Yarn-Level Simulation of Woven Cloth

Gabriel Cirio Jorge Lopez-Moreno David Miraut Miguel A. Otaduy

URJC Madrid

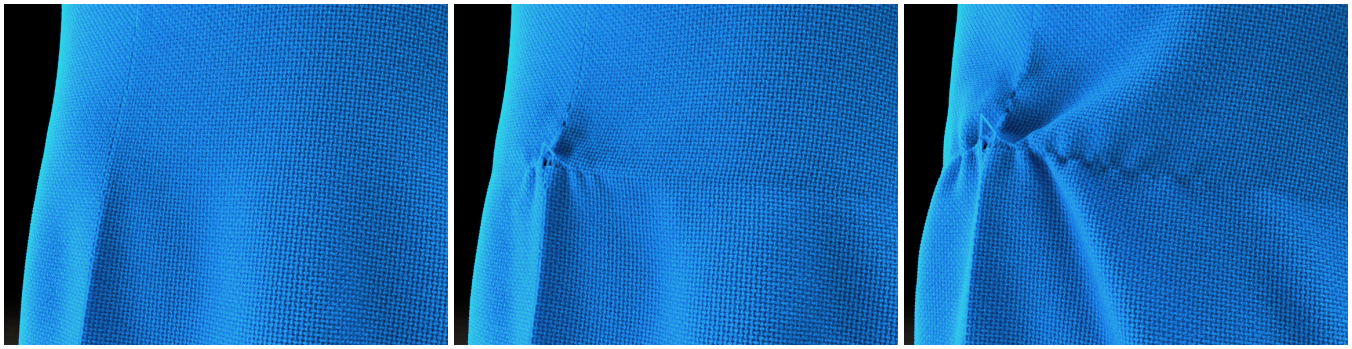

Figure 1: *Yarn-level simulation of a shirt with* 2023 *yarns and* 350530 *crossing nodes. We produce a snag on the shirt by pulling on a seam node. Fine-scale deformations showing yarn sliding and thin wrinkles are combined with large-scale motion of the shirt.*

Abstract

The large-scale mechanical behavior of woven cloth is determined by the mechanical properties of the yarns, the weave pattern, and frictional contact between yarns. Using standard simulation methods for elastic rod models and yarn-yarn contact handling, the simulation of woven garments at realistic yarn densities is deemed intractable. This paper introduces an efficient solution for simulating woven cloth at the yarn level. Central to our solution is a novel discretization of interlaced yarns based on yarn crossings and yarn sliding, which allows modeling yarn-yarn contact implicitly, avoiding contact handling at yarn crossings altogether. Combined with models for internal yarn forces and inter-yarn frictional contact, as well as a massively parallel solver, we are able to simulate garments with hundreds of thousands of yarn crossings at practical framerates on a desktop machine, showing combinations of large-scale and fine-scale effects induced by yarn-level mechanics.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling

Keywords: Cloth, Yarns, Physically based simulation

Links: DL PDF

1 Introduction

Woven cloth is formed by interlacing yarns, typically two sets of orthogonal yarns called warp and weft. Interlaced yarns undergo friction forces at yarn-yarn contacts, and this friction holds together the woven fabric, in contrast to knitted fabrics, which are held together by stitching yarns. Woven cloth is ubiquitous, and it exhibits diverse weave patterns and yarn materials, both stiff and elastic. Common woven fabrics include chiffon, corduroy, denim, flannel, gabardine, sheeting, or velvet.

Large-scale mechanics of woven cloth are dictated by the fine-scale behavior of yarns, their mechanical properties, arrangement, and contact interactions. However, popular cloth models, with the notable exception of the work of Kaldor et al. [2008; 2010], do not model yarn-level mechanics. They use either discrete elements, as in the case of mass-spring systems [Breen et al. 1994; Provot 1995], or discretizations of continuum formulations, as in the case of finite-element models [Etzmuss et al. 2003].

Such discretized models are often sufficient for capturing relevant behavior of woven cloth, in particular draping under moderate forces. But yarn-level models introduce exciting possibilities for computer animation. Visually interesting effects such as detailed tearing, snags, or loose yarn ends require modeling individual yarns. Moreover, yarn-based models can constitute the cornerstone to develop accurate solutions to large-scale cloth simulation, revealing the nonlinearities and complex interplays measured in real fabrics [Wang et al. 2011; Miguel et al. 2012; Miguel et al. 2013].

Computational cost has been the key challenge to address yarn-level cloth simulation. Capturing the mechanics of individual yarns requires the use of rod models [Pai 2002; Spillmann and Teschner 2009; Bergou et al. 2008; Casati and Bertails-Descoubes 2013], and weave patterns produce a number of contacts that is quadratic in the number of yarns. Modeling even low yarn-density fabrics soon leads to an explosion in the number of DoFs and contacts, and common fabrics may contain in the order of 100 yarns/inch.

In this paper, we present a novel model of woven cloth that enables efficient yarn-level simulation. The key aspect of our model is a discretization focused on yarn crossings (Fig. 2). It consists of the 3D position of the crossing point plus two additional degrees of freedom to capture yarn sliding, following the Eulerian rod discretization of Sueda et al. [2011]. Inter-yarn contact is handled implicitly, and we avoid altogether the computation of collision detection and collision response between crossing yarns.

Based on our discretization, we formulate force models for low-

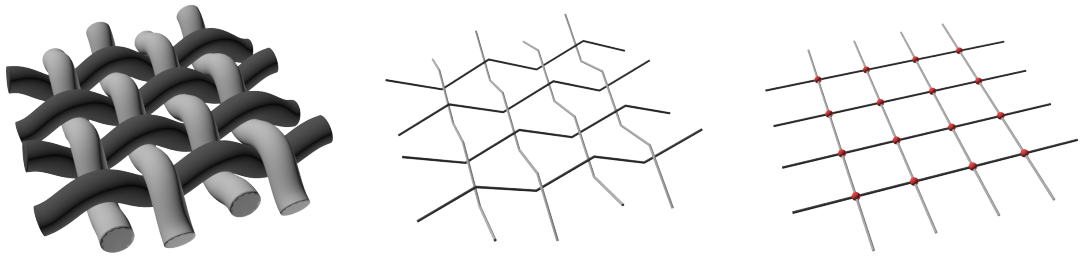

Figure 2: *Models of woven yarns used in this work. Warp yarns are in black, weft yarns in white. The 3D volumetric yarns (left) of a piece of fabric are replaced by interlaced rod segments with crimp (middle) for normal force compuation, and by rod segments crossing at 5-DoF crossing nodes (right, crossing nodes in red) for everything else. The volumetric appearance is restored at rendering.*

level yarn mechanics. These include stretch and bending forces of individual yarns. But, most importantly, our novel discretization enables simple formulations of inter-yarn contact forces, in particular sliding friction at yarn crossings and contact between adjacent parallel yarns. Interesting effects, such as plasticity at yarn level (Fig. 1) or the influence of the weave pattern on large-scale behavior (Fig. 3) are obtained naturally thanks to yarn-level mechanics.

To robustly simulate yarn-level clothing, we apply implicit integration to the dynamics equations. We have designed a massively parallel solver that leverages the weave pattern as well as our novel discretization. With our GPU implementation, we simulate clothing with over 300K nodes and 2K yarns at just over 2min/frame (where one frame is 1/24th of a second) on a desktop machine.

2 Related Work

Cloth Simulation: Most cloth simulation models in computer graphics consider cloth as a thin shell and formulate an elastic deformation model to capture its mechanics [Terzopoulos et al. 1987]. Then, cloth modeling faces the challenge of defining deformation energies and discretizations that are numerically robust and match the behavior of real cloth. Some key milestones in cloth modeling in computer graphics include: mass-spring models that approximate the behavior of real woven fabrics [Breen et al. 1994], the addition of strain limiting to model inextensibiliy [Provot 1995], efficient handling of self-collisions [Volino et al. 1995], definition of deformation energies from constraints with efficient time integration [Baraff and Witkin 1998], robust models to handle buckling [Choi and Ko 2002], consistent bending models [Bridson et al. 2003; Grinspun et al. 2003], efficient inextensibility [Goldenthal et al. 2007], and efficient dynamic remeshing [Narain et al. 2012].

Recent work in computer animation has also aimed to match the nonlinear behavior in real cloth. Relevant works include the design of nonlinear parametric models [Volino et al. 2009], estimation of material coefficients from force and deformation examples [Wang et al. 2011; Miguel et al. 2012], and design of internal friction models to capture cloth hysteresis [Miguel et al. 2013].

In contrast to popular thin shell models, Kaldor et al. [2008] modeled the dynamics of knitted cloth at the yarn level, allowing them to predict the large-scale behavior of full garments from fundamental yarn mechanics. They captured the mechanics of individual yarns using an inextensible rod model, and yarn-yarn contact with a combination of stiff penalty forces and velocity-filter friction. Later in [2010], they extended their work to accelerate yarn-yarn contact handling, by using local rotated linearizations of penalty forces. We present a more efficient solution for the case of woven cloth that avoids altogether handling yarn-yarn contact at yarn crossings. Metaaphanon et al. [2009] proposed a yarn-level model for woven cloth. They modeled yarn-yarn interaction by setting constraints between the end points of warp and weft springs. In addition, they designed an automatic transition from a mass-spring model to the yarn-level model.

Yarn-Level Models in Textile Research: Yarn-level models have been thoroughly studied in the field of textile research. Yarn-based analytical models [Hearle et al. 1969] were used to predict the mechanical behavior of fabric under specific modes of deformation, usually based on geometric yarn models. These analytical models, such as Peirce's parametric circular cross-section yarns [Peirce 1937] or Kawabata's much simpler pin-joined trusses [Kawabata et al. 1973], model yarns at crossover points assuming persistent contact and accounting for crimp separation. However, as for most analytical models these approaches are limited to the specific cases they were designed for, and developing an analytical framework for general load cases would be extremely complex [King et al. 2005], let alone entire garments.

Mesostructure-based continuum models emerged to simulate larger fabric samples [Boisse et al. 1997; Parsons et al. 2010]. These models approximate woven fabric as a continuum, where every material point represents a section of yarns. Each section is then simulated using a greatly simplified analytic unit cell employing, for instance, Kawabata's pin-joined truss model.

Another family of models attempts to simulate the full fabric at yarn level using finite element discretizations of volumetric yarns, accounting for all yarn interactions [Ng et al. 1998; Page and Wang 2000; Duan et al. 2006]. However, the large computational requirements make them intractable for moderately large samples. Greater computational efficiency was achieved by replacing the complex volumetric yarns by simpler elements such as beams, trusses and membranes [Reese 2003; McGlockton et al. 2003], similiar to our work where yarns are modeled as rods. Another interesting approach is to resort to costly yarn-level mechanics only when needed, using multiscale models that couple continuum and yarn-level descriptions [Nadler et al. 2006].

Somewhat hybrid techniques rely on mesostructure-based continuum approaches, but using a discrete model for unit cells. These cells allow axial compliance and can be augmented with bending and crossover springs to simulate cross-sectional deformation and shear at crossover points [King et al. 2005; Xia and Nadler 2011].Shear jamming is achieved by introducing truss elements normal to the yarns to simulate contact forces between the yarns [King et al. 2005]. However, since yarns are pinned together at crossover points, these unit-cell approaches prevent yarn sliding. Parsons and collaborators [2013] address yarn sliding by introducing a slip velocity field at the continuum level, with forces computed at meso-level using the unit cell. Slippage friction forces are proportional to the normal forces at the crossover points. However, these approaches usually do not simulate every yarn in the fabric,

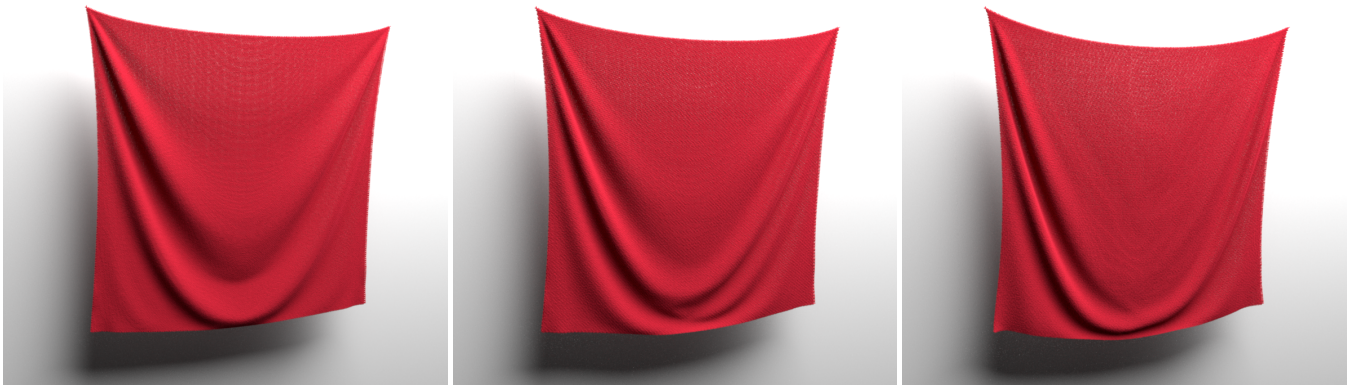

Figure 3: *From left to right, weaves with more floats: plain, twill, and satin. More floats lead to lower shear resistance, hence cloth falls lower when pinned from two corners.*

thus preventing interesting single yarn effects such as snags, frayed edges, yarn fracture and yarn pullouts. In addition, typical yarn-level models in textile research assume persistent contact between woven yarns, but they do not resolve yarn positions under free garment motions, only controlled experiments. Our approach allows to simulate every yarn in the fabric as a rod, while greatly reducing costly contact interactions by making contact persistent and introducing additional sliding degrees of freedom.

Rod Models and Contact: An essential aspect of yarn-level simulation is the choice of rod model to capture the mechanics of individual yarns. Pai [2002] developed an efficient algorithm to simulate rods modeled following Cosserat theory. Spillmann and Teschner [2007] improved on Cosserat models to handle contact efficiently, and later in [2009] they extended them to handle branched and looped structures. Bergou et al. [2008] presented an approach for rod simulation that decouples centerline dynamics from a quasi-static solution of twist based on parallel transport. Casati and Bertails-Descoubes [2013] have recently evolved clothoid-based models to efficiently resolve the dynamics of rich and smooth rods with very few control points.

As outlined in the introduction, the major challenge in modeling cloth at the yarn level is efficient contact handling between yarns. Sueda et al. [2011] presented a model suited for simulating efficiently highly constrained rods. The key insight of their model is to describe the kinematics of constrained rods using an optimal set of generalized coordinates, formed by so-called Lagrangian coordinates that capture absolute motion, and so-called Eulerian coordinates that capture sliding on constraint manifolds. We find that this approach fits nicely for representing constrained yarns in woven cloth, and we design a discretization for a case not handled by Sueda et al., consisting of two rods in sliding contact. Finally, hair contact is a more general problem involving constrained rods, and solutions designed for this problem [Daviet et al. 2011] could be applied to frayed edges in yarn-level cloth.

3 Discretization Based on Yarn Crossings

We start this section describing how we construct cloth models based on yarns. Next, we present the main ingredient of our yarn-based cloth model: the discretization of yarn kinematics based on the positions of yarn crossing points and yarn sliding. To conclude the section, we outline the formulation of the equations of motion based on this discretization.

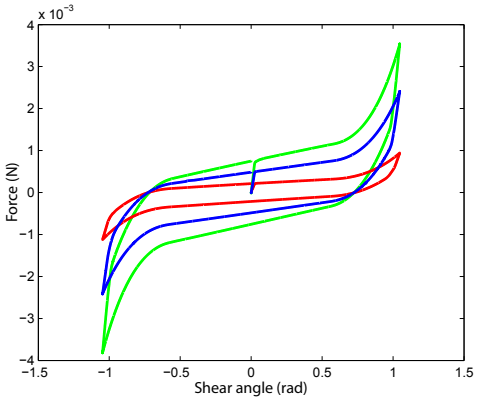

Figure 4: *Force (in N) with respect to shear angle (in radians) for the 3 hanging sheet examples (plain weave in green, twill in blue, satin in red). Shear force, friction and jamming are clearly visible, as well as the different behaviors according to the weave pattern.*

3.1 Yarn-Based Woven Garments

To construct garments at the yarn level, we follow a tailoring approach. We take as input the 2D pattern that forms a garment, and we lay warp and weft yarns as orthogonal straight lines on each 2D panel independently, at an inter-yarn distance L. At each seam we place an additional yarn, and weft and warp yarns are connected to seam yarns by sharing nodes. At cloth boundaries we choose between adding seam yarns or letting yarn endings hang freely. 3D cloth models can be obtained automatically from commercial patterns [Berthouzoz et al. 2013], hence our modeling process can also be easily automated.

A *float* constitutes a gap between two yarns of the same type where the other yarn is not interlaced. Different weave patterns, such as plain weave (with no floats), twill, satin, etc. are obtained by varying the distribution of floats, thereby affecting the mechanics of the resulting fabric as shown in Fig. 3 and Fig. 4. To model the weave pattern, we choose an arbitrary orientation for each panel, and we simply store at each yarn crossing a flag indicating which yarn, warp or weft, is on top. This simple strategy allows us to model plain weave, twill, satin, and all other common weaves.

In our kinematic representation we ignore the volume of yarns, and all yarns are initialized flat on the same plane. However, for the

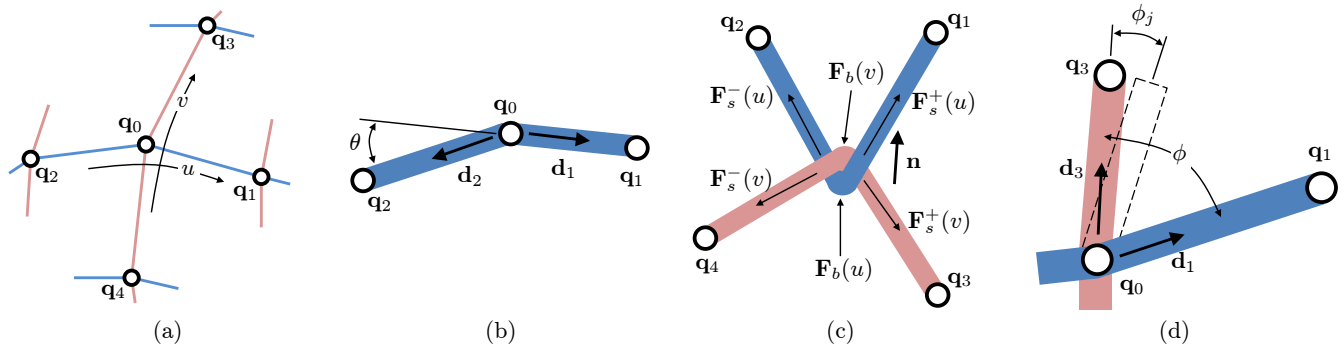

Figure 5: *a: Warp (u) and weft (v) yarns crossing at node $\mathbf{q}_0$, and the four adjacent yarn crossings. b: Bending angle θ between two adjacent warp segments. c: Forces producing normal compression at a crossing node. Subscripts s and b denote stretch and bending; superscripts $+$ and $-$ denote positive and negative yarn directions. d: Shear angle ϕ and shear jamming angle ϕ_j between two adjacent warp and weft yarns.*

purpose of force computation and rendering we account for the volume of yarns. *Crimp* is the bending introduced in warp and/or weft yarns to allow interlacing, as shown in Fig. 2 (left). Crimp also produces compressive forces between interlaced yarns, and this compression allows the existence of friction forces that hold the fabric together. In our implementation, we apply crimp to both weft and warp yarns, offsetting them by the yarn radius R in opposite directions, as shown in Fig. 2 (middle). Our implementation could be extended to allow for anisotropic crimp.

For ease of presentation, in the writing we assume that inter-yarn distance L and yarn radius R are the same for warp and weft, but it is trivial to relax this assumption, and indeed our implementation supports anisotropic cloth.

3.2 Discretization and Kinematics

In woven cloth, the vast majority of yarns are in contact at yarn crossings, and we can generally assume that such contacts are maintained throughout the simulation. The motion of cloth could be described as a constrained dynamics problem, with a node-based discretization of yarns, plus a large number of contact constraints that maintain a zero-distance between yarns at yarn crossings. But we observe that, instead of detecting and resolving such contacts, it is utterly more efficient to choose a convenient discretization based on yarn crossings.

We parameterize warp and weft yarns based on their rest arc length, u and v respectively. Then, we describe a yarn crossing by its 3D position, $\mathbf{x}$, and the parametric coordinates of the warp and weft material points at the yarn crossing. The variation of u and v coordinates models, respectively, the sliding of warp and weft yarns. Following the classification of Sueda et al. [2011], a yarn crossing is a 5-DoF node with 3 Lagrangian DoFs and 2 Eulerian DoFs. We denote the $\mathbb{R}^5$ coordinates of the i^{th} yarn crossing node as $\mathbf{q}_i \equiv (\mathbf{x}_i, u_i, v_i)$.

We propose to discretize woven cloth with a combination of 5-DoF yarn-crossing nodes and regular 3-DoF nodes. We set a 5-DoF node at each yarn crossing, and regular 3-DoF nodes at the end-points of yarns, as well as between yarn crossings, when the inter-crossing distance exceeds a predefined threshold. Fig. 5-a shows a regular set-up with a yarn-crossing node and its 4 adjacent nodes.

The derivation of kinematics and dynamics involving yarn-crossing nodes follows the theory proposed by Sueda et al. Given a warp segment $[\mathbf{q}_0, \mathbf{q}_1]$ (and similarly for weft segments), we linearly interpolate positions according to the arc length u. Then, the 3D position of a point inside the segment is given by:

$$\mathbf{x}(u) = \frac{u_1 - u}{\Delta u}\,\mathbf{x}_0 + \frac{u - u_0}{\Delta u}\,\mathbf{x}_1, \tag{1}$$

where $\Delta u = u_1 - u_0$ is the rest length of the segment.

The velocity of a point inside the segment depends on the velocities of yarn crossing points, but also on yarn sliding, and it follows by differentiating (1):

$$\dot{\mathbf{x}}(u) = \frac{u_1 - u}{\Delta u}\,(\dot{\mathbf{x}}_0 - \dot{u}_0\,\mathbf{w}) + \frac{u - u_0}{\Delta u}\,(\dot{\mathbf{x}}_1 - \dot{u}_1\,\mathbf{w}), \tag{2}$$

where $\mathbf{w} = \frac{\mathbf{x}_1 - \mathbf{x}_0}{\Delta u}$.

3.3 Lagrangian Mechanics

By concatenating the coordinates of all yarn crossings, we define a vector of generalized coordinates $\mathbf{q}$. We can then derive the equations of motion using the Lagrange-Euler equations [Goldstein et al. 2001], as suggested also by Sueda et al. The kinetic energy is $T = 1/2\,\dot{\mathbf{q}}^T\,\mathbf{M}\,\dot{\mathbf{q}}$, with a generalized mass matrix $\mathbf{M}$, V denotes the potential energy, and ∇ the generalized gradient. Then, the Euler-Lagrange equations can be written as

$$\mathbf{M}\,\ddot{\mathbf{q}} = \nabla T - \nabla V - \dot{\mathbf{M}}\,\dot{\mathbf{q}}. \tag{3}$$

We assume that mass is distributed uniformly along yarns, with density ρ. Then, following the definition (2) of velocity for an arbitrary point in a warp segment, the kinetic energy of the segment $[\mathbf{q}_0, \mathbf{q}_1]$ (and similarly for a weft segment) is:

$$T_{0,1} = \frac{1}{2}\begin{pmatrix}\dot{\mathbf{x}}_0^T & \dot{u}_0 & \dot{\mathbf{x}}_1^T & \dot{u}_1\end{pmatrix}\mathbf{M}_{0,1}\begin{pmatrix}\dot{\mathbf{x}}_0\\ \dot{u}_0\\ \dot{\mathbf{x}}_1\\ \dot{u}_1\end{pmatrix}, \quad \text{with} \tag{4}$$

$$\mathbf{M}_{0,1} = \frac{1}{6}\,\rho\,\Delta u\begin{pmatrix} 2\,\mathbf{I}_3 & -2\,\mathbf{w} & \mathbf{I}_3 & -\mathbf{w}\\ -2\,\mathbf{w}^T & 2\,\mathbf{w}^T\mathbf{w} & -\mathbf{w}^T & \mathbf{w}^T\mathbf{w}\\ \mathbf{I}_3 & -\mathbf{w} & 2\,\mathbf{I}_3 & -2\,\mathbf{w}\\ -\mathbf{w}^T & \mathbf{w}^T\mathbf{w} & -2\,\mathbf{w}^T & 2\,\mathbf{w}^T\mathbf{w}\end{pmatrix}. \tag{5}$$

The potential energy V includes multiple terms, such as gravity and conservative internal forces. Gravity is defined, e.g., for the warp segment $[\mathbf{q}_0, \mathbf{q}_1]$ as

$$V_{0,1} = \rho\,\Delta u\,\mathbf{g}^T\,\frac{\mathbf{x}_0 + \mathbf{x}_1}{2}. \tag{6}$$

The formulation of internal forces is discussed in detail in the next section. In addition to the conservative forces derived from energy potentials, we also incorporate other force terms directly to the right-hand side of the Euler-Lagrange equations (3), such as friction and contact forces. We also incorporate damping through the Rayleigh damping model, with mass and stiffness-proportional terms controlled by parameters α and β respectively.

4 Force Model

We consider two types of internal forces in woven cloth. This section starts with the description of forces due to the deformation of individual yarns, which include stretch and bending forces. We do not consider yarn torsion, as its effect is minimal on cloth. Next, we describe internal forces due to contact between interlaced yarns, which include normal compression, sliding friction, shear, and contact between parallel yarns.

We describe conservative forces in a concise manner using energy potentials. In the general case, these potentials will produce forces on both the yarn crossing points and the sliding coordinates. In addition, the application of numerical integration requires the computation of force Jacobians, including mixed-terms relating crossing points and sliding coordinates. For ease of implementation, we include the explicit formulations of forces and their Jacobians in a supplementary document.

4.1 Stretch

To model stretch, we follow the approach by Spillmann et al. [2007], who define a stretch energy that is quadratic in the strain along the yarn's centerline. With our discretization, stretch strain is constant on each yarn segment. For the warp segment $[\mathbf{q}_0, \mathbf{q}_1]$ in Fig. 5-d, it is simply $\varepsilon = \|\mathbf{w}\| - 1$. Then, the stretch energy of the segment for a stiffness k_s can be computed as:

$$V_{0,1} = \frac{1}{2}\, k_s\, \Delta u\, (\|\mathbf{w}\| - 1)^2, \tag{7}$$

where $k_s = Y\pi R^2$, and Y is the elastic modulus. Yarns of woven cloth are often close to inextensible, which requires the use of a high elastic modulus. An alternative would be to enforce inextensibility through constraints and Lagrange multipliers [Sueda et al. 2011]. However, we have designed a solver for implicit integration, described in Section 6, which allows the efficient simulation of stiff yarns.

4.2 Bending

For bending, we take a discrete differential geometry approach, and we define bending energies based on discrete curvatures at yarn crossings, separately for warp and weft yarns. There are several possible definitions of discrete curvature at yarn crossings [Sullivan 2008], and we choose to define it simply as the angle between yarn segments. This curvature is transformed into a curvature density dividing it by the arc length between segment centers. For the warp yarn in Fig. 5-b, given an angle θ between segments $[\mathbf{q}_2, \mathbf{q}_0]$ and $[\mathbf{q}_0, \mathbf{q}_1]$, curvature density at node $\mathbf{q}_0$ is defined as $\kappa = \frac{2\,\theta}{u_1 - u_2}$. We define a bending energy density with stiffness k_b that is quadratic in curvature. Integrating it over the half-segments adjacent to $\mathbf{q}_0$ results in a discrete bending energy

$$V_{2,0,1} = k_b\, \frac{\theta^2}{u_1 - u_2}, \tag{8}$$

where $k_b = B\pi R^2$, and B is the bending modulus. The expression could turn numerically unstable if yarn crossings became arbitrarily close. However, this does not happen in practice due to the contact model between parallel yarns described in Section 4.6. Bergou et al. [2008] choose a different discrete curvature metric, based on the tangent of the angle between segments. The resulting energy grows to infinity if a yarn bends completely, and we found this to create excessive resistance to bending in practice. Yet another option is to use a discrete curvature metric based on the sine of the half-angle between segments, but this metric produces a non-convex bending energy.

4.3 Compression at Yarn Crossings

Woven cloth is held together by inter-yarn friction, and admissible friction forces are a function of inter-yarn normal compression at yarn crossings. Our yarn discretization ignores the relative motion between warp and weft yarns along their normal direction, hence we cannot model normal compression as an elastic potential. Instead, we propose a quasi-static approximation that captures the desired friction effects. In essence, we estimate the compression force by averaging the normal components of warp and weft forces, depicted in Fig. 5-c, and then this compression can be used to model friction and shear forces [Page and Wang 2000].

The detailed computation is as follows. At each yarn crossing, we compute a best-fitting plane using the positions of the node and its four adjacent ones. We choose as normal direction $\mathbf{n}$ the normal of the plane pointing from the warp yarn toward the weft yarn. We apply crimp by offsetting the positions of warp and weft points in the normal direction by the yarn radii (Fig. 2, middle), and we recompute bending forces. At each yarn crossing, we estimate the compression force by summing the normal components of stretch and bending forces, $\mathbf{F}_s$ and $\mathbf{F}_b$ respectively, and averaging the resulting forces for warp and weft directions, i.e.,

$$F_n = \frac{1}{2}\, \mathbf{n}^T \left(\mathbf{F}_s(u) + \mathbf{F}_b(u) - \mathbf{F}_s(v) - \mathbf{F}_b(v)\right). \tag{9}$$

If the compression force is negative, we consider the yarns to be separating, and we clamp the force to zero. It would be possible to extend this model to handle adhesion.

Note that we account only for stretch and bending forces. If the fabric is stretched, then compression is dominated by stretch. However, when it is not stretched, then it is dominated by bending. For a flat cloth, it is crucial to account for the misalignments produced by crimp, otherwise friction forces cannot hold the yarns in place, and this is why we recompute bending forces after offsetting warp and weft points.

4.4 Friction

At each yarn crossing, we compute friction forces that try to prevent sliding between warp and weft yarns. We opt to model inter-yarn friction using a penalty-based approximation of the Coulomb model, similar to the one of Yamane and Nakamura [2006]. Our discretization based on yarn crossings simplifies greatly the formulation of friction, and a simple spring on each sliding coordinate produces effective results.

Given the yarn crossing $\mathbf{q}_0$, we set an anchor position $\bar{u}_0$ on the warp yarn, and similarly for the weft yarn. The anchor position is initialized as the warp sliding u_0 at the crossing. We model friction as a zero-rest-length viscoelastic spring between the anchor position and the actual warp coordinate.

The Coulomb model sets a limit $\mu\, F_n$ on the elastic component of the friction force, where F_n is the inter-yarn compression as computed in (9) above. If the limit is not reached, the contact is in

Figure 6: *Loose tank top with* 2023 *yarns and* 350530 *crossing nodes, showing large motions as well as small scale folds and wrinkles. Despite the highly dynamic motion of the mannequin, the simulation is robust.*

stick mode, and the force is defined by the spring. If the limit is exceeded, the contact is in slip mode, and the force is given by the Coulomb limit. In summary, the warp friction force is computed as:

$$F_{u_0} = \begin{cases} -k_f\,(u_0 - \bar{u}_0) - d_f\,\dot{u}_0, & \text{if stick} \\ -\text{sign}(u_0 - \bar{u}_0)\,\mu\,F_n - d_f\,\dot{u}_0, & \text{if slip.} \end{cases} \tag{10}$$

In addition, in slip mode we maintain the anchor position at a constant distance from the warp coordinate, such that the resulting spring force equals the Coulomb limit.

4.5 Shear

At yarn crossings, adjacent warp and weft yarns rotate on top of each other as a function of the shear angle ϕ, as shown in Fig. 5-d. This rotation produces two effects: yarn compression and contact friction. In addition, at interlaced crossings yarns suffer jamming as they collide.

To capture these effects, for every pair of warp and weft segments at a yarn crossing, we model an angular friction force and an elastic potential that depend on the shear angle ϕ. Let us consider, for example, the warp segment $[\mathbf{q}_0, \mathbf{q}_1]$ and the weft segment $[\mathbf{q}_0, \mathbf{q}_3]$ in Fig. 5-d. We define a shear energy density given by the shear rotation $\phi - \frac{\pi}{2}$, and we integrate it over the two half-segments incident in $\mathbf{q}_0$. In this integration, we found that it is sufficient to use the default inter-yarn distance L. This approximation has little effect in practice and it eliminates the need to compute shear forces and their Jacobians for sliding coordinates. The resulting shear energy with stiffness k_x is

$$V_{0,1,3} = \frac{1}{2}\,k_x\,L\,\left(\phi - \frac{\pi}{2}\right)^2, \tag{11}$$

where $k_x = SR^2$, and S is the contact shear modulus.

Normal compression increases the resistance to shear, and we model this effect by making the shear stiffness a function of the compression force, i.e., $k_x(F_n)$. Moreover, if either the warp or weft yarn segment is interlaced, we consider also shear jamming, modeled according to the following heuristics. We define the shear jamming angle ϕ_j as the angle at which the end-points of the warp and weft segments with radius R touch each other, i.e., $\phi_j = 2\,\arcsin\left(\frac{R}{L}\right)$. We model jamming as a strong nonlinearity in the shear stiffness, leaving it as a constant for shear angles above the jamming angle, and making it grow cubically for smaller angles.

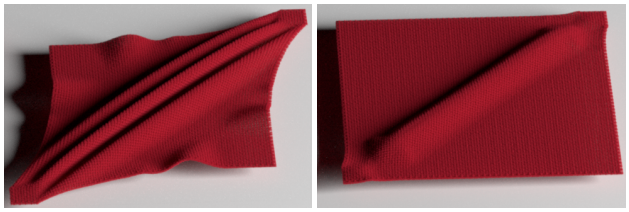

Figure 7: *Shear friction effects. The sample is stretched (left) and then relaxed (right), leaving a persistent wrinkle.*

Shear friction can be modeled using an angular spring between the current shear angle ϕ and an anchor angle $\bar{\phi}$, following the same approach as for sliding friction described in Section 4.4. We apply shear friction force only to the position of yarn crossing nodes, and it can be computed for each of the three nodes $\mathbf{q}_i$ in the example in Fig. 5-d as:

$$\mathbf{F}_{\mathbf{x}_i} = \begin{cases} -k_{f\phi}\,(\phi - \bar{\phi})\,\frac{\partial\phi}{\partial\mathbf{x}_i}^T, & \text{if stick} \\ -\text{sign}(\phi - \bar{\phi})\,\mu_\phi\,F_n\,\frac{\partial\phi}{\partial\mathbf{x}_i}^T, & \text{if slip.} \end{cases} \tag{12}$$

One of the visual effects of internal friction is the creation of persistent wrinkles, as demonstrated by Miguel and collaborators [2013]. Fig. 7 shows a small fabric sample that is first stretched (left) and then relaxed (right), leaving a persistent wrinkle along the stretch direction due to shear friction.

4.6 Contact between Parallel Yarns

Contact between adjacent parallel yarns can be easily modeled by adding a penalty energy if two yarn crossings get too close. We define the distance threshold d as four times the yarn radius if there is an interlaced yarn between the two crossings, and as twice the radius if the two yarns form a float. Given, for example, the weft yarns passing through $\mathbf{q}_0$ and $\mathbf{q}_1$ in Fig. 5-a, we define an energy density based on the distance between crossing points, and integrate this density over the weft half-segments incident on both nodes. Assuming that yarns are practically inextensible, the distance between crossing points can be approximated as the difference between warp

Figure 8: *Long shirt with* 3199 *yarns and* 559241 *crossing nodes. Using a yarn-level model inherently produces high resolution cloth dynamics, as shown by the small-scale wrinkles throughout the sleeves.*

sliding coordinates. And same as for shear, we found that it was sufficient to integrate the energy density using the default inter-yarn distance L, thus eliminating the need to compute complex coupling forces with weft sliding coordinates. The resulting penalty energy with stiffness k_c is:

$$V_{0,1} = \begin{cases} \frac{1}{2}\, k_c\, L\, (u_1 - u_0 - d)^2, & \text{if } u_1 - u_0 < d \\ 0, & \text{if } u_1 - u_0 \geq d. \end{cases} \qquad (13)$$

5 Contact Handling

Contact between interlaced yarns is handled implicitly by our discretization, and contact between adjacent parallel yarns is easily handled as described above. On the other hand, contact with other objects as well as long-range self-collisions require explicit collision processing. We rely on existing methods to detect and resolve collisions in our examples. As a summary, we define a thin volume around the cloth, which allows us to compute penetration depth and implement collision response through penalty energies.

To detect contact with volumetric objects, we use distance fields. In our examples we have used only rigid or articulated objects, hence it was sufficient to compute the distance field once as preprocessing. Given an object O, at every time step we query every yarn node $\mathbf{x}$ against the distance field of O, and define a collision if the distance to O is smaller than γ (in our examples, γ is 4 times the yarn radius R). The collision information is formed by the crossing point $\mathbf{x}$, the closest point $\mathbf{p}$ on the surface of O, and a contact normal $\mathbf{n}$. We have used as contact normal the normal at $\mathbf{p}$, although other options are possible.

To detect self-collisions, we define small volumetric elements on the surface of the cloth, and query the yarn nodes against these volumetric elements following the approach of Teschner et al. [2003]. We form two triangles with the 4 nodes defined by every two pairs of adjacent warp and weft yarns, and we protrude the triangles by a distance γ in the directions of and opposite the normal at each crossing point to form each volumetric element. Recall that in Section 4.3 we discuss the estimation of normals. We query all nodes against the protruded triangles, hashing AABBs of the protruded triangles on a regular grid for culling [Teschner et al. 2003]. If a point $\mathbf{x}$ is inside a protruded triangle, we define a collision, find the projected point $\mathbf{p}$ on the surface, and compute a contact normal $\mathbf{n}$ by interpolating the normals of the triangle's nodes.

For collision response, both with external objects or in self-collisions, we define a penalty force on the colliding point $\mathbf{x}$, with penalty distance $\mathbf{n}^T(\mathbf{p} - \mathbf{x}) + \gamma$, and direction $\mathbf{n}$. In the self-collision case, we also distribute the opposite force to the nodes that define the triangle, by using as weights the barycentric coordinates of $\mathbf{p}$ in the triangle. In addition to penalty response, we apply Coulomb friction approximated through clamped springs [Yamane and Nakamura 2006].

The obvious limitation of penalty-based response on thin objects is the chance of suffering pop-through problems. In our examples, we prevented pop-through by adding a damping term to the penalty response. A more robust solution would be to use a barrier method [Harmon et al. 2009].

6 GPU-Parallel Time Integration

We integrate the equations of motion (3) using backward Euler implicit integration [Baraff and Witkin 1998] with Newton's method and adaptive time stepping to ensure convergence. In practice, most solve steps require only one Newton iteration, and we allow up to five before halving the time step. The simulation cost is dominated by the solution to linear systems of the type $\mathbf{A}\,\mathbf{v} = \mathbf{b}$, where $\mathbf{v} = \dot{\mathbf{q}}$ is the vector of generalized velocities, and the system matrix is defined as $\mathbf{A} = \mathbf{M} - \Delta t\, \frac{\partial \mathbf{F}}{\partial \mathbf{v}} - \Delta t^2\, \frac{\partial \mathbf{F}}{\partial \mathbf{q}}$. With our discretization based on yarn crossings, $\mathbf{A}$ is formed by blocks of size 5×5.

We solve the linear system using the conjugate residual (CR) method. It is more robust than the conjugate gradient (CG) method for matrices that are close to semi-definite, as it computes the least-squares solution to the linear problem, at the expense of slightly higher cost and memory consumption.

An advantage of our yarn-level cloth model is the strong regularity of the system matrix $\mathbf{A}$, which allows a highly efficient implementation of numerical integration on GPUs, similar to GPU cloth solvers for regular triangle meshes [Tang et al. 2013]. We parallelize the computation of internal forces, their Jacobians, and the solution to the linear system on the GPU, but we execute collision detection on the CPU. All in all, the bottleneck of the solver is the sparse matrix-vector multiplication needed on each iteration of PCR.

We have optimized this product in multiple ways, exploiting the regularity of yarn-level cloth and our yarn-crossing discretization. Due to regularity of the weave pattern, the internal force of a node yields non-zero Jacobians w.r.t. 13 nodes (excluding collisions, each node interacts with 12 neighbors plus itself). Then, we split the system matrix $\mathbf{A}$ as the sum of a regular matrix $\mathbf{A}_r$ and the remaining tail matrix $\mathbf{A}_t$, where $\mathbf{A}_r$ contains the 13 regular 5×5 blocks per node, and $\mathbf{A}_t$ contains other blocks resulting from collisions. $\mathbf{A}_t$ amounts to only 14% of the product cost even with simple COO storage. We store the coefficients of $\mathbf{A}_r$ in a dense matrix in column-major order, with one row and $325 = 13 \times 25$ columns per node (2.53KB per node in double precision). The indices of $\mathbf{A}_r$ require a much smaller matrix, with one row and only 13 columns per node. Products involving $\mathbf{A}_r$ are parallelized on a per-node basis, and column-major storage of the coefficients provides extremely efficient coalescent access to the data. Our parallelization strategy mimics the one of the ELL and HYB matrix

Figure 9: *Middle: Snag formed by pulling two yarns while blocking the outward motion of the cloth. Left: Real snagging under similar conditions. Right: Close-up of the snag. We encourage the reader to zoom in to fully appreciate the details, such as rendered yarn torsion.*

formats in the Cusp CUDA library [Bell and Garland 2012], but we obtain more than a 40% speed-up over Cusp in sparse matrix-vector products thanks to node-level parallelization.

Overall, we achieve a 16x to 24x speedup over a multithreaded CPU implementation, depending mainly on the number of nodes (higher speedup for a higher number of nodes) and the number of collisions (lower speedup for a higher number of collisions, since collisions are treated on the CPU).

7 Rendering

For off-line rendering of our results, we transform the yarn geometry into a volumetric representation at the microfiber level, and use the volumetric path tracer Mitsuba [Jakob 2010]. It accounts for the anisotropic scattering of microfibers using a micro-flakes model. We have used 15 bounces of light in our examples.

For each yarn, we generate a polyline with offsetted node positions to account for crimp as in Section 4.3. We smooth the polyline yarns using Catmull-Rom splines, and we then use a modified version of the Lumislice method [Chen et al. 2003; Lopez-Moreno et al. 2014] to define the volumetric representation of the yarn geometry to be passed to the Mitsuba renderer. Each smoothed yarn is composed of thousands of twisted microfibers, and we set slices representing the microfiber density distribution perpendincularly along the thread segment at regular steps and incremental rotations. We compute the density by intersecting the absolute position of each texel at each slice with a 3D texture volume. This is done in a fragment shader with asynchronous 3D texel store calls. Our implementation is based on OpenGL shaders and instanced geometry, and we process up to 8M slices in less than 100ms on a standard desktop machine. We also store the tangents of microfibers in a 3D texture, computing differentials of texel positions at the current and previous slices, which differ mainly by the rotation along the yarn axis. Due to the asynchronous nature of texel calls in the shader, the previous slice is not accessible, hence we precompute the local differentials and pass them to each slice as a texture. All in all, the density and orientation of yarns at the microfiber level are stored in volumetric textures (3GB per frame in the examples).

8 Results

We now describe visual and numerical large-scale examples for several woven cloth simulation scenarios. All our examples were executed on a 3.4 GHz Quad-core Intel Core i7-3770 CPU with 32GB of memory, with an NVIDIA Titan Black graphics card with 6GB of memory. Collision detection is parallelized on the CPU, while the solution to dynamics is parallelized on the GPU as discussed in Section 6. All the simulations were executed with a time step of 1ms. The parameter values used in these examples are listed in Table 1. Representative timings are summarized in Table 2. Please see our accompanying video for all animation results.

Loose Tank Top We dressed a male mannequin with a loose tank top made of 2023 yarns and 350530 crossing nodes, one seam on each side and one seam on each shoulder (see Fig. 6). Yarn density is one yarn per millimeter (25 yarns per inch). The mannequin performs highly dynamic karate motions.

The simulation shows large-scale motion and folds resolved at the yarn level, combined with fine-scale effects. Cloth dynamics and contact resolution are robust even under such challenging motions.

Long Shirt We designed a shirt with sleeves to dress a dancing female mannequin (see Fig. 8). The shirt is made of 3199 yarns and 559241 crossing nodes, with seams on the sides of the body, the shoulders, the sleeve-body junctions and along the sleeves. Yarn density is one yarn per millimeter (25 yarns per inch).

Compared to the loose tank top, this simulation shows a higher complexity due to a higher crossing node count and the additional dynamics and contact mechanics of the sleeves.

Snags These examples show how extreme deformations of cloth produce highly complex plastic deformations at the yarn level, as well as the influence of local yarn dynamics on the global shape of the fabric. We produce a snag in the loose tank top, by pinching a node on the side seam and pulling it outwards very fast (see Fig. 1). The deformation due to pulling generates a small hole: the warp yarn being pulled pushes the weft yarns away, in a clear example of yarn sliding and yarn contact dynamics. In addition, the snag is transferred all across the shirt, showing fine wrinkles as the complex effect of yarn sliding and friction. Such a plastic effect can only be achieved by simulating fabric at the yarn level with yarn-yarn interaction.

A second snag is produced in the belly area of the loose tank top by pulling a crossing node and fixing the four neighboring nodes that are not on the yarns being pulled (see Fig. 9). This setup tries to mimic the pulling of a yarn while locally blocking the outward motion of the fabric with the hand. The cloth wrinkles forming a cross shape, showing another familiar snagging pattern.

Tearing Simulating the tearing of cloth using our yarn-level model is straightforward, as the complex and visually rich behavior of frayed edges and loose yarns come naturally with yarn-level dynamics. We implemented fracture simply by splitting yarns when a threshold of stretch stress is exceeded, followed by a relaxation step to allow a correct stress relief and avoid erratic crack propagation. More sophisticated approaches could be used, such as separation

Figure 10: *The tank top is torn open by grabbing some nodes and pulling them apart. Yarns are detached and edges are frayed in the process.*

Example	Seg. length (mm)	Yarn radius (mm)	Elastic mod. (Pa)	Bending mod. (Pa)	Contact Shear mod. (Pa)	Sliding Fric. Coef.	α, β
Tank top (Fig. 6)	1	0.25	1e7	1e-2	1e4	0.3	10, 0.05
Long shirt (Fig. 8)	1	0.25	1e7	2e-2	1e4	0.3	20, 0.05
1M nodes sheet (Fig. 11)	0.254	0.06	1e7	1e-2	3e5	0.3	35, 0.01

Table 1: *Parameter values used in our examples.*

Example	Collisions	Forces	Jacobians	Solver
Tank top	322	28	280	2513
Long shirt	519	34	365	4033
1M nodes sheet	803	35	385	930

Table 2: *Average cost per time step (in milliseconds) for our examples, broken down by step. The time step is 1ms.*

tensors [O'Brien and Hodgins 1999] and local relaxation substeps [Pfaff et al. 2014]. Node resampling is frequently triggered during fracture and highly plastic behaviors due to yarn pullouts and sliding past the end of a yarn.

We tear the loose tank top by pinching two sets of crossing nodes in the torso area and pulling them apart in opposite directions, creating a vertical fracture path and a diamond-shaped opening as shown in Fig. 10. Individual yarns detach from the edges of the crack, and either hang or stretch the edges across the opening. These loose yarns and the resulting frayed edges are commonly seen in the tearing of many types of fabric. More subtle plastic deformations can be observed around the crack, mainly due to yarn sliding.

Weaving Patterns Our yarn-level model allows for easy configuration and simulation of different weave patterns. As mentioned in Section 3.1, configuring the fabric for a specific weave pattern is just a matter of setting a flag for each node that specifies which yarn is on top. Weave patterns directly affect the global and local behavior of cloth, mainly due to the different number of floats. Shear, for instance, is greatly influenced by the number of crossings and floats in the fabric. The visual aspect of the cloth also changes according to the pattern.

We simulated three 25x25cm cloth sheets by hanging them from two corners. Yarn density is one yarn per millimeter (25 yarns per inch). The three sheets are exactly the same except for the weaving pattern, where the first one is plain weave, the second one is twill and the third one is satin. Fig. 3 shows a still of each sheet after 2 seconds of simulation. The sheets exhibit clearly distinctive behaviors: from left to right, the wrinkles move to the bottom, the bottom edge of the fabric falls lower, and the top edge shows higher curvature. These effects are due to lower shear stiffness for weaves with more floats, which is the expected result in reality. Lower shear stiffness results in better drape quality. The visual appearance is also different between the 3 stills. In the top part of each sheet, the "see-through" effect due to stretch reveals the different weaving structures of the cloth. We can also observe how the twill woven sample exhibits its characteristic diagonal pattern.

We put the three sheets through a shear frame test and measured the overall shear through time. Results are plotted in Fig. 4, showing force-angle plots for each weave pattern (plain in green, twill in blue, satin in red). The plots exhibit hysteresis due to friction and nonlinearity due to jamming as observed on real fabrics [Miguel et al. 2012], as well as the influence of the weave pattern. Again, weave patterns with more floats are less resistant to shear, as expected.

We simulated a fourth cloth sheet using plain weave, but this time with 4 yarns per millimeter (100 yarns per inch). Given the sheet's size, this yarn density translates into 1 million crossing nodes. This example, shown in Fig. 11, shows how our model can handle very high yarn densities found in common woven fabrics such as bed linen. As per textile nomenclature, our 100 yarns per inch is equivalent to a thread count of 200.

9 Conclusions and Future Work

In this paper, we have presented an efficient method to simulate woven cloth at the yarn level. The key novelty in our model is a discretization of yarn crossings that resolves yarn-yarn contact implicitly and represents inter-yarn sliding efficiently. Effects such as inter-yarn friction, shear, and contact are also captured with simple force models. We show that our yarn-level model enables the simulation of effects such as tearing with frayed edges, plasticity due to snags, or nonlinear behavior due to fine-scale friction.

One of the potential advantages of yarn-level models is the possibility to replicate with high fidelity the nonlinear mechanics of real

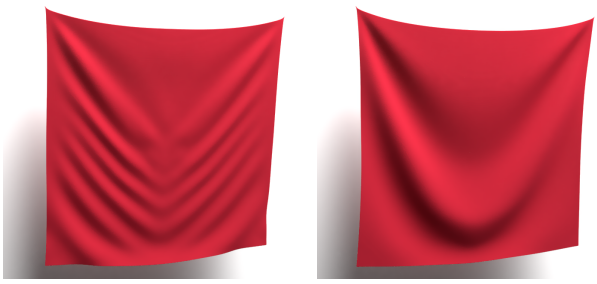

Figure 11: *A 100-yarn-per-inch plain-weave sheet (1 million crossing nodes). Small wrinkles appear during motion (left) until the sheet comes to rest exhibiting large draping wrinkles (right).*

cloth. This would require estimating the parameters of our model from force-deformation measurements of real cloth. The fitting results could be compared to those of nonlinear cloth models.

Our model could be extended in several ways. One limitation of the model is that it approximates the compression between crossing yarns as a function of stretch and bending forces. A future possibility would be to incorporate compression as an extra DoF, and add a compression potential to the system energy. We currently model stretch forces using a stretch potential, but another possibility would be to consider yarns to be inextensible, and account for the compression produced during stretch due to crimping.

Even though our examples are limited to orthogonal weave patterns, our discretization is general and could be applied to arbitrary settings with interlaced yarns. One simple extension would be to handle triaxial weaving. But it would also be interesting to explore the application of our discretization to knitted cloth.

A final important limitation in our current implementation is due to the use of penetration-depth queries and penalty-based collision response. To ensure robustness of contact handling, we must use stiff penalty energies and limit the amount of motion per time step. Robustness could be improved using continuous collision detection and constraint-based response, although contact handling might then become the bottleneck.

Acknowledgements

We wish to thank Eder Miguel and the GMRV group for diverse help with our submission, Wenzel Jacob for support with Mitsuba, and the Berkeley Garment Library [de Joya et al.] for the mannequin model and animations. This work was supported in part by the Spanish Ministry of Economy (TIN2012-35840) and the European Research Council (ERC-2011-StG-280135 Animetrics). The work of Gabriel Cirio was funded by the Spanish Ministry of Science and Education through a Juan de la Cierva Fellowship.

References

BARAFF, D., AND WITKIN, A. 1998. Large steps in cloth simulation. In *Proceedings of ACM SIGGRAPH 98*, 4354.

BELL, N., AND GARLAND, M., 2012. Cusp: Generic parallel algorithms for sparse matrix and graph computations. Version 0.3.0.

BERGOU, M., WARDETZKY, M., ROBINSON, S., AUDOLY, B., AND GRINSPUN, E. 2008. Discrete elastic rods. *ACM Trans. Graph. 27*, 3, 63:163:12.

BERTHOUZOZ, F., GARG, A., KAUFMAN, D. M., GRINSPUN, E., AND AGRAWALA, M. 2013. Parsing sewing patterns into 3D garments. *ACM Trans. Graph. 32*, 4, 85:1–85:12.

BOISSE, P., BORR, M., BUET, K., AND CHEROUAT, A. 1997. Finite element simulations of textile composite forming including the biaxial fabric behaviour. *Composites Part B: Engineering 28*, 4, 453–464.

BREEN, D. E., HOUSE, D. H., AND WOZNY, M. J. 1994. Predicting the drape of woven cloth using interacting particles. In *Proceedings of ACM SIGGRAPH 94*, 365–372.

BRIDSON, R., MARINO, S., AND FEDKIW, R. 2003. Simulation of clothing with folds and wrinkles. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer animation 2003*, 2836.

CASATI, R., AND BERTAILS-DESCOUBES, F. 2013. Super space clothoids. *ACM Trans. Graph. 32*, 4, 48.

CHEN, Y., LIN, S., ZHONG, H., XU, Y.-Q., GUO, B., AND SHUM, H.-Y. 2003. Realistic rendering and animation of knitwear. *IEEE Transactions on Visualization and Computer Graphics 9*, 1 (Jan.), 43–55.

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

DAVIET, G., BERTAILS-DESCOUBES, F., AND BOISSIEUX, L. 2011. A hybrid iterative solver for robustly capturing coulomb friction in hair dynamics. *ACM Trans. Graph. 30*, 6, 139:1–139:12.

DE JOYA, J.M., NARAIN, R., O'BRIEN, J., SAMII, A., AND ZORDAN, V. Berkeley garment library. http://graphics.berkeley.edu/resources/GarmentLibrary/.

DUAN, Y., KEEFE, M., BOGETTI, T. A., AND POWERS, B. 2006. Finite element modeling of transverse impact on a ballistic fabric. *International Journal of Mechanical Sciences 48*, 1, 33–43.

ETZMUSS, O., KECKEISEN, M., AND STRASSER, W. 2003. A fast finite element solution for cloth modelling. In *Proceedings of Pacific Graphics 2003*, 244–251.

GOLDENTHAL, R., HARMON, D., FATTAL, R., BERCOVIER, M., AND GRINSPUN, E. 2007. Efficient simulation of inextensible cloth. *ACM Trans. Graph. 26*, 3, 49.

GOLDSTEIN, H., POOLE, C. P., AND SAFKO, J. L. 2001. *Classical Mechanics (3rd Edition)*, 3 ed. Addison-Wesley.

GRINSPUN, E., HIRANI, A. N., DESBRUN, M., AND SCHRÖDER, P. 2003. Discrete shells. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer animation 2003*, 6267.

HARMON, D., VOUGA, E., SMITH, B., TAMSTORF, R., AND GRINSPUN, E. 2009. Asynchronous contact mechanics. *ACM Trans. Graph. 28*, 3, 97.

HEARLE, J. W. S., GROSBERG, P., AND BACKER, S. 1969. *Structural Mechanics of Fibers, Yarns, and Fabrics*, vol. 1. John Wiley & Sons Inc, New York.

JAKOB, W., 2010. Mitsuba renderer. http://www.mitsuba-renderer.org.

KALDOR, J. M., JAMES, D. L., AND MARSCHNER, S. 2008. Simulating knitted cloth at the yarn level. *ACM Trans. Graph. 27*, 3, 65:165:9.

KALDOR, J. M., JAMES, D. L., AND MARSCHNER, S. 2010. Efficient yarn-based cloth with adaptive contact linearization. *ACM Trans. Graph. 29*, 4, 105:1–105:10.

KAWABATA, S., NIWA, M., AND KAWAI, H. 1973. The finite-deformation theory of plain-weave fabrics part i: The biaxial-deformation theory. *Journal of the Textile Institute 64*, 1, 21–46.

KING, M. J., JEARANAISILAWONG, P., AND SOCRATE, S. 2005. A continuum constitutive model for the mechanical behavior of woven fabrics. *International Journal of Solids and Structures 42*, 13, 3867–3896.

LOPEZ-MORENO, J., CIRIO, G., MIRAUT, D., AND OTADUY, M.A. 2014. GPU Visualization and Voxelization of Yarn-Level Cloth. Proceedings of the Spanish Computer Graphics Conference.

MCGLOCKTON, M. A., COX, B. N., AND MCMEEKING, R. M. 2003. A binary model of textile composites: III high failure strain and work of fracture in 3D weaves. *Journal of the Mechanics and Physics of Solids 51*, 8, 1573–1600.

METAAPHANON, N., BANDO, Y., CHEN, B.-Y., AND NISHITA, T. 2009. Simulation of tearing cloth with frayed edges. *Comput. Graph. Forum 7*, 1837–1844.

MIGUEL, E., BRADLEY, D., THOMASZEWSKI, B., BICKEL, B., MATUSIK, W., OTADUY, M. A., AND MARSCHNER, S. 2012. Data-driven estimation of cloth simulation models. *Comp. Graph. Forum 31*, 519–528.

MIGUEL, E., TAMSTORF, R., BRADLEY, D., SCHVARTZMAN, S. C., THOMASZEWSKI, B., BICKEL, B., MATUSIK, W., MARSCHNER, S., AND OTADUY, M. A. 2013. Modeling and estimation of internal friction in cloth. *ACM Trans. Graph. 32*, 6, 212:1–212:10.

NADLER, B., PAPADOPOULOS, P., AND STEIGMANN, D. J. 2006. Multiscale constitutive modeling and numerical simulation of fabric material. *International Journal of Solids and Structures 43*, 2, 206 – 221.

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

NG, S.-P., TSE, P.-C., AND LAU, K.-J. 1998. Numerical and experimental determination of in-plane elastic properties of 2/2 twill weave fabric composites. *Composites Part B: Engineering 29*, 6, 735–744.

O'BRIEN, J. F., AND HODGINS, J. K. 1999. Graphical modeling and animation of brittle fracture. In *Proceedings of ACM SIGGRAPH 99*, 137146.

PAGE, J., AND WANG, J. 2000. Prediction of shear force and an analysis of yarn slippage for a plain-weave carbon fabric in a bias extension state. *Composites Science and Technology 60*, 7, 977 – 986.

PAI, D. K. 2002. Strands: Interactive simulation of thin solids using cosserat models. *Comput. Graph. Forum 21*, 3, 347–352.

PARSONS, E. M., WEERASOORIYA, T., SARVA, S., AND SOCRATE, S. 2010. Impact of woven fabric: Experiments and mesostructure-based continuum-level simulations. *Journal of the Mechanics and Physics of Solids 58*, 11, 1995–2021.

PARSONS, E. M., KING, M. J., AND SOCRATE, S. 2013. Modeling yarn slip in woven fabric at the continuum level: Simulations of ballistic impact. *Journal of the Mechanics and Physics of Solids 61*, 1, 265–292.

PEIRCE, F. T. 1937. The geometry of cloth structure. *Journal of the Textile Institute Transactions 28*, 3, T45–T96.

PFAFF, T., NARAIN, R., DE JOYA, J. M., AND O'BRIEN, J. F. 2014. Adaptive tearing and cracking of thin sheets. *ACM Trans. Graph. 33*, 4, 110:1–9.

PROVOT, X. 1995. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *In Graphics Interface*, 147–154.

REESE, S. 2003. Anisotropic elastoplastic material behavior in fabric structures. In *IUTAM Symposium on Computational Mechanics of Solid Materials at Large Strains*, 201–210.

SPILLMANN, J., AND TESCHNER, M. 2007. CoRdE: cosserat rod elements for the dynamic simulation of one-dimensional elastic objects. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2007*, 6372.

SPILLMANN, J., AND TESCHNER, M. 2009. Cosserat nets. *IEEE Transactions on Visualization and Computer Graphics 15*, 2, 325 –338.

SUEDA, S., JONES, G. L., LEVIN, D. I. W., AND PAI, D. K. 2011. Large-scale dynamic simulation of highly constrained strands. *ACM Trans. Graph. 30*, 4, 39:1–10.

SULLIVAN, J. M. 2008. Curves of finite total curvature. In *Discrete Differential Geometry*, A. I. Bobenko, J. M. Sullivan, P. Schröder, and G. M. Ziegler, Eds., vol. 38 of *Oberwolfach Seminars*. Birkhäuser, Basel, 137–161.

TANG, M., TONG, R., NARAIN, R., MENG, C., AND MANOCHA, D. 2013. A GPU-based streaming algorithm for high-resolution cloth simulation. *Computer Graphics Forum 32*, 7, 21–30.

TERZOPOULOS, D., PLATT, J., BARR, A., AND FLEISCHER, K. 1987. Elastically deformable models. In *Proceedings of ACM SIGGRAPH 87*, 205–214.

TESCHNER, M., HEIDELBERGER, B., MUELLER, M., POMERANETS, D., AND GROSS, M. 2003. Optimized spatial hashing for collision detection of deformable objects. 47–54.

VOLINO, P., COURCHESNE, M., AND MAGNENAT THALMANN, N. 1995. Versatile and efficient techniques for simulating cloth and other deformable objects. In *Proceedings of ACM SIGGRAPH 95*, 137–144.

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

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

XIA, W., AND NADLER, B. 2011. Three-scale modeling and numerical simulations of fabric materials. *International Journal of Engineering Science 49*, 3, 229–239.

YAMANE, K., AND NAKAMURA, Y. 2006. Stable penalty-based model of frictional contacts. In *Proceedings of IEEE International Conference on Robotics and Automation 2006*, 1904–1909.