) + \mathbf{s}$, with $\mathbf{s} \in \mathbb{R}_+^m$ being nonnegative slack variables. The augmented Lagrangian method replaces the constraints with a combination of Lagrange multipliers and penalty functions, defining the augmented Lagrangian as the function

$$\mathcal{L}_A(\mathbf{x}, \mathbf{s}; \boldsymbol{\lambda}, \mu) = f(\mathbf{x}) + \boldsymbol{\lambda}^T \hat{\mathbf{g}}(\mathbf{x}, \mathbf{s}) + \frac{\mu}{2} \|\hat{\mathbf{g}}(\mathbf{x}, \mathbf{s})\|^2. \quad (17)$$

Here $\boldsymbol{\lambda} \in \mathbb{R}^m$ and $\mu \in \mathbb{R}$ are parameters, $\boldsymbol{\lambda}$ in particular acting as an estimate of the Lagrange multipliers. The method proceeds by alternating between two steps:

1. Update \mathbf{x} and \mathbf{s} by minimizing \mathcal{L}_A for fixed $\boldsymbol{\lambda}$ and μ .
2. Update $\boldsymbol{\lambda}$ via the first-order update rule

$$\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \mu \hat{\mathbf{g}}(\mathbf{x}, \mathbf{s}). \quad (18)$$

It can be shown that for large enough μ these iterations converge, with $\boldsymbol{\lambda}$ approaching the true Lagrange multipliers and \mathbf{x} approaching the constrained solution. We take $\mu = 10^3$, which we found to be adequate in practice.

In our implementation, we perform an additional simplification. We eliminate \mathbf{s} as a variable, replacing it with its optimal value, $\mathbf{s}^* = \max(-\mathbf{g}(\mathbf{x}) - \boldsymbol{\lambda}/\mu, 0)$ the maximum being taken componentwise. After some algebra, this reduces the augmented Lagrangian to

$$\mathcal{L}_A(\mathbf{x}; \boldsymbol{\lambda}, \mu) = f(\mathbf{x}) + \frac{\mu}{2} \|\tilde{\mathbf{g}}(\mathbf{x})\|^2 - \frac{\|\boldsymbol{\lambda}\|^2}{2\mu}, \quad (19)$$

and the update rule to

$$\boldsymbol{\lambda} \leftarrow \mu \tilde{\mathbf{g}}(\mathbf{x}), \quad (20)$$

where the penalty function $\tilde{\mathbf{g}}$ is given by

$$\tilde{\mathbf{g}}(\mathbf{x}) = \max(\mathbf{g}(\mathbf{x}) + \boldsymbol{\lambda}/\mu, 0). \quad (21)$$

The minimization of the simplified augmented Lagrangian is subject to no constraints, and can be performed using standard techniques for unconstrained optimization. The derivation of these equations is given in this paper’s supplementary material.

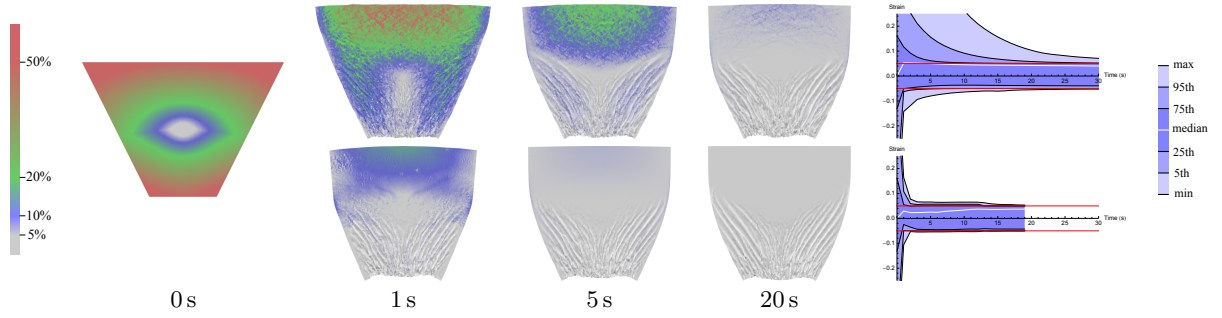


Figure 7: (left) We take a square mesh of over 20k faces and stretch one side to $1.5\times$ its rest length and compress the opposite side to $0.5\times$, then apply isotropic strain limits of $[-5\%, 5\%]$ using (middle top) Jacobi iterations and (middle bottom) the augmented Lagrangian method. Intermediate results are shown with faces colored according to their worst absolute strain. (right) The distribution of face strains over time is visualized, with the limits $[-5\%, 5\%]$ overlaid in red.

5.2 Strain Limiting

The application to strain limiting is straightforward. Our objective function is simply the sum of mass-weighted squared displacements of the vertices,

$$f(\mathbf{x}) = \frac{1}{2} \sum_{\text{vertex } i} m_i \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2. \quad (22)$$

Assuming isotropic strain limits, we have for each face j four scalar constraints, namely upper and lower bounds on the two principal strains:

$$g_{j\alpha, \text{lower}}(\mathbf{x}) = w_j(\varepsilon_{\min} - \varepsilon_{j\alpha}(\mathbf{x})) \quad (23)$$

$$g_{j\alpha, \text{upper}}(\mathbf{x}) = w_j(\varepsilon_{j\alpha}(\mathbf{x}) - \varepsilon_{\max}) \quad (24)$$

where $\alpha = 1, 2$ indexes over the two principal strains. We take the weight w_j to be the square root of the material-space area of the face, as this gives better conditioning. The principal strains $\varepsilon_{j\alpha}$ are determined as one less than the singular values of the deformation gradient $\mathbf{F} = \nabla \mathbf{x}$, computed using the finite element basis matrix.

To minimize \mathcal{L}_A as defined in (19), we may use any unconstrained nonlinear optimization routine. The nonlinear conjugate gradient method and the L-BFGS algorithm are good choices, as they do not require the full Hessian of the objective (in our case, the augmented Lagrangian); in our experiments using the ALGLIB numerical analysis library,⁷ the nonlinear conjugate gradient method was less expensive. Both the above methods require the first derivative to be provided analytically. This involves the first derivatives of the face strains $\varepsilon_{j\alpha}$ with respect to the vertex positions \mathbf{x} ; equivalently, we require the first derivatives of the singular values of the deformation gradient \mathbf{F} .

These derivatives can be computed from two facts. First, the deformation gradient is linear in the vertex positions \mathbf{x} . In particular, it is given by

$$\mathbf{F} = \mathbf{X}\Delta\beta, \quad (25)$$

where $\mathbf{X} = [\mathbf{x}_0 \ \mathbf{x}_1 \ \mathbf{x}_2]$ contains the positions of the face’s vertices, Δ is the 3×2 matrix which maps \mathbf{X} to $\mathbf{X}\Delta = [\mathbf{x}_1 - \mathbf{x}_0 \ \mathbf{x}_2 - \mathbf{x}_0]$, and β is the finite element basis matrix for the face. Second, if the singular value decomposition of \mathbf{F} is $\mathbf{F} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, then the derivatives of its k th singular value with respect to the entries of \mathbf{F} is given by [Papadopoulos and Lourakis 2000]

$$\frac{\partial d_k}{\partial f_{ij}} = u_{ik}v_{jk}. \quad (26)$$

Combining these facts, we find that the gradient of d_k with respect to \mathbf{x}_ℓ is

$$\nabla_{\mathbf{x}_\ell} d_k = (\Delta \ell \cdot \beta \mathbf{V} \cdot k) \mathbf{U} \cdot k, \quad (27)$$

where \mathbf{A}_i and \mathbf{A}_j denote the i th row and j th column of a matrix \mathbf{A} . Again, the full derivation is given in the supplementary material. With this, we can compute the derivatives of face strains with respect to any incident vertex.

In the above discussion, we have treated only the case of isotropic strain limiting. If coordinate-aligned (componentwise) strain limits are desired, as in some previous work [Goldenthal et al. 2007; Thomaszewski et al. 2009], the limits can be applied to the diagonal entries of the Green strain $\mathbf{G} = \mathbf{F}^T \mathbf{F} - \mathbf{1}$ instead. Using equation (25), one can directly differentiate \mathbf{G} to find the corresponding gradients.

6 Results and Discussion

Figure 2 shows an example of a $1 \text{ m} \times 1 \text{ m}$ square piece of cloth interacting with a moving sphere. A uniform high-resolution mesh with no edge longer than 10 mm, shown in (a), has 65.5k faces and takes over a minute of computation time per 25 Hz frame. For our adaptive anisotropic remeshing scheme, shown in (b), we set $\ell_{\min} = 10 \text{ mm}$, $\ell_{\max} = 200 \text{ mm}$, and $\alpha_{\min} = 0.1$. The number of faces in the mesh varies from just 128 to 25.4k over the course of the simulation. This simulation runs seven times faster, taking an average of 9 seconds per frame. Disabling anisotropy by setting $\alpha_{\min} = 1$, as in (c), gives a more uniform mesh, but with excessive refinement along long wrinkles. We also generated a medium-resolution mesh with a similar number of faces as the average of the adaptive anisotropic mesh. This took 7.6 seconds per frame, but shows visibly coarser detail and does not match the high-resolution behavior as well as the adaptively remeshed simulation does.

Detailed numbers for the above and other examples are reported in Table 1. Full animated results are shown in the supplementary video.

To illustrate that remeshing can handle wrinkles that move continuously across the material, we show in Figure 8 a simulation of a $3 \text{ m} \times 2 \text{ m}$ flag in a constant 10 m/s wind field [Wejchert and Haumann 1991]. The “waves” in the flag move unimpeded across the length of the material while the mesh dynamically adapts to their motion.

Figure 9 shows a cloth sleeve over a pair of cylinders connected by a ball joint. The cloth exhibits a characteristic diamond-shaped buckling pattern when the joint is bent. Remeshing automatically refines and coarsens the mesh as its configuration changes. Anisotropic meshing allows fine resolution in the circumferential direction without needless perpendicular refinement.

⁷Sergey Bochkhanov and Vladimir Bystritsky, <http://www.alglib.net/>

Name	Figure	Faces		Time per frame (seconds)				
		Input	Mean	Integration	Strain lim.	Collision	Remeshing	Total
Kick	1	884	37,627	13.055	14.075	22.300	0.551	49.980
Sphere – high-res	2(a)		65,536	34.861	25.36	4.894		65.116
– anisotropic	2(b)	4	11,313	4.077	3.557	1.222	0.177	9.033
– isotropic	2(c)	4	30,325	14.731	11.177	2.608	0.549	29.064
– average	2(d)		11,319	4.481	2.469	0.630		7.580
Flag	8	97	11,710	3.103	3.559	1.353	0.428	8.443
Cylinders	9	32	5,267	0.684	0.625	0.455	0.050	1.814
Blue dress	10(a)	112	13,149	4.028	3.656	3.811	0.196	11.691
Yellow dress	10(b)	428	22,785	10.075	6.997	5.396	0.340	22.809

Table 1: Performance timings for all our examples. These were collected on a quad-core 3.30 GHz Intel Core i5 CPU with 3.75 GB RAM. Disk I/O and rendering times are not included in the total. Most examples used $\Delta n_{\max} = 0.15$, $\Delta v_{\max} = 0.7$ m/s, $c_{\max} = 7 \times 10^{-2}$, $(\ell_{\min}, \ell_{\max}) = (15$ mm, 200 mm), $\alpha_{\min} = 0.1$.

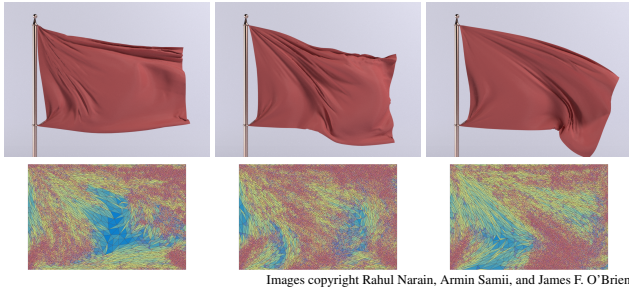


Figure 8: An adaptively remeshed flag flapping in the wind.

We show three examples of garments worn by animated characters in Figures 1 and 10. The mannequins have rigid parts and are driven by motion capture data from [Zordan and Van Der Horst 2003]. The blue dress worn by the female is composed of five panels illustrated in Figure 3, and the yellow dress has eleven panels. The male’s pants and t-shirt are composed of four panels each, and his vest six panels. The fine wrinkles and long folds in the garments show visually significant dynamic motion that cannot easily be captured in a separate detail model. The anisotropic nature of our meshes also causes edges to tend to align parallel to such features, which generally improves the appearance of the rendered images.

As Table 1 shows, the cost of remeshing itself is negligible compared to the rest of the simulation. In all of our examples, it accounts for between 1% and 5% of the total computation time.

Apart from performance, another significant benefit of adaptive refinement is ease of use: it frees the animator from concerns of choosing the right resolution of the simulation mesh that will run quickly while capturing all the desired detail. In fact, all of our results were created from extremely coarse input meshes containing only a few hundred faces.

All of the examples shown use our new strain limiting technique, as Jacobi-style iterations simply do not converge on our highly nonuniform meshes. We applied isotropic strain limits of $[-10\%, 10\%]$ on all examples. Our method works equally well on tighter strain limits. In particular, when dressing one of our mannequins we kinematically stretch the garment around the figure and then apply strain limiting with both strain limits set to zero. This process brings the input mesh as close as possible to an initial unstretched configuration, and prevents rapid transient motion at the beginning of the simulation.

As noted previously, we are using the constitutive model and measured parameters published by Wang et al. [2011]. We have found that the appearance of cloth is significantly improved with this model compared to simple isotropic elastic models with ad hoc parameters. The pants, t-shirt, and vest in Figure 1 use materials “11oz Black

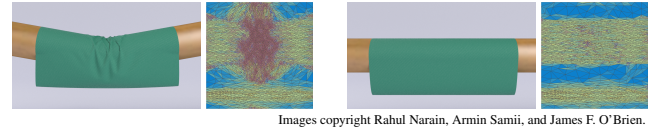


Figure 9: Wrinkles appear and disappear as a pair of connected cylinders bend and straighten.

Denim,” “Gray Interlock,” and “Camel Ponte Roma” respectively; the sheet in Figure 2 uses a modified “Gray Interlock” with 5 times the density and stiffness; the flag in Figure 8 uses “Camel Ponte Roma;” and the dresses in Figure 10 use (a) modified “Navy Sparkle Sweat” with 1/10th the bending stiffness, and (b) “Gray Interlock.” While these measured data and data from other sources [Volino et al. 2009; Miguel et al. 2012] are a great resource, further work remains to be done for developing comprehensive constitutive models for cloth. For example, the denim material in Figure 1 bends and creases in a way that is fairly consistent with denim, but its dynamic motion appears more consistent with a lighter material. Measurements of properties, such as the internal damping of denim, would likely improve the realism for that example.

In summary, we have presented a technique for dynamically refining a triangular finite element mesh, producing anisotropic elements with controllable size and shape to adaptively resolve detail in cloth simulations. The remeshing scheme preserves dynamical detail by anticipating the buckling of the material and its interaction with obstacles. We have also described an optimization-based approach that efficiently performs strain limiting on large and irregular meshes using the augmented Lagrangian method.

6.1 Limitations

In scenarios where the vertex velocities are smoothly varying, for example in a uniform rotation of the entire mesh, the velocity difference term could cause slightly more refinement of the mesh than strictly necessary. However, this conservative approach preserves subtle waves in the cloth for general motions.

For most of the parameters controlling the remeshing criteria, the same values work well for many different materials. However, the compression criterion c_{\max} is quite material-dependent: a material with high bending stiffness does not need to be refined as much under compression. At present, we have not worked out the exact relationship between the stretching and bending stiffnesses that should govern this parameter; doing so seems particularly difficult for the realistic nonlinear material models we use. A related limitation is that if a material is modeled as exactly incompressible, the compressive Green strain will always be zero, and the compression criterion will not be useful to predict buckling.



Images copyright Rahul Narain, Armin Samii, and James F. O'Brien.

Figure 10: Two different dresses simulated on a dancing character.

Working with dynamic data structures does make efficient implementation harder compared to working with static structures. For example, we do not attempt to maintain a cache-friendly memory layout of the mesh. However, we think the benefits from adaptive data structures justify the added complexity, and believe that more efficient implementation is still feasible with further work.

Our current strain limiting approach is not aware of self-collision constraints, which are left to the subsequent collision resolution procedure. Collision resolution may thus cause the cloth to violate the strain limits in colliding regions. In particular, this causes difficulties in scenarios like a tightly stretched knot. We believe this limitation could be overcome by incorporating additional constraints in the strain limiting optimization.

Acknowledgments

The authors thank the following people who supported this work: Victor Zordan for helping with the garment modeling, building the character models, and providing motion data; Juan Miguel de Joya for helping with the garment modeling; Jonathan Shewchuk for invaluable discussions concerning remeshing and anisotropy; Huamin Wang for the use of his code and for insightful discussions; and Alexander Reshetov and Alexei Soupikov for helping with code optimization.

This work was supported by funding from the Intel Science and Technology Center for Visual Computing, NSF Grant IIS-0915462, UC Lab Fees Research Program Grant 09-LR-01-118889-OB RJ, and a gift from Pixar Animation Studios.

The images in this paper were rendered using the Mitsuba renderer by Wenzel Jakob.⁸

References

- ALLIEZ, P., ATTENE, M., GOTSMAN, C., AND UCELLI, G. 2007. Recent advances in remeshing of surfaces. In *Shape Analysis and Structuring*. Springer, November, ch. 2.
- BARAFF, D., AND WITKIN, A. 1998. Large steps in cloth simulation. *Proc. SIGGRAPH '98*, 43–54.
- BARAFF, D., WITKIN, A., AND KASS, M. 2003. Untangling cloth. *ACM Trans. Graph.* 22, 3 (July), 862–870.
- BARGTEIL, A. W., WOJTAN, C., HODGINS, J. K., AND TURK, G. 2007. A finite element method for animating large viscoplastic flow. *ACM Trans. Graph.* 26, 3.
- BICKEL, B., BÄCHER, M., OTADUY, M. A., MATUSIK, W., PFISTER, H., AND GROSS, M. 2009. Capture and modeling of non-linear heterogeneous soft tissue. *ACM Trans. Graph.* 28, 3 (July), 89:1–89:9.

- BOSSEN, F., AND HECKBERT, P. 1996. A pliant method for anisotropic mesh generation. In *5th Intl. Meshing Roundtable*, 63–74.
- BRIDSON, R., FEDKIW, R., AND ANDERSON, J. 2002. Robust treatment of collisions, contact and friction for cloth animation. In *Proc. ACM SIGGRAPH 02*, 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*, Eurographics Association, SCA '03, 28–36.
- BROCHU, T., AND BRIDSON, R. 2009. Robust topological operations for dynamic explicit surfaces. *SIAM Journal on Scientific Computing* 31, 4, 2472–2493.
- BROCHU, T., EDWARDS, E., AND BRIDSON, R. 2012. Efficient geometrically exact continuous collision detection. *ACM Trans. Graph.* 31, 4 (July), 96:1–96:7.
- CARIGNAN, M., YANG, Y., THALMANN, N. M., AND THALMANN, D. 1992. Dressing animated synthetic actors with complex deformable clothes. In *Proc. ACM SIGGRAPH 92*, 99–104.
- CATMULL, E., AND CLARK, J. 1978. Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-Aided Design* 10, 6, 350 – 355.
- CHENTANEZ, N., FELDMAN, B. E., LABELLE, F., O'BRIEN, J. F., AND SHEWCHUK, J. R. 2007. Liquid simulation on lattice-based tetrahedral meshes. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2007*, 219–228.
- CHOI, K.-J., AND KO, H.-S. 2002. Stable but responsive cloth. In *Proc. ACM SIGGRAPH 02*, 604–611.
- DE AGUIAR, E., SIGAL, L., TREUILLE, A., AND HODGINS, J. K. 2010. Stable spaces for real-time clothing. *ACM Transactions on Graphics* 1, 212.
- DECAUDIN, P., JULIUS, D., WITHER, J., BOISSIEUX, L., SHEFFER, A., , AND CANI, M. 2006. Virtual garments: A fully geometric approach for clothing design. *Computer Graphics Forum* 25, 3, 625–634.
- ENGLISH, E., AND BRIDSON, R. 2008. Animating developable surfaces using nonconforming elements. In *Proc. ACM SIGGRAPH 08*, ACM, SIGGRAPH '08, 66:1–66:5.
- FENG, W.-W., YU, Y., AND KIM, B.-U. 2010. A deformation transformer for real-time cloth animation. *ACM Transactions on Graphics* 1, 212, 1–9.
- GARLAND, M., AND HECKBERT, P. S. 1997. Surface simplification using quadric error metrics. In *Proc. ACM SIGGRAPH 97*, SIGGRAPH '97, 209–216.
- GARLAND, M., AND HECKBERT, P. S. 1998. Simplifying surfaces with color and texture using quadric error metrics. In *Proc. Conference on Visualization '98*, VIS '98, 263–269.
- GOLDENTHAL, R., HARMON, D., FATTAL, R., BERCOVIER, M., AND GRINSPUN, E. 2007. Efficient simulation of inextensible cloth. *ACM Trans. Graph.* 26, 3 (July).
- GRINSPUN, E., KRYSL, P., AND SCHRÖDER, P. 2002. CHARMS: a simple framework for adaptive simulation. In *Proc. ACM SIGGRAPH 2002*, 281–290.
- 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. *SIGGRAPH (ACM Transactions on Graphics)* 27, 3 (Aug), 1–4.

⁸<http://www.mitsuba-renderer.org>

- HILSMANN, A., AND EISERT, P. 2012. Image-based animation of clothes. *Eurographics*, 1–4.
- HOPPE, H. 1996. Progressive meshes. In *Proc. ACM SIGGRAPH 96*, ACM, SIGGRAPH '96, 99–108.
- 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.
- JIAO, X., COLOMBI, A., NI, X., AND HART, J. 2006. Anisotropic mesh adaptation for evolving triangulated surfaces. In *Proc. 15th International Meshing Roundtable*, Springer, P. P. Pébay, Ed., 173–190.
- KAVAN, L., GERSZEWSKI, D., BARGTEIL, A. W., AND SLOAN, P.-P. 2011. Physics-inspired upsampling for cloth simulation in games. *ACM Transactions on Graphics* 1, 212, 1–10.
- KLINGNER, B. M., AND SHEWCHUK, J. R. 2007. Aggressive tetrahedral mesh improvement. In *Proc. 16th International Meshing Roundtable*, 3–23.
- KLINGNER, B. M., FELDMAN, B. E., CHENTANEZ, N., AND O'BRIEN, J. F. 2006. Fluid animation with dynamic meshes. In *Proc. ACM SIGGRAPH 2006*, 820–825.
- LABELLE, F., AND SHEWCHUK, J. R. 2003. Anisotropic voronoi diagrams and guaranteed-quality anisotropic mesh generation. In *Proc. 19th annual Symposium on Computational Geometry*, SCG '03, 191–200.
- LI, L., AND VOLKOV, V. 2005. Cloth animation with adaptively refined meshes. In *Proc. 28th Australasian Computer Science Conference*, vol. 38.
- LOOP, C. 1987. *Smooth subdivision surfaces based on triangles*. Master's thesis, University of Utah.
- MARTIN, S., THOMASZEWSKI, B., GRINSPUN, E., AND GROSS, M. 2011. Example-based elastic materials. *ACM SIGGRAPH 1*, 212, 1.
- MAX, N. 1999. Weights for computing vertex normals from facet normals. *J. Graph. Tools* 4, 2 (Mar.), 1–6.
- MIGUEL, E., BRADLEY, D., THOMASZEWSKI, B., BICKEL, B., MATUSIK, W., OTADUY, M. A., AND MARSCHNER, S. 2012. Data-driven estimation of cloth simulation models. *Eurographics* 31, 2.
- MÜLLER, M., AND CHENTANEZ, N. 2010. Wrinkle meshes. *ACM SIGGRAPH Symposium on Computer Animation*.
- MÜLLER, M., AND GROSS, M. 2004. Interactive virtual materials. In *Proc. Graphics Interface 2004*, Canadian Human-Computer Communications Society, GI '04, 239–246.
- NEALEN, A., MLLER, M., KEISER, R., BOXERMAN, E., AND CARLSON, M. 2006. Physically based deformable models in computer graphics. *Computer Graphics Forum* 25, 4, 809–836.
- NOCEDAL, J., AND WRIGHT, S. J. 2006. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York.
- PAPADOPOULOU, T., AND LOURAKIS, M. 2000. Estimating the Jacobian of the singular value decomposition: Theory and applications. In *Computer Vision - ECCV 2000*, vol. 1842 of *Lecture Notes in Computer Science*. 554–570.
- POPA, T., ZHOU, Q., BRADLEY, D., KRAEVOY, V., FU, H., SHEFFER, A., AND HEIDRICH, W. 2009. Wrinkling captured garments using spacetime datadriven deformation. *Computer Graphics* 28, 2.
- POPOVIĆ, J., AND HOPPE, H. 1997. Progressive simplicial complexes. In *Proc. ACM SIGGRAPH 97*, 217–224.
- PROVOT, X. 1995. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *Proc. Graphics Interface 95*, 147–154.
- ROHMER, D., POPA, T., CANI, M.-P., HAHMANN, S., AND SHEFFER, A. 2010. Animation wrinkling : Augmenting coarse cloth simulations with realistic-looking wrinkles. *ACM Transactions on Graphics* 29, 6, 1–8.
- RUSINKIEWICZ, S. 2004. Estimating curvatures and their derivatives on triangle meshes. In *Symposium on 3D Data Processing, Visualization, and Transmission*, 486–493.
- SHEWCHUK, J. R., 2002. What is a good linear finite element? Interpolation, conditioning, anisotropy, and quality measures.
- SIMNETT, T. J. R., LAYCOCK, S. D., AND DAY, A. M. 2009. An edge-based approach to adaptively refining a mesh for cloth deformation. *Theory and Practice of Computer Graphics*, 77–84.
- TANG, M., MANOCHA, D., AND TONG, R. 2010. Fast continuous collision detection using deforming non-penetration filters. In *Proc. 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ACM, 7–13.
- THOMASZEWSKI, B., WACKER, M., STRAER, W., LYARD, E., LUIBLE, C., VOLINO, P., KASAP, M., MUGGEO, V., AND MAGNENAT-THALMANN, N. 2007. Advanced topics in virtual garment simulation. In *Eurographics 2007 - Tutorials*, 795–855.
- THOMASZEWSKI, B., PABST, S., AND STRASSER, W. 2009. Continuum-based strain limiting. *Eurographics* 28, 2.
- UMETANI, N., KAUFMAN, D. M., IGARASHI, T., AND GRINSPUN, E. 2011. Sensitive couture for interactive garment modeling and editing. *ACM Trans. Graph.* 30, 4 (Aug.), 90:1–90:12.
- VILLARD, J., AND BOROUCHE, H. 2002. Adaptive meshing for cloth animation. In *Proceedings, 11th International Meshing Roundtable*, Springer-Verlag, 243–252.
- VOLINO, P., MAGNENAT-THALMANN, N., AND FAURE, F. 2009. A simple approach to nonlinear tensile stiffness for accurate cloth simulation. *ACM Transactions on Graphics* 28, 4 (Aug.), 1–16.
- WANG, H., HECHT, F., RAMAMOORTHI, R., AND O'BRIEN, J. 2010. Example-based wrinkle synthesis for clothing animation. *ACM Transactions on Graphics* 29, 4, 1–8.
- WANG, H., O'BRIEN, J. F., AND RAMAMOORTHI, R. 2010. Multi-resolution isotropic strain limiting. In *Proc. ACM SIGGRAPH Asia 2010*, 160:1–10.
- WANG, H., O'BRIEN, J. F., AND RAMAMOORTHI, R. 2011. Data-driven elastic models for cloth: modeling and measurement. *ACM Transactions on Graphics* 30, 4.
- WEJCHERT, J., AND HAUMANN, D. 1991. Animation aerodynamics. *SIGGRAPH Comput. Graph.* 25, 4 (July), 19–22.
- WICKE, M., RITCHIE, D., KLINGNER, B. M., BURKE, S., SHEWCHUK, J. R., AND O'BRIEN, J. F. 2010. Dynamic local remeshing for elastoplastic simulation. In *Proc. ACM SIGGRAPH 2010*, 49:1–11.
- WOJTAN, C., AND TURK, G. 2008. Fast viscoelastic behavior with thin features. *ACM Trans. Graph.* 27, 3, 1–8.
- ZORDAN, V. B., AND VAN DER HORST, N. C. 2003. Mapping optical motion capture data to skeletal motion using a physical model. In *Proc. 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, SCA '03, 245–250.