EUROGRAPHICS 2009 / P. Dutré and M. Stamminger
(Guest Editors)

Volume 28 (2009), Number 2

Continuum-based Strain Limiting

Bernhard Thomaszewski Simon Pabst Wolfgang Straßer

WSI/GRIS, Universität Tübingen, Germany

Abstract

We present Continuum-based Strain Limiting (CSL) – a new method for limiting deformations in physically-based cloth simulations. Despite recent developments for nearly inextensible materials, the efficient simulation of general biphasic textiles and their anisotropic behavior remains challenging. Many approaches use soft materials and enforce limits on edge elongations, leading to discretization-dependent behavior. Moreover, they offer no explicit control over shearing and stretching unless specifically aligned meshes are used. Based on a continuum deformation measure, our method allows accurate control over all strain components using individual thresholds. We impose deformation limits element-wise and cast the problem as a 6×6 system of linear equations. CSL can be combined with any cloth simulator and, as a velocity filter, integrates seamlessly into standard collision handling.

1. Introduction

The notion of textiles covers a wide area of materials, ranging from quasi-inextensible, tightly woven cotton to soft and stretchable wool weaves. Between these extremes, most woven textiles can be approximated as biphasic materials: resistance to small deformations is typically weak, corresponding to yarns straightening out, but a high stiffness is observed beyond a material-specific threshold. Depending on the material of the yarns and on the weave pattern, a textile's stretch resistance can be substantially different in weft and warp directions. This yarn structure leads to anisotropic behavior and is also responsible for a much lower resistance to shear deformation.

Recent developments have led to solvers which excel at time stepping quasi-inextensible cloth [GHF+07, EB08]. Many textiles are, however, not inextensible and general biphasic materials are usually simulated using weaker elastic forces coupled with edge-based strain limiting [Pro95] and strain rate limiting [BFA02]. Since an edge can only respond to deformation along its direction, a regular warp/weft aligned mesh is necessary to reproduce anisotropic materials and deformation limits - a significant disadvantage for practical cloth animations (see Fig. 1). For the edge-constraint methods by Goldenthal et al. and English et al., this argument applies as well. We note that for quasi-inextensible materials, anisotropy is only of marginal importance since deformations are assumed to be very small. For many general textile materials, however, this assumption does not hold.

Motivated by this fact, we present a new approach to deformation limiting aimed at simulating biphasic, anisotropic materials. In contrast to edge-based methods, our approach is based on a continuum deformation measure. As such, it allows accurate control of stretch and shear deformations with individual limits, regardless of the underlying mesh. Our Continuum-based Strain Limiting (CSL) is simple to implement and can be used in a plug-and-play style to replace the common strain limiting in the widely used collision handling framework by Bridson et al. [BFA02]. Though we

Figure 1: *Continuum-based strain limiting (CSL) allows the simulation of anisotropic, biphasic materials by selectively imposing individual constraints for all deformation modes. CSL acts as a velocity filter and can thus replace edge-based strain limiting in standard collision handling frameworks.*

encourage the use of a continuum-based membrane model to map material anisotropy, our method can also augment conventional mass-spring systems with anisotropic limits. Despite the broader range of materials that can be reproduced with our method, its computational impact remains comparable to edge-based approaches.

Contributions We propose a novel deformation limiting method for anisotropic biphasic materials. We impose deformation limits element-wise based on the linear corotated strain tensor (Sec. 3). Using linear finite elements for discretization, we propose a simple and elegant way to obtain nodal displacements that exactly obey individual strain limits for *every* deformation mode at the same time (Sec. 4). The corresponding elemental corrections are physically sound since they are guaranteed to preserve both linear and angular momentum. We cast the elemental problem as a 6×6 linear system and show how to further improve computational efficiency using problem decomposition (Appendix A). There are several alternatives for combining element responses and we consider three of them in this work: direct enforcement, Jacobi and Gauss-Seidel iterative enforcement (Sec. 5). Our examples provide evidence for the necessity of anisotropic materials and show how our new method can faithfully reproduce this important behavior (Sec. 6). As demonstrated by further examples, our method fits seamlessly into the collision handling framework described by Bridson et al. (Sec. 4.1). Finally, Sec. 6 provides comparative computation times.

Before discussing our method in detail, we first summarize relevant related work in Sec. 2 and recapitulate basic deformation measures and their discretization in Sec. 3.

2. Related Work

Many existing approaches model textiles as sheets of elastic material [TPBF87, VCMT95, BW98, CK02]. Using only elastic forces, the compliance to deformation limits has to arise from stiff materials, degrading numerical efficiency [HE01]. But since biphasic cloth is not well approximated by stiff materials in the small deformation regime, nonlinear material models, e.g., in the form of strain-stress curves have to be used. This situation is more challenging for the stability of the (nonlinear) solver, leading to increased computation times. Regardless of the material model, over-elongations due to collision response can still occur.

Provot [Pro95] was the first to address the problem of over-elongations in cloth simulation. He suggested to use softer materials and to geometrically correct end points of overly strained edges in an iterative manner. Provot established a 10% deformation threshold and subsequent work followed his example. Bridson et al. [BFA02] extended Provot's technique to strain-rate limiting, recasting the method into a velocity-correcting formulation. Another extension was proposed by Müller et al. [MHHR06, Mül08] who use general position constraints as the basic simulation principle. In contrast to iterative methods, Hong et al. [HCJ+05] use a linearized implicit formulation for global enforcement of edge-length constraints.

Efficient methods for simulating quasi-inextensible cloth were introduced only recently, starting with the fast projection method by Goldenthal et al. [GHF+07]. Using implicit edge constraints on a quad-dominant mesh, very low strains in weft and warp directions can be enforced using this technique. Based on fast projection, English et al. [EB08] animate inextensible triangle meshes, enforcing zero in-plane deformation via edge-length constraints. Bergou et al. [BWR+08] simulate inextensible rods, using an extended version of fast projection as well.

Tsiknis [Tsi06] describes a three step scheme consisting of force estimation, strain limiting and global response. The second step forms an exception to traditional strain limiting as it works on triangles, not on edges. However, while our method simultaneously enforces constraints for every deformation mode, this approach can only handle one principal strain direction. Unfortunately, the first and third step, vital to his method, are again based on edge constraints and as such inherit the aforementioned disadvantages.

3. Deformation and Discretization

Edge-based elongation measures are not suitable for an accurate control of deformation and we will use a continuum approach instead. This section describes the formulation and discretization of the continuous strain measure used in this work. In the following, plain face symbols are used to denote continuous and tensorial quantities while bold face letters refer to matrices and vectors in the discrete setting.

Continuum Formulation A deformable solid in its current state is described by its configuration mapping φ. This mapping provides a Lagrangian description of the object in a displacement formulation as

$$x(t) = \varphi(X,t) = X + u(t)\ . \tag{1}$$

Here, $x(t)$ denotes the current position of the material particle initially positioned at X and $u(t)$ is the displacement field. Using this mapping we define a geometric deformation measure, the *Green Strain*, as

$$\varepsilon_G = \frac{1}{2}(\nabla\varphi^T\nabla\varphi - id) = \frac{1}{2}(\nabla u^T + \nabla u + \nabla u^T\nabla u)\ , \tag{2}$$

where id is the identity mapping. This expression is nonlinear in displacements and imposing deformation constraints on it would require the solution of a constrained nonlinear optimization problem, which is computationally expensive. We can simplify the problem by carrying out a linearization to obtain

$$\varepsilon_C = \frac{1}{2}(\nabla u^T + \nabla u)\ . \tag{3}$$

This tensor is not invariant under rotations, but supposing that the purely rotational part $R(t)$ of $u(t)$ is known, we can use the corotational formulation [MDM+02, EKS03, HS04]

$$\varepsilon_{CR}(\varphi) = \varepsilon_C(R^T\varphi)\ . \tag{4}$$

We will use this deformation measure in the remainder and write $\varepsilon := \varepsilon_{CR}$ as shorthand.

Linear Finite Elements In order to impose deformation limits on cloth meshes we have to discretize (4). For this purpose, we will make use of linear Finite Elements [ZT00]. A linear triangular element is characterized by its nodal positions in the current and rest configuration $\mathbf{x}_i$ and $\mathbf{p}_i$ and its three nodal *shape functions* N_i. The shape functions, which are depicted in Fig. 2, live in the 2D space of the planar element and are defined by the requirement $N_i(\mathbf{p}_j) = \delta_{ij}$. The approximation of the displacement field $\mathbf{u}$ over an element is

$$\mathbf{u}(x,y) = \sum_i N_i(x,y)\mathbf{u}_i\ ,$$

where $\mathbf{u}_i$ are nodal in-plane displacements in the 2D coordinate system of the element. Note that the transformation from an element's 2D space to 3D world space is merely a composition of rotation and translation – it is, however, never needed explicitly.

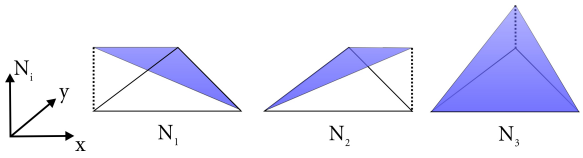

Figure 2: *Linear shape functions of a triangular element.*

Textiles are usually made of flat patterns such that an element's rest state and its orientation in the material plane are directly available. For curved objects, shape functions can still be computed but the material orientation cannot be determined automatically and has to be supplied by the user, e.g., via texture coordinates. For the linear triangle element, an explicit expression for the shape functions in terms of 2D Cartesian coordinates can readily be derived geometrically [ZT00]. We will take a different and more general approach here. Every triangular element can be transformed to a generic element as shown in Fig. 3.

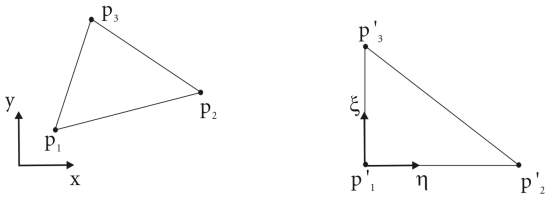

Figure 3: *Geometry of a triangular element (*left*) and its corresponding generic element (*right*).*

In this generic space with coordinates ξ and η, the shape functions can directly be read off from Fig. 3:

$$N_1 = 1-\xi-\eta, \quad N_2 = \xi, \quad N_3 = \eta\ .$$

Likewise, it can easily be verified that the shape function derivatives with respect to the generic coordinates follow as

$$\frac{\partial N_1}{\partial \vartheta} = \begin{bmatrix}-1\\-1\end{bmatrix}, \quad \frac{\partial N_2}{\partial \vartheta} = \begin{bmatrix}1\\0\end{bmatrix}, \quad \frac{\partial N_3}{\partial \vartheta} = \begin{bmatrix}0\\1\end{bmatrix}\ ,$$

where $\vartheta = \begin{bmatrix}\xi & \eta\end{bmatrix}^t$. In the following calculations, the shape functions themselves are not needed, only their derivatives are required. These are obtained using the chain rule as

$$\frac{\partial N_i}{\partial x_j} = \frac{\partial N_i}{\partial \xi}\frac{\partial \xi}{\partial x_j} + \frac{\partial N_i}{\partial \eta}\frac{\partial \eta}{\partial x_j}.$$

In practice, it is more convenient to first compute $\mathbf{J}$ with $\mathbf{J}_{ij} = \frac{\partial x_i}{\partial \vartheta_j}$ and then use $\mathbf{J}^{-1}$ in the above equation.

Discretization We can now proceed to calculating the strain for a given triangular element. In every time step, the simulator provides us with 3D world space displacements for all nodes. In order to compute the strain for a given element we have to extract the 2D rotation-free displacement field first. To this end, we use a polar decomposition of the deformation gradient, which is defined as

$$\mathbf{F}_{ij} = \sum_k \frac{\partial N_k}{\partial x_j}\tilde{\mathbf{x}}_{k,i}\ . \tag{5}$$

Here, $\tilde{\mathbf{x}}_{k,i}$ is the i-th component of nodal position $\tilde{\mathbf{x}}_k$ and the tilde denotes a 3D world space quantity. Since x_j span a 2D space we have $\mathbf{F} \in (3\times 2)$. As a visual interpretation, $\mathbf{F}$ maps vectors in the 2D undeformed element space to their deformed counterparts in 3D space. Since the deformation gradient can be written as $\mathbf{F} = \mathbf{RU}$, where $\mathbf{R} \in (3\times 2)$ is a rotation from the element's 2D rest state to its current configuration and $\mathbf{U} \in (2\times 2)$ is a pure deformation, we can determine $\mathbf{R}$ by polar decomposition [EKS03]. Having computed the rotation for a given element, we obtain its rotation-free 2D nodal displacements as $\mathbf{u}_i = \mathbf{R}^t\tilde{\mathbf{u}}_i$. We can now apply the Finite Element interpolation to the displacement field and write the strain in matrix form as

$$\boldsymbol{\varepsilon} = \sum_{i=1}^{3}\mathbf{B}_i\mathbf{u}_i = \mathbf{B}\mathbf{u}^e \tag{6}$$

where $\mathbf{u}^e = [\mathbf{u}_1\mathbf{u}_2\mathbf{u}_3]$ and $\mathbf{B}_i$ is the 2×3 linear strain-displacement matrix, defined as

$$\mathbf{B}_i = \begin{bmatrix}\frac{\partial N_i}{\partial x} & 0\\ 0 & \frac{\partial N_i}{\partial y}\\ \frac{1}{2}\frac{\partial N_i}{\partial y} & \frac{1}{2}\frac{\partial N_i}{\partial x}\end{bmatrix}\ . \tag{7}$$

4. Continuum-based Deformation Limiting

We want to enforce deformation limits component-wise, i.e. we assume a vector ε^{lim} holding a maximum allowable value

for every deformation mode. We can now pose the actual problem as follows: given an element's displacement vector $\mathbf{u}^e$ with a corresponding strain tensor $\varepsilon(\mathbf{u}^e)$ that violates one or more deformation limits , find a correcting velocity vector $\mathbf{v}^e$ such that $\varepsilon(\mathbf{u}^e + \Delta t \mathbf{v}^e)$ obeys the limit. Here, Δt is the step size and due to the linearity of the deformation we can write

$$\mathbf{B}(\Delta t \mathbf{v}^e) = \mathbf{b} \,, \tag{8}$$

which gives us three equations for the six unknown correcting velocity components. We want those components of the deformation that do not violate the limits to remain unchanged and define the right hand side $\mathbf{b}$ as

$$\mathbf{b}_i = \begin{cases} \varepsilon_i - \varepsilon_i^{lim} & \text{, if } \varepsilon_i > \varepsilon_i^{lim} \\ 0 & \text{, otherwise.} \end{cases} \tag{9}$$

Imposing deformation limits is not enough since the correcting velocity may change the linear and angular momentum of the element. Requiring $\mathbf{v}^e$ to not affect linear momentum yields two additional equations,

$$\sum_{i=1}^{3} m_i \mathbf{v}_{i,x} = 0 \qquad \text{and} \qquad \sum_{i=1}^{3} m_i \mathbf{v}_{i,y} = 0 \,, \tag{10}$$

where m_i are nodal masses. Finally, we obtain the sixth equation by enforcing $\mathbf{v}^e$ to not change rotational momentum: defining relative positions $\mathbf{l}_i = \mathbf{x}_i - \mathbf{x}_m$, where $\mathbf{x}_m$ is the center of mass of the element, we obtain a single equation

$$\sum_{i=1}^{3} m_i (\mathbf{l}_{i,x} \mathbf{v}_{i,y} - \mathbf{l}_{i,y} \mathbf{v}_{i,x}) = 0 \,. \tag{11}$$

We write the final system of equations as $\mathbf{A}\mathbf{v}^e = \mathbf{b}$, where $\mathbf{A}$ is a 6×6 matrix (see Appendix A). Given the frequency of the operation, the direct solution of a general 6×6 system is relatively expensive. However, many entries of $\mathbf{A}$ are rest state quantities and therefore stay constant during simulation. This motivates an approach based on Schur decomposition, which allows us to transfer most of these costs to precomputations (refer to Appendix A). In this way, the time for the solution of the linear system can be reduced by a factor of more than 10.

4.1. Integration

As shown in Algorithm 1, CSL can be integrated into any cloth simulator as a post-processing step to time integration and collision handling (see [BFA02]). The simulation module provides us with *candidate* positions (l.4), which are fed to CSL for deformation correction (l.5). The strain resolving velocities are used to recompute corrected nodal positions (l.6), which in turn are passed on to the collision handling engine. This yields collision resolving velocities as well as final nodal positions (l.7). Lastly, another CSL pass is applied to the final velocity in order to limit the strain rate (l.8). For this purpose, the same expressions as above can be used, replacing the strain ε with the rate of deformation $\dot{\varepsilon} = \mathbf{B}\mathbf{v}^e$.

Algorithm 1 CSL

1: //Simulation loop:
2: **for** ($t = 0$ to $t = t_{end}$) **do**
3: $(\mathbf{x}_0, \mathbf{v}_0)$ = getState(t);
4: $(\mathbf{x}_n^{cand}, \mathbf{v}_n^{cand})$ = step$(\mathbf{x}_0, \mathbf{v}_0)$;
5: $\mathbf{v}_n^{corr}$ = limitStrain$(\mathbf{x}_0, \mathbf{x}_n^{cand})$;
6: $\mathbf{x}_n^{corr} = \mathbf{x}_0 + \Delta t \mathbf{v}_n^{corr}$;
7: $(\mathbf{x}_n, \mathbf{v}_n^{coll})$ = handleCollisions$(\mathbf{x}_n^{corr}, \mathbf{v}_n^{corr})$;
8: $\mathbf{v}_n$ = limitStrainRate $(\mathbf{v}_n^{coll})$;
9: **end for**

Collision Handling In the presence of collisions, the final nodal positions (l.7) may violate deformation limits. It is not safe to correct the final positions since this might reintroduce collisions. We can, however, correct the strain rate for the collision resolving velocity in order to reduce the chance of encountering over deformed elements in the next time step. During this process, it is important to adhere to the collision resolving velocity for nodes in contact with rigid bodies since otherwise collisions in the next time step follow inevitably. We can put significant stress off the collision handling stage by treating velocities for these nodes as fixed such as to propagate the collision resolving velocities around the contact zones. This extension reduces the average number of over-deformed elements after collision handling and it allowed us to simulate even violent rigid body motions as in our second test scene (see Sec. 6).

5. Global Enforcement

In the same way as for edge-based approaches, corrected elements have to be combined into a global response. There are several alternatives to do so and we will consider three of them in this section.

Jacobi Iterative Enforcement Using the expressions from Sec. 4 we can correct all elements in isolation to obtain an intermediate state with a set of incompatible elements (see Fig. 4). A simple way to obtain a conforming configuration from incompatible elements is to use nodal averaging.

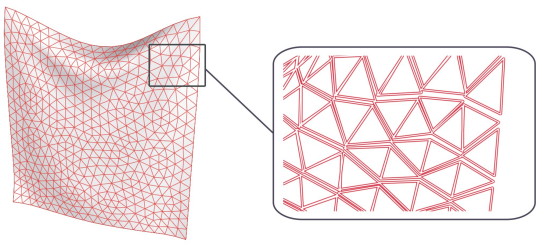

Figure 4: *Correcting elements in isolation leads to an intermediate state with incompatible elements* (right).

Let S be the set of indices denoting all elements which are incident to a given vertex $\mathbf{x}_i$ in the conforming configuration and let $\mathbf{x}_i^k$ with $k \in S$ be the corresponding vertex of element k in the incompatible configuration (see Fig. 5). The final

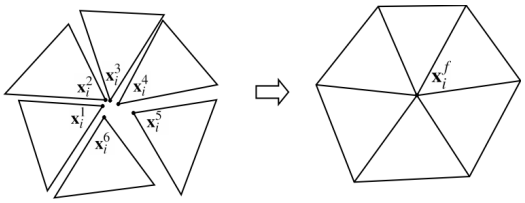

Figure 5: *Incompatible elements (*left*) are combined into a conforming configuration (*right*) using nodal averaging.*

position of $\mathbf{x}_i^f$ can be defined as the weighted average of the corresponding nodal positions in the incompatible configurations,

$$\mathbf{x}_i^f = \sum_{k|k\in S} \mathbf{x}_i^k w_i^k \quad \text{, where } \sum_i w_i^k = 1\,. \tag{12}$$

Here, w_i^k denote weights which can, e.g., be used to put more emphasis on contributions from elements with fixed vertices (other than $\mathbf{x}_i^k$) in order to accelerate convergence.

Using nodal averaging, an iterative method to globally enforce deformation limits proceeds as follows: correct elements in isolation, average nodes and reiterate this process until the deformation limit is met throughout the entire cloth. In analogy to the solution of linear systems of equations, this process can be described as Jacobi iterative enforcement, since element responses of one iteration become visible to other elements only in the subsequent iteration. As an advantage of this approach, the result is completely independent of mesh ordering. The number of necessary iterations depends on the problem under consideration, including the time step and the material used. Section 6 provides a comparative analysis of the convergence.

Gauss-Seidel Iterative Enforcement Instead of using an intermediate state, element responses can as well be committed immediately and thus have direct effect on neighboring elements in the same iteration. For edge-based strain limiting, this process, which can be considered as Gauss-Seidel iterative enforcement, has been reported to show faster convergence [BFA02]. This comes, however, at the price of no longer being independent of element ordering, which may lead to visible bias. This can be reduced using randomized element traversal and we did not encounter visible problems due to bias. The Gauss-Seidel method convergences faster, but the Jacobi variant lends itself more readily to a parallel implementation. We compare both methods in Sec. 6.

Direct Enforcement The Constraint Lagrangian Mechanics (CLM) formalism is commonly used to simulate mechanical systems subject to constraints. The latter are enforced using Lagrange multipliers, which give rise to constraint forces and augment the discretized equations of motion by additional degrees of freedom (see e.g. [MR02]). A solution to this nonlinear problem will satisfy physics and all (nonlinear) constraints simultaneously, but is generally expensive to compute. Goldenthal et al. [GHF+07] proposed a more efficient solution based on a fast manifold projection method and integrating it with CSL-constraints is a viable alternative. In any case, CLM methods require the computation of constraint gradients with respect to *3D* positions. This would degrade the efficiency of our formulation, which stems from a transformation of elemental strain limiting to *2D*: in this setting, large parts of the solution can be precomputed – an advantage that would be lost when computing the full 3D gradient. We will therefore not pursue this option further here.

Another approach for direct enforcement is to assemble all elemental problems into a global, overdetermined system and solve it in a least squares sense. This system has to be solved in every time step using, e.g., singular value decomposition [PTVF07]. With increasing problem size, however, this soon becomes prohibitively expensive.

Tsiknis [Tsi06] suggested a similar approach in which corrected triangles are *stitched* together by minimizing the sum of squared differences between edge vectors in the incompatible and conforming configuration. An appealing feature of his method is that it can be posed as a linear least squares problem with a matrix staying constant during simulation. However, the norm which is used in minimization leads to two problems. First, it is *not invariant* under edge rotations although these do not affect deformation. This reduces the possibilities of finding a good conforming state. Second, the norm *is invariant* under translations. This can lead from significant artificial damping up to motion locking since the previous time step's solution (satisfying deformation limits) is always a solution to the current minimization problem.

In conclusion, Jacobi and Gauss-Seidel iterative enforcement seem most attractive in our case. The Gauss-Seidel variant is slightly more efficient, but the Jacobi variant has a greater potential for parallelization and is independent of mesh ordering.

6. Results

This section presents results obtained with our new deformation limiting method and compares its performance to previous approaches. For our experiments, we use a cloth simulator based on continuum mechanics and finite elements. As one of several advantages over mass-spring systems, this approach is more accurate since it allows to explicitly model stretching, shearing and transverse contraction. We use the first order implicit Euler integration scheme with a time step of $0.001s$. In the following, deformation limits refer to maximum tensile strain but we found it useful to also limit the

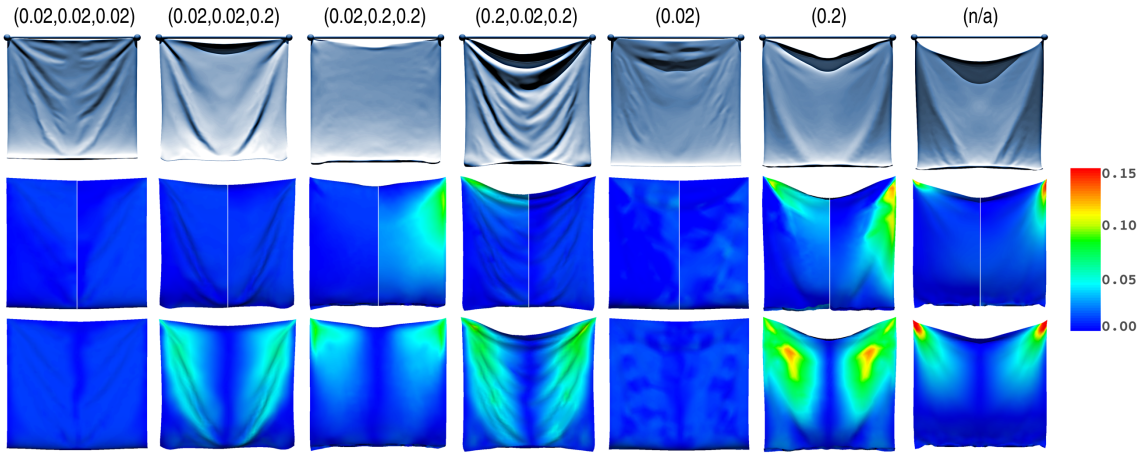

Figure 6: *Representative frame from example 1, which compares CSL (column 1–4), ESL(colums 5–6) and no strain limiting (column 7) using different deformation constraints as indicated above each column. The top row shows a rendered view, the middle row shows visualizations of weft and warp deformation in split images while the last row depicts shear deformation. The material parameters were set to (100,100,30) N/m for all cases. Weft and warp directions coincide with the horizontal and vertical axis of the image plane, respectively.*

maximum compression of the elements. This forces the cloth to buckle and to produce folds in places where compression would have been observed otherwise. All examples were performed on a standard PC with two 2.0 GHz dual core CPUs. Unless stated otherwise, only a single core was used.

In our first example we investigate continuum-based strain limiting (CSL) with anisotropic deformation constraints and compare the results to the edge-based variant (ESL). In this scene, a square, irregularly tesselated mesh with 3616 faces is pinned at two corners and swings under the influence of gravity. We impose deformation constraints with a strictness varying from 0.02 to 0.2. Fig. 6 shows a rendered frame along with strain distributions for the cases studied. Only a single constraint value can be used for ESL (columns 5–6), but with CSL we can additionally experiment with differently strict constraints for each deformation mode (columns 1–4). In order to best isolate the effect of strain limiting, material parameters were set to (100,100,30) N/m in all cases. Using CSL, substantially different material behaviors are obtained just by switching anisotropic constraints along weft and warp direction (compare columns 3 and 4). Likewise, setting strict constraints for weft and warp but a weak constraint for shear deformation gives completely different though realistic results. Relaxing the single constraint for ESL from 0.02 to 0.2 (columns 5 and 6), the fabric becomes globally softer and resembles the unconstrained case (column 7). The strain visualizations in rows 2 and 3 reveal that in all cases deformation constraints are accurately observed. The plots also show that CSL achieves a clean separation between the different deformation modes. In particular, we can observe higher deformation for strain components with softer constraints. Finally, the strain distribution for ESL is not as smooth as for CSL. This can be attributed to the fact that some edges are well aligned with the material directions while others are not.

In a similar scene (Fig. 8) we additionally investigated the influence of external collisions, leading to different deformation patterns. Again, differences between continuum- and edge-based strain limiting are clearly perceptible, emphasizing the capability of CSL to simulate a much broader range of material behavior.

The second example is a practical cloth animation, including rapid rigid body motions which lead to challenging cloth-object and self collisions. A rigid bar is placed between the straps of a dress comprised of 8891 faces. A soft biphasic material (300,300,75) N/m with low stretch but high shear limits (0.1,0.1,0.4) provides the necessary deformation freedom for the dress to fold and buckle into complex shapes. Under the influence of gravity, the dress starts to drape but is soon accelerated forwards by the abrupt motion of the bar, moving back and forth. This motion leads to high velocity impact collisions which are challenging for both strain limiting and collision handling. Subsequently, the bar is rotated with increasing angular velocity such as to literally spin and swirl the dress around itself (see Fig. 7, left). Complicated self collisions result in multiple twisted layers and provoke large shear deformations. Again, the difference between continuum- and edge-based deformation limiting can be seen very clearly. The edge-based scheme cannot satisfy stretch constraints and allow large shear deformations at the same time (see Fig. 7, right). Using CSL, larger shear deformations can result in accordance to the material law (Fig. 7, middle). For example 2 (see Figs 1 and 7), time integration took about 16.09 minutes for 5 seconds of simulation.

The average iteration count for CSL was 14.92 and the corresponding computation time was 12.19 minutes, due to the advanced degree of complexity that this scene represents.

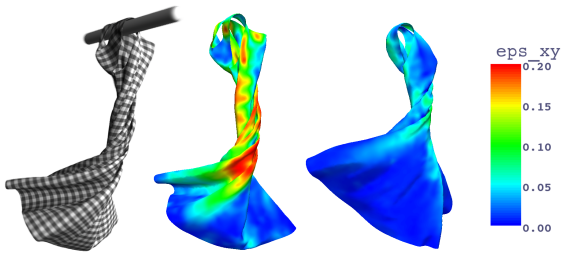

Figure 7: *Simulation snapshot (*left*) and shear strain plots using CSL (*middle*) and ESL (*right*). Using CSL, larger shear strains can be allowed while enforcing low stretch deformation. ESL can only enforce a single limit on stretch and shear deformation, leading to substantially different results.*

Finally, we compare the computational costs and convergence properties of CSL and ESL. The setup is the same as in example 1, but a set of meshes with increasing resolution was used. In order to facilitate comparison to the edge-based variant, an isotropic material with material coefficients of (500,500,250) N/m was used. Deformation limits were set to (0.1,0.1,0.2) for CSL and 0.1 for ESL. Table 1 provides iteration counts and run times for 1 second of simulation.

	226 faces			904 faces			3616 faces		
	#it	t_sl	t_int	#it	t_sl	t_int	#it	t_sl	t_int
CSL-GS	1.00	0.34	3.24	1.04	1.39	14.14	3.81	18.98	77.29
CSL-JAC-1	1.00	0.47	3.32	1.04	1.85	14.11	5.45	36.90	73.20
CSL-JAC-4	1.00	0.26	3.51	1.04	0.90	14.04	5.45	15.88	76.01
ESL	1.00	0.13	3.35	1.05	0.62	13.99	3.37	4.87	73.85

Table 1: *Iteration counts and computation times in seconds for example 3. #it denotes the average number iterations to convergence for strain limiting and t_sl the corresponding computation times. t_int lists time spent on time integration.*

As expected, the Gauss-Seidel variant CSL-GS is slightly faster than the single-threaded Jacobi variant CSL-JAC-1. However, using 4 threads on 4 CPUs, the parallel Jacobi variant (CSL-JAC-4) it is faster than CSL-GS. The computational costs for the CSL variants are higher than for ESL, but given the improved simulation results, this difference seems acceptable.

7. Conclusion

The efficient simulation of general biphasic textiles and their anisotropic behavior is a challenging topic. We have shown that traditional edge-based deformation limiting is unsuitable for a faithful simulation of these materials. By contrast, our new method allows accurate control of deformation and enables the use of individual thresholds. This is a vital ingredient for simulating materials which, e.g., exhibit a soft resistance to shear deformation, but allow only small stretch deformations (see Fig 7). Due to its simplicity and computational efficiency, CSL is an attractive alternative to use in standard collision handling frameworks.

Like edge-based approaches [Pro95, BFA02], our method relies on iterative constraint enforcement. As such, it inherits the drawbacks with respect to performance scaling with increasing mesh sizes, discussed in [GHF+07]. An interesting direction for future work would therefore be to explore ways to improve the asymptotic convergence behavior, e.g., using multi-resolution techniques.

8. Acknowledgments

We would like to thank Rony Goldenthal and Eitan Grinspun for inspiring discussions and valuable comments. We also thank the reviewers for their helpful suggestions and ideas.

Appendix A: Linear system and its solution

The 6×6 matrix $\mathbf{A}$ for the strain limiting problem reads

$$\mathbf{A} = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} \\ \frac{1}{2}\frac{\partial N_1}{\partial y} & \frac{1}{2}\frac{\partial N_1}{\partial x} & \frac{1}{2}\frac{\partial N_2}{\partial y} & \frac{1}{2}\frac{\partial N_2}{\partial x} & \frac{1}{2}\frac{\partial N_3}{\partial y} & \frac{1}{2}\frac{\partial N_3}{\partial x} \\ m_1 & 0 & m_2 & 0 & m_3 & 0 \\ 0 & m_1 & 0 & m_2 & 0 & m_3 \\ -m_1\mathbf{l}_{1,y} & m_1\mathbf{l}_{1,x} & -m_2\mathbf{l}_{2,y} & m_2\mathbf{l}_{2,x} & -m_3\mathbf{l}_{3,y} & m_3\mathbf{l}_{3,x} \end{bmatrix}$$

The computational cost for the solution of the associated linear system can be significantly reduced using the observation that, except for the last row, all entries of $\mathbf{A}$ are rest state quantities. We first decompose the linear system and write

$$\begin{bmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{D} & \mathbf{E} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \end{bmatrix} . \tag{13}$$

Here, $\mathbf{B}$ is upper-left 5×5 sub-matrix of $\mathbf{A}$ and $[\mathbf{s}_1\mathbf{s}_2]^t$ and $[\mathbf{r}_1\mathbf{r}_2]^t$ are partitioned solution and right hand side vectors, respectively. Using the *Schur complement* $\mathbf{S} = \mathbf{E} - \mathbf{D}\mathbf{B}^{-1}\mathbf{C}$, the solution of (13) is recast into two steps:

$$\begin{aligned} \mathbf{s}_2 &= \mathbf{S}^{-1}(\mathbf{r}_2 - \mathbf{D}\mathbf{B}^{-1}\mathbf{r}_1) , \\ \mathbf{s}_1 &= \mathbf{B}^{-1}(\mathbf{r}_1 - \mathbf{D}\mathbf{s}_2) . \end{aligned}$$

Since $\mathbf{S}$ is a scalar, its inversion is trivial. Computing the 5×5 inverse matrix $\mathbf{B}^{-1}$ is relatively expensive, but since it only depends on rest state quantities it can be precomputed. Using an optimized code, only 1 division, 33 additions and 37 multiplications have to be performed at run time to solve (13). Compared to the solution via direct inversion of $\mathbf{A}$ as described in Section. 4, the operation count is thus reduced by a factor of more than 13. Note that in the presence of fixed vertices the system has to be modified. We transfer

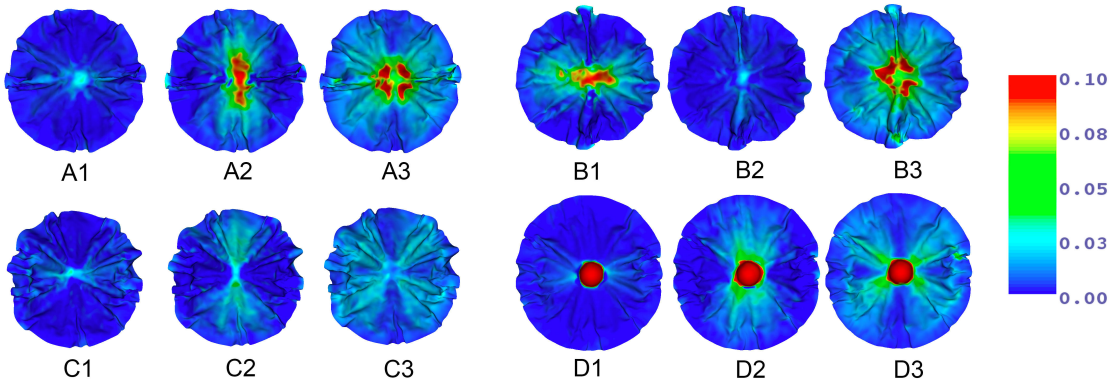

Figure 8: *Strain plots for weft, warp and shear strain (left to right) in groups of three. An irregular mesh (4424 faces) with radius 0.5m is draped over a small sphere. Material properties of (1000,200,100) N/m were used for weft, warp and shear directions. Deformation limits were set to (0.04,0.2,0.4) for CSL and 0.04 for ESL.* Top left*: CSL. In accordance to the anisotropic limits, low deformation in the weft direction is accurately observed (A1), while the more generous limits for warp and shear directly translate into larger strains in these directions (A2-A3).* Bottom left*: ESL. Strain limits are observed but, in contrary to the material properties, deformation is completely isotropic (C1-C3).* Bottom right*: Using no strain limiting, deformation is distributed inhomogeneously and bounds are exceeded by up to a factor of 5 (D1-D3).* Top right*: CSL. Same as top left except for switched material properties and limits in weft and warp directions. Results are consistent with A1-A3.*

known quantities to the right hand side and solve the resulting overdetermined system in a least squares sense using singular value decomposition.

References

[BFA02] R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. In *Proceedings of ACM SIGGRAPH '02*, pages 594–603, 2002.

[BW98] D. Baraff and A. Witkin. Large steps in cloth simulation. In *Proceedings of ACM SIGGRAPH '98*, pages 43–54, 1998.

[BWR+08] M. Bergou, M. Wardetzky, S. Robinson, B. Audoly, and E. Grinspun. Discrete elastic rods. *ACM Trans. Graph.*, 27(3):1–12, 2008.

[CK02] K.-J. Choi and H.-S. Ko. Stable but responsive cloth. In *Proceedings of ACM SIGGRAPH '02*, pages 604–611, 2002.

[EB08] E. English and R. Bridson. Animating developable surfaces using nonconforming elements. In *Proceedings of ACM SIGGRAPH '08*, pages 1–5. ACM, 2008.

[EKS03] O. Etzmuss, M. Keckeisen, and W. Straßer. A Fast Finite Element Solution for Cloth Modelling. In *Proceedings of Pacific Graphics*, pages 244–251, 2003.

[GHF+07] R. Goldenthal, D. Harmon, R. Fattal, M. Bercovier, and E. Grinspun. Efficient simulation of inextensible cloth. In *Proceedings of ACM SIGGRAPH '07*, pages 281–290, 2007.

[HCJ+05] M. Hong, M.-H. Choi, S. Jung, S. Welch, and J. Trapp. Effective constrained dynamic simulation using implicit constraint enforcement. In *Proceedings of International Conference on Robotics and Automation*, pages 4520–4525. IEEE, 2005.

[HE01] M. Hauth and O. Etzmuss. A high performance solver for the animation of deformable objects using advanced numerical methods. In *EG 2001 Proceedings*, pages 319–328. Blackwell Publishing, 2001.

[HS04] M. Hauth and W. Straßer. Corotational simulation of deformable solids. *Journal of WSCG*, 12(1):137–145, 2004.

[MDM+02] M. Müller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler. Stable real-time deformations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation (SCA 2002)*, pages 49–54, 2002.

[MHHR06] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff. Position based dynamics. In *Virtual Reality Interactions and Physical Simulation (VRIPHYS)*, pages 71–80, 2006.

[MR02] J. Marsden and T. Ratiu. *Introduction to Mechanics and Symmetry*. Springer, 2nd edition, 2002.

[Mül08] M. Müller. Hierarchical position based dynamics. In *Virtual Reality Interactions and Physical Simulation (VRIPHYS)*, pages 1–10, 2008.

[Pro95] X. Provot. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *Graphics Interface '95*, pages 147–154. Canadian Human-Computer Communications Society, 1995.

[PTVF07] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes: The art of scientific computing*. Cambridge University Press, 3rd edition, 2007.

[TPBF87] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. In *Proceedings of ACM SIGGRAPH '87*, pages 205–214, 1987.

[Tsi06] K. D. Tsiknis. Better cloth through unbiased strain limiting and physics-aware subdivision. Master's thesis, The University of British Columbia, 2006.

[VCMT95] P. Volino, M. Courchesne, and N. Magnenat-Thalmann. Versatile and efficient techniques for simulating cloth and other deformable objects. In *Proceedings of ACM SIGGRAPH '95*, pages 137–144, 1995.

[ZT00] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method. Volume 1: The Basis*. Butterworth Heinemann, 5th edition, 2000.