Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2014), pp. 1–10
Vladlen Koltun and Eftychios Sifakis (Editors)

Position-based Elastic Rods

Nobuyuki Umetani Ryan Schmidt Jos Stam

Autodesk Research

Abstract

We present a novel method to simulate complex bending and twisting of elastic rods. Elastic rods are commonly simulated using force based methods, such as the finite element method. These methods are accurate, but do not directly fit into the more efficient position-based dynamics framework, since the definition of material frames are not entirely based on positions. We introduce ghost points, which are additional points defined on edges, to naturally endow continuous material frames on discretized rods. We achieve robustness by a novel discretization of the Cosserat theory. The method supports coupling with a frame, a triangle, and a rigid body at the rod's end point. Our formulation is highly efficient, capable of simulating hundreds of strands in real-time.

Categories and Subject Descriptors (according to ACM CCS): I.6.8 [Computer Graphics]: Simulation and modeling—Animation

1. Introduction

Position-based dynamics (PBD) has been widely accepted in the field of computer animation due to its efficiency, robustness and simplicity. The goal of the PBD is not to simulate physics as accurately as possible, but rather to sacrifice some quantitative accuracy to generate visually plausible simulation results very quickly. To this end, PBD has broadly been applied in many game engines and visual effects, where speed and controllability is crucial. PBD has primarily been used to simulate various physical phenomena associated with solid and thin-shell (i.e., clothing) deformations [MHHR07]. We present a new application of PBD to elastic rod simulation, which is essential for animating thin strands such as hair, fur, ropes, and so on.

Because of the existence of twist, rod simulations are more complicated than solid or thin-shell simulations, making it hard to implement them in the PBD framework. PBD requires an object's deformations be characterized with discrete positions of points. However twist of a rod cannot be directly specified with only positions of the rod *center line*, because twist requires angular information –rather than positional– which describes how much the material is twisted around the center line. Since complex nonlinear deformation, such as out-of-plane buckling under torsional strain (e.g., Fig. 2), is caused by an interaction between twist and bending, handling of twist is critical in nonlinear deformation of rods.

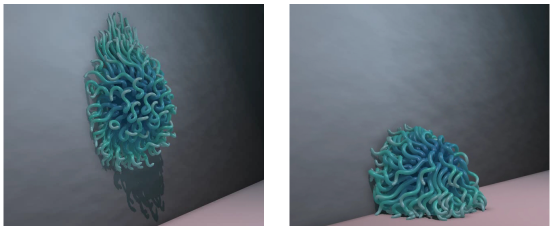

Figure 1: *A squishy ball hits a wall. Tentacles of the squishy ball was modeled with our position-based elastic rods.*

Representation of twist in elastic rods is actively researched, but so far no previous work has successfully implemented it in a PBD framework. While bending and stretching can be characterized by the position of the centerline, the twist is represented using material frames defined on the centerline. Constructing a material frame is difficult because its orientation is coupled with the centerline. Various approaches have been proposed. For example, Spillman et al. [ST07] use the penalty method to orient the material frame to rods' tangential direction. [BWR*08] defined a material frame by rotating a Bishop frame in the direction of the rod's tangential direction. However, theses approaches are not suitable for PBD in which all deformations are described via discrete point positions, not frame orientations.

Our goal is to incorporate the twisting and bending physics of elastic rods into the PBD framework. With proper handling of twist, we can produce (i) visually plausible simulation of rod deformations, capable of reproducing various phenomena, which are (ii) fully controllable by physically meaningful parameters, (iii) the PBD framework ensures robustness, speed and simple implementation. Our position-based elastic rod model is derived from Cosserat theory [CC07], which describes the physical principles of elastic rods. Even though our model may not quantitatively agree with real-world experiments, it reproduces plausible deformation such as looping phenomena.

Our key contribution is to introduce *ghost points*, which are placed on edges to represents material points distributed around the edge. Using ghost points, we can represent the material frame so that its orientation naturally follows the rods' tangential direction, leading to an efficient formulation. The dynamics of these ghost points are treated straightforwardly in the framework of PBD, hence our implementation is simple and fast.

We propose a novel discretization of the Cosserat theory, which further enhances robustness and simplicity of the simulation. Qith this enhancement, we can enforce the twisting and bending forces without computing expensive trigonometric functions or their inverses. Moreover, our formula prevents the rod from flipping in high-torsion situations. We extend our discretization to also specify boundary conditions, in which the rod is connected to a rigid body or a triangle with one- or two-way coupling.

We find that we can interpret the traditional constraint enforcement procedure of PBD in a variational formulation. This allows us to extend PBD so that it can handle multiple constraints simultaneously, resulting in a significant speed-up. Furthermore, because the material's deformation is usually written in a vector or tensor form, our variational approach enables us to manipulate the multi-dimensional anisotropic stiffness parameters of the material.

Our formulation is very efficient. For example, on a modern workstation we can simulate 200 strands of hair, discretized with 20 edges, at an interactive rate. Our rods are also highly stable, to the point of being able to recover from a random initial deformation without kinks or instability. We demonstrate the flexibility of our approach by showing various examples coupling rods with other rigid and deformable objects. Our contributions include:

- Ghost points to discretize material frames on a rod which are directly compatible with position-based dynamics.
- New robust discrete elastic rod formulation based on Cosserat theory which also can specify boundary conditions of rods.
- New variational interpretation of the PBD constraint enforcement, which enables fusing multiple constraints.

2. Related Work

Since Pai et al. [Pai02] first introduced the Cosserat theory of elastic rod simulation into graphics, it has remained an active research topic. Most elastic rod models are based on Cosserat theory, which describes nonlinear deformation of rods. A primary difference between formulations is whether the centerline is represented implicitly or explicitly.

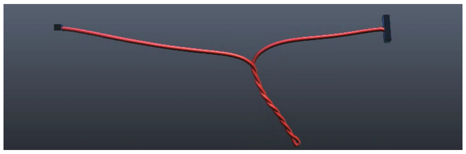

Figure 2: *Rods show dynamic looping phenomenon known as Plectoneme when the both ends of a rod is clamped and twisted. This nonlinear behavior cannot be reproduced with a model without coupling of twist and bending.*

Implicit discretization of Cosserat theory. This approach uses curvatures to represent curved rod shape and recovers the rods' explicit centerline shape by integrating frames from one end. The first model from Pai et al. [Pai02] falls into this category. Later, this approach was further extended for helical elastic rods [BAC*06] to simulate curled hairs. Recently, a super-clothoid model has been proposed [CBD13] that uses power-series to integrate curvature to find the center line of a clothoid with a stability guarantee. Implicit models achieve high-order smoothness of the rod with fewer DoFs and are suitable for hair simulations where one end is clamped and another end is free. However, since this approach solves for derivative information of material frames, it cannot be combined with position based approaches.

Explicit discretization of Cossarat theory. Our methods are greatly inspired by the CORDE model proposed by Spillman et al. [ST07]. We both begin with an explicit representation of the centerline and define material frames to apply Cosserart theory. However, Spillman et al. define material frames independently from the centerline position, and thus require penalty-based methods to make the frames follow the centerline. This approach does not directly fit into the PBD framework because the variables have frame orientation, in addition to the position of the centerline.

Formulation based on Bishop frame. In the explicit centerline model, the definition of a material frame that tracks the rods' tangential direction is not straightforward. The discrete elastic rod proposed by Bergou et al. [BWR*08] was a breakthrough because their material frames, constructed from the Bishop frame, automatically follows the deformation of the rod center line. The Bishop frame construction was later transferred from the spatial domain to the temporal domain to enhance efficiency [BAV*10, KJM10]. However as transported frames are not straightforward to represent in a point-based formulation, formulating constraints for PBD is a challenge with Bishop-based approaches.

Mass-spring model. Several studies [SLF08, MSW*09] modeled hair deformation with network of masses connected by springs. The mass-spring model is simple and robust, but it is difficult to separate the bending stiffness from twisting stiffness. In contrast, our model is highly controllable; the properties of rods can be manipulated with independent, physically meaningful parameters.

Position-based dynamics. PBD provides fast and robust simulation tailored for computer animations [MHHR07, BMOT13]. The core ideas, such as Verlet integration and stiffness based on constraints, are presented in [Jak01]. While PBD is not quantitatively accurate, in that it cannot reproduce real-world experimental data, PBD animation is sufficiently plausible for computer animations. PBD has been applied in game engines such as Nvidia PhysX$^{\text{TM}}$and visual effect engines such as Nucleus [Sta09]. The numerous games and films produced with these tools are a testament to the importance and utility of PBD. Recently, application of PBD has been extended to fluid dynamics [MM13]. Muller et al. [MKC12] presented a fast method to apply inextensible constraints in the PBD for hair and fur without the twisting and bending forces. To our knowledge, twisting and bending of elastic rods have yet to be solved in the PBD framework.

Shape matching method [MHTG05] is the most similar to PBD, and in that domain several studies tackle simulation of elastic rods. The model in [RKM*12] represents the elastic rod by appending twisting force, computed by method using Bishop frame, to the shape matching deformation. The twisting force is modeled as an external force hence this method doesn't benefit from shape matching techniques unconditional stability. Oriented particles [MC11] consider rotational inertia at the nodes and perform rod simulation stably, but do not handle the coupling of bending and twisting. Shape matching method treats elasticity as a force to get back to the rigid goal shape, thus cannot easily handle coupling of twisting and bending where the rigid shape cannot be a goal shape.

3. Discrete formulation of rod

Like other PBD formulations, our position-based elastic rod handles elasticity in the form of constraints. Similar to the Gauss-Seidel method, the particle locations are iteratively updated, so that linearized constraints are locally satisfied. Hence, how to construct constraint for twisting and bending is essential for elastic rod simulation.

We our technique is based on the Cosserat theory, which we briefly review (§ 3.1). To define differentiable continuous material frames on a rod in a discrete setting, we introduce ghost points (§ 3.2). Next, we describe a novel discretization of Cosserat theory that makes simulation simpler and faster (§ 3.3). We extend the types of constraints that PBD can handle by introducing a variational interpretation (§ 4). Using this extension, constraints for our elastic rod are described (§ 5). Finally, we describe end point constraints that fix the position and orientation of a rod at its end (§ 5.1).

3.1. Cosserat theory

We briefly review elastic rod simulation based on the Cosserat theory. For a more detailed discussion, please refer to [Ant05]. We assume that a rod is parameterized by arclength $s \in [0, \bar{L}]$, where $\bar{L}$ is the initial rod length. The rod's one dimensional configuration is described by the *centerline*, where $\mathbf{p}(s) \in \mathbb{R}^3$ is the position of the rod's centroid at the location s.

The twist of a rod cannot be represented by the centerline positions. Thus, at every point s on the rod, we define an orthonormal material bases $\mathbf{d}^1(s), \mathbf{d}^2(s), \mathbf{d}^3(s)$. A *material frame* is a 3×3 orthogonal matrix $\mathbf{D}(s) = [\mathbf{d}^1, \mathbf{d}^2, \mathbf{d}^3]$ which has material bases as columns. In differential geometry, the *Darboux vector* describes how the material frame evolves along the curve, and is a key component of Cosserat theory. The Darboux vector is an axial vector of frame rotation with respect to change of s. More specifically, using the bases of the material frame, the Darboux vector can be written as

$$\boldsymbol{\omega}(s) = \frac{1}{2} \sum_{k=1}^{3} \mathbf{d}^k(s) \times \mathbf{d}^k(s)', \tag{1}$$

where the prime ($'$) denotes derivative with respect to s.

Material coordinates of the Darboux vector at a location s become $\omega_i(s) = \boldsymbol{\omega}(s) \cdot \mathbf{d}^i(s)$. This ω_i encodes how the frame rotates around the material frame basis $\mathbf{d}^i$. In Cosserat theory, the rod's bending and twisting energy is defined using the material coordinates of the Darboux vector as

$$V = \int_0^{\bar{L}} \frac{1}{2} \sum_{i=1}^{3} \sum_{j=1}^{3} (\omega_i - \bar{\omega}_i) K_{ij} (\omega_j - \bar{\omega}_j) dt, \tag{2}$$

where $\bar{\omega}_i$ is the material coordinate of the Darboux vector in an initial zero-stress configuration, and $K_{ij} \in \mathbb{R},\ \ i, j \in \{1, 2, 3\}$ is a positive definite symmetric stiffness matrix.

The initial material frame $\mathbf{D}(s)$ can be chosen arbitrarily. In practice, the material coordinates are chosen such that one of the basis vectors, such as $\mathbf{d}^3$, corresponds the rod's tangential direction so that $\mathbf{p}(s)'$ become parallel to $\mathbf{d}^3$ (this frame choice is called the *adapted frame* [BWR*08]). Next, $\mathbf{d}^1$ and $\mathbf{d}^2$ are chosen as the principal axes of the second moment of area computed for the rod's cross-section. In this particular setting, assuming isotropic homogeneous material, $\mathbf{K}$ becomes diagonal and the K_{11} and K_{22} correspond to stiffness of bending around the axes $\mathbf{d}^1$ and $\mathbf{d}^2$, and K_{33} specifies the twisting stiffness.

Material frames must track the motion of the centerline over time. More specifically, when a rod changes its centerline, the material frames should move so that the tangential direction always corresponds to the basis $\mathbf{d}^3$. This constraint on the material frame complicates the simulation of rods. In contrast to the penalty method used in the CORDE model, the construction of the material frame that we describe in the following section naturally satisfies the above condition.

3.2. Continuous material frames on a discrete rod

We explicitly model the rod's centerline as a polyline with $N+1$ vertices $\{\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_N\}$. Edge $e \in \{1,\dots,N\}$ has end points $\mathbf{p}_{e-1}$ and $\mathbf{p}_e$, and masses $\{m_0,\dots,m_N\}$ are associated with the vertices. As shown in Fig. 3, we assume each edge e has a *ghost point* $\mathbf{p}_e^g \in \mathbb{R}^3$, which is placed on the perpendicular bisector of each edge. Masses $\{m_1^g,\dots,m_N^g\}$ are associated with the ghost points .

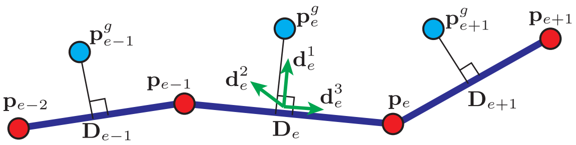

Figure 3: *Configuration of edges, points (red) and ghost points (cyan).*

Within the PBD framework, our ghost points are treated in the same manner as any other points. The distance between the ghost point and the edge is set to a fixed value $\bar{l}_g$, which we set to the average edge length. Ghost points represent material distributed around the edges, thus inertia for the rotation around an edge becomes $\bar{l}_g^2 m_e^g$. As a result, we can control rotational inertia by changing $\bar{l}_g$.

Material frame at the center of an edge. Using the ghost points, we define frame $\mathbf{D}_e$ at the center of edge e as

$$\left\{\begin{array}{lcl} \mathbf{d}_e^3 & = & (\mathbf{p}_e - \mathbf{p}_{e-1}) \,/\, |\mathbf{p}_e - \mathbf{p}_{e-1}|, \\ \mathbf{d}_e^2 & = & \mathbf{d}_e^3 \times (\mathbf{p}_e^g - \mathbf{p}_{e-1}) \,/\, |\mathbf{d}_e^3 \times (\mathbf{p}_e^g - \mathbf{p}_{e-1})|, \\ \mathbf{d}_e^1 & = & \mathbf{d}_e^2 \times \mathbf{d}_e^3. \end{array}\right. \tag{3}$$

For the rest of the rod, the frame is continuously interpolated from these edge centers. Following [ST07], we define an *orientation element e* ($e \in \{1,\dots,N-1\}$) as a segment starting at the midpoint of edge e and ending at the midpoint of edge $e+1$. Within the orientation element, the frame is interpolated from the two end frames $\mathbf{D}_e = [\mathbf{d}_e^1, \mathbf{d}_e^2, \mathbf{d}_e^3]$ and $\mathbf{D}_{e+1} = [\mathbf{d}_{e+1}^1, \mathbf{d}_{e+1}^2, \mathbf{d}_{e+1}^3]$.

Interpolation of frames. In order to compute the Darboux vector and its material coordinates for Cosserat theory, we interpolate between two frames $\mathbf{D}_e$ and $\mathbf{D}_{e+1}$ inside the orientation element e (see Fig. 4-left). We assume that the material frame changes *uniformly* from $\mathbf{D}_e$ into $\mathbf{D}_{e+1}$ (i.e., spherical linear interpolation). This uniform interpolation between two frames can be easily computed using an axis-angle formulation. The rotation matrix between two frames $\mathbf{R} = \mathbf{D}_{e+1}(\mathbf{D}_e)^T$ is represented with a 3-vector $\mathbf{\Theta} = \theta\mathbf{n}$, where $\theta \in [0,\pi)$ is a rotation angle and $\mathbf{n}$ is an unit axis vector (see Fig. 4-middle). Using the matrix exponential, the rotation matrix can then be written as

$$\mathbf{R} = \exp([\mathbf{\Theta}]), \tag{4}$$

where "$[\cdot]$" denotes the cross product matrix (i.e., $[\mathbf{a}]$ is a 3x3 skew-symmetric matrix as $[\mathbf{a}]\mathbf{b} = \mathbf{a}\times\mathbf{b},\ \ \mathbf{a},\mathbf{b} \in \mathbb{R}^3$).

The vector $\mathbf{\Theta}$ parameterizes the rotation, hence we can scale rotation by scaling this vector with the ratio inside orientation element as $\mathbf{R}(s) = \exp([\mathbf{\Theta}]r)$, where $r = (s - s_e)/\bar{l}_e$. Here, $\bar{l}_e$ is the initial length of the orientation element e and s_e is the arclength parameter s at the midpoint of the edge e. With this uniform scaling, bases of the frame are interpolated with s as

$$\mathbf{d}^i(s) = \exp([\mathbf{\Theta}]r)\mathbf{d}_e^i, \tag{5}$$

The differentiation of basis $\mathbf{d}^i(s)$ with respect to s can be written as $\mathbf{d}^i(s)' = ([\mathbf{\Theta}]/\bar{l})\mathbf{d}^i(s)$. Thus, from (1), the Darboux vector can be written as $\boldsymbol{\omega}(s) = \mathbf{\Theta}/\bar{l}$, which is constant over s. Note that the constant Darboux vector means this orientation element is modeled as a helix. However our discretization is different from super-helix [BAC*06] in that helices are an internal representation to model bending and twisting; they are not used to represent outer shape of the rod. In fact, our helices are not connected to each other.

Because the basis $\mathbf{d}^i$ is rotated around the axis vector $\mathbf{n}$, the angle between the axis vector and the basis does not change. Hence, the inner product of the two vectors $\mathbf{n}$ and $\mathbf{d}^i(s)$ is constant with s (see Fig. 4-right). Thus, the material coordinate of the Darboux vector is also constant as

$$\omega_i(s) = \frac{1}{\bar{l}}\mathbf{\Theta}\cdot\mathbf{d}_e^i = \frac{1}{\bar{l}}\mathbf{\Theta}\cdot\mathbf{d}_{e+1}^i, \qquad i = 1,2,3. \tag{6}$$

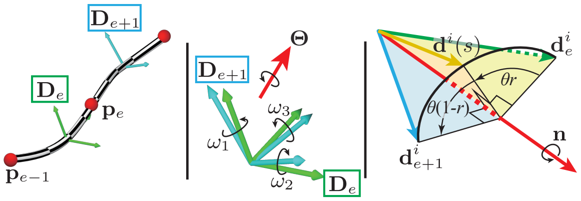

Figure 4: ***Left:*** *configuration of points and frames.* ***Middle:*** *the rotation between neighboring frames.* ***Right:*** *rotation of basis around an axis.*

3.3. Modified frame twist representation

So far, we described a straightforward way to obtain the Darboux vector using axis-angle parameterization ($\mathbf{R} \rightarrow \mathbf{\Theta}$). This parameterization involves inverse trigonometric functions and is computationally intensive for PBD, where constraint application may be iterated hundreds of times within a single time step. In this section, we further simplify our model - while increasing robustness - by modifying the axis-angle parameterization $\mathbf{\Theta}$ of the rotational matrix $\mathbf{R}$.

To obtain a better parameterization of $\mathbf{\Theta}$, we start from Rodrigues' rotation formula, which describes the relationship between the rotation matrix and axis-angle parameter:

$$\mathbf{R} = \mathbf{I} + [\mathbf{n}]\sin\theta + (1-\cos\theta)(\mathbf{n}\mathbf{n}^T - \mathbf{I}), \tag{7}$$

where $\mathbf{I}$ is an identity matrix. To obtain $\cos\theta$, we take the trace of both sides of (7), giving

$$1+2\cos\theta = \mathrm{tr}\mathbf{R} = \sum_{n=1}^{3} \mathbf{d}_e^n \cdot \mathbf{d}_{e+1}^n. \tag{8}$$

Here we use the relationship $\mathrm{tr}(\mathbf{XY}) = \mathrm{tr}(\mathbf{XY})$, which is applicable for square matrices $\mathbf{X}$ and $\mathbf{Y}$. Let "vect$(\cdot)$" be an operator $\mathbb{R}^{3\times 3} \mapsto \mathbb{R}^3$ which maps a matrix to an axis vector which is the anti-symmetric part of the matrix (i.e., $\mathrm{vect}(\mathbf{A}) = 1/2\,(A_{32}-A_{23}, A_{13}-A_{31}, A_{21}-A_{12})$). Using Rodrigues' parameter [MLSS94], rotation $\mathbf{R}$ can be parameterized with 3-vector $\boldsymbol{\Psi}$ as

$$\boldsymbol{\Psi} = 2\mathbf{n}\tan\frac{\theta}{2} = \frac{4\mathrm{vect}\,(\mathbf{R})}{1+\mathrm{tr}\mathbf{R}} = \frac{2\sum_{k=1}^{3}\mathbf{d}_e^k \times \mathbf{d}_{e+1}^k}{1+\sum_{n=1}^{3}\mathbf{d}_e^n \cdot \mathbf{d}_{e+1}^n}. \tag{9}$$

Here we use a trigonometric identity $\tan\theta/2 = \sin\theta/(1+\cos\theta)$. Note that Rodrigues' parameter $\boldsymbol{\Psi}$ have the same orientation $\mathbf{n}$ as $\boldsymbol{\Theta}$, but the parameter has a different magnitude $|\boldsymbol{\Psi}| = 2\tan(\theta/2)$.

Material coordinates of the Darboux vector. In our Cosserat-based formulation, we interchange axis angle vector $(\mathbf{n}\theta)$ with Rodrigues' parameter $\boldsymbol{\Psi}$ in the Darboux vector as $\boldsymbol{\omega}(s) \simeq \boldsymbol{\Psi}/\bar{l} \equiv \boldsymbol{\Omega}(s)$. We call this approximated Darboux vector the *modified discrete Darboux vector*, denoted by $\boldsymbol{\Omega}$. This modification may affect the quantitative accuracy of our position-based elastic rod, but brings us closer to our goal of simplicity and robustness.

There are four motivations for using this approximation. First, Rodrigues' parameter $\boldsymbol{\Psi}$ is a good approximation of $\boldsymbol{\Theta}$ if the rotation angle θ is small. Second, the $\boldsymbol{\Psi}$ is easy to compute from the coordinate bases without using any trigonometric functions or their inverse. Third, as mentioned in [BWR*08], the material coordinate of the modified discrete Darboux vector Ω_k goes to infinity when the rod is kinked completely $(\theta = \pi)$. In such a case, the energy also goes to infinity, as it tries to push the rod back to the original shape. Finally, since the magnitude $2\tan(\theta/2)$ is a monotonically increasing convex function, this makes the relative rotation at $V = 0$ unique. In other words, the Rodrigues' rotation parameter is a bijective map to the rotation matrix. This ensures that there are no kinked rest shapes other than the initial shape.

From (9), the local coordinate of the modified discrete Darboux vector $\Omega_i(s) = \boldsymbol{\Omega}(s)\cdot\mathbf{d}^i(s)$, $(i = 1,2,3)$ can be written as

$$\Omega_i = \left(\frac{4}{\bar{l}}\right)\frac{\mathrm{vect}(\mathbf{Q})_i}{1+\mathrm{tr}\mathbf{Q}} = \left(\frac{2}{\bar{l}}\right)\frac{\mathbf{d}_e^j\cdot\mathbf{d}_{e+1}^k - \mathbf{d}_e^k\cdot\mathbf{d}_{e+1}^j}{1+\sum_{n=1}^{3}\mathbf{d}_e^n\cdot\mathbf{d}_{e+1}^n}, \tag{10}$$

where $\mathbf{Q} = (\mathbf{D}_e)^T\mathbf{D}_{e+1}$ and indexes i,j,k permute as $\{i,j,k\} = \{1,2,3\},\{2,3,1\},\{3,1,2\}$. Note that $\mathbf{R}$ and $\mathbf{Q}$ are both rotation matrices from $\mathbf{D}_e$ to $\mathbf{D}_{e+1}$, but they are different. While $\mathbf{R}$ is rotation in the global coordinate, $\mathbf{Q}$ is relative rotation describing rotation to $\mathbf{D}_{e+1}$ observed from the frame $\mathbf{D}_e$. Since the coordinate of the modified discrete Darboux vector is also constant over an orientation element, the bending and twisting energy on the orientation element e can be written as

$$V_e = \frac{\bar{l}_e}{2}\sum_{i=1}^{3}K_{ii}(\Omega_i - \bar{\Omega}_i)^2. \tag{11}$$

Note that in our formula, the rods' bending and twisting energy are computed exactly for both bending and twisting components. Hence, the bending and the twisting stiffness parameters are completely decoupled. In other words, we can make a rod that has zero twisting resistance with high bending stiffness or vice versa. This gives the computer animation artist fine-grained control over the rods' behavior.

4. Variational interpretation of constraint enforcement

In the PBD framework, elasticity is formulated via constraints. These constraints take zero values for the rest shape and increase if strain is applied. In each time step, the particle is updated in the gradient direction of the constraint, representing the internal force pushing object back to the rest shape. Let each constraint be written in the form $\mathbf{C}(\mathbf{p}) = 0$, where $\mathbf{p}$ is a concatenation of the positions of points involve in this constraint. We denote $\mathcal{A}$ as the set of points involving in this constraint, and we assume the constraint is a M-dimensional function. While the constraint function is scalar in the original PBD formula, here we want generalize it for multi-dimensional function. For this purpose, we introduce a variational formula for the constraint enforcement.

In the classical mechanics, the points trajectory under constraints is known to comply the Gauss's principle of least constraint [GJS01]. This principle states that the trajectories of constrained points under gravity $\mathbf{g}$ minimize a quantity $Z = \sum_{a\in\mathcal{A}} m_a\,|\ddot{\mathbf{p}}_a - \mathbf{g}|^2$, where $\ddot{\mathbf{p}}$ is a points' accelerations. Because the $\ddot{\mathbf{p}}_a - \mathbf{g}$ means how much the constraint changed the accelerations, the principle means constrained motion takes a path that minimizes the sum of acceleration change caused by constraints. Batty et al. [BBB07] leverage this variational principle for incompressible constraint around coupling boundary of fluid against rigid bodies.

Let us explain how the Gauss's principle unfolds for the PBD time integration scheme. Let $\mathbf{p}^t$ and $\mathbf{v}^t$ are positions and velocities of points at time t and Δt is the time step. In the PBD scheme [MHHR07], the positions in the next time step can be computed as: $\mathbf{p}^{t+\Delta t} = \mathbf{p}^t + \Delta t(\mathbf{v}^t + \Delta t\mathbf{g}) + \Delta\mathbf{p}$, where the $\Delta\mathbf{p}$ is the position updates by constraint enforcement. Hence, the velocity of points becomes $\mathbf{v}^{t+\Delta t} = (\mathbf{p}^{t+\Delta t} - \mathbf{p}^t)/\Delta t = \mathbf{v}^t + \Delta t\mathbf{g} + \Delta\mathbf{p}/\Delta t$. Finally, the acceleration of a point can be written as

$$\ddot{\mathbf{p}} = (\mathbf{v}^{t+\Delta t} - \mathbf{v}^t)/\Delta t = \Delta\mathbf{p}/\Delta t^2 + \mathbf{g}. \tag{12}$$

By plugging in (12) in the Gauss's principle, we obtain the following variational formula for the position update:

$$\Delta\mathbf{p} = \arg\min_{\Delta\mathbf{p}} \sum_{a\in\mathcal{A}} m_a\,|\Delta\mathbf{p}_a|^2\,,\ \ where\ \mathbf{C}(\mathbf{p}+\Delta\mathbf{p}) = 0. \quad (13)$$

This means the position update minimizes weighted sum of its squares while satisfying the constraints. This QP problem can be easily solved by introducing a Lagrange multiplier $\boldsymbol{\lambda}\in\mathbb{R}^M$. First, we add $\boldsymbol{\lambda}^T\mathbf{C}$ to the minimization target, which has to have zero derivative at the solution:

$$\nabla_b\left(\sum_{a\in\mathcal{A}} m_a\,|\Delta\mathbf{p}_a|^2 + \boldsymbol{\lambda}^T\mathbf{C}\right) = 0\ \ (\forall b\in\mathcal{A}). \quad (14)$$

Here ∇_b means derivative with respect to the position of point b. Then, with a little calculation exercise, we get the point update using the Lagrange multiplier as:

$$\Delta\mathbf{p}_a = -0.5 w_a \boldsymbol{\lambda}^T\nabla_a\mathbf{C}, \quad (15)$$

where w_a is a inverse of the point weight. Nest, we put this (15) into linearized constraint

$$\mathbf{C} + \nabla\mathbf{C}\Delta\mathbf{p}\ =\ 0, \quad (16)$$

to solve the Lagrange multiplier. Finally, we obtain the final position update:

$$\Delta\mathbf{p}_a = (w_a\nabla_a\mathbf{C})\left(\sum_{b\in\mathcal{A}} w_b\nabla_b\mathbf{C}^T\nabla_b\mathbf{C}\right)^{-1}\mathbf{C}. \quad (17)$$

Note that when the mass of all points are equal, this update is equivalent to multiplying the $\mathbf{C}$ with Moore-Penrose matrix inverse of the matrix $\nabla\mathbf{C}$. This formula is equal to the original one when the constraint is a scalar function ($M = 1$) and thus, is the natural extension of the original formula for solving multiple constrains at the same time. In other words, the original PBD approach solves constraint in the Gauss-Seidel manor, but we solved in the block Gauss-Seidel manor, which gives better convergence if there is a block structure in the coefficient matrix.

This position update preserves the linear and angular momentum. The constraints are chosen to be invariant to the rigid transformation of points (i.e., its value doesn't change when the points are translated and rotated). Hence, the gradient of constraint $\nabla\mathbf{C}$ is perpendicular to the null space of the constraint. As long as we update the positions in the gradient space as (15), each constraint enforcement conserves translational and rotational momentum, and consequently, the sequence of constraint enforcements also consequently do.

We have also tried a different approach to handle the multi-dimensional constraint. Inspired by the penalty-based method, we create a new scalar constraint by taking square norm of the multi-dimensional constraint ($C = \mathbf{C}^T\mathbf{C}$). This approach was also functional, because if the constraint is zero, the square norm becomes also zero. However, since the square norm is quadratic, the linearization of the constraint is not a good approximation. Thus, more iterations are required to achieve a rod with the same stiffness.

5. Constraints for elastic rod simulation

Based on the previous sections, we construct constraints suitable for elastic rod simulation. Please refer to the accompanying supplemental document for differentiations of these constraint functions.

Edge constraints. We consider three constraints for each edge. The first one is an edge length constraint

$$C_e^L(\mathbf{p}_{e-1},\mathbf{p}_e) = |\mathbf{p}_{e-1}-\mathbf{p}_e| - \bar{L}_e, \quad (18)$$

where $\bar{L}_e$ is the initial length of edge e. This constraint keeps the length of the edge similar to the rest length. The second constraint enforces that the ghost point lies on the edge's perpendicular bisector

$$C_e^P(\mathbf{p}_{e-1},\mathbf{p}_e,\mathbf{p}_e^g) = (\mathbf{p}_e^g - \mathbf{p}_{e-1/2})^T(\mathbf{p}_e-\mathbf{p}_{e-1}), \quad (19)$$

where $\mathbf{p}_{e-1/2} = (\mathbf{p}_{e-1}+\mathbf{p}_e)/2$ is the midpoint of the edge e. The third constraint enforces a constant distance $\bar{L}_e^g$ between the ghost point and the edge

$$C_e^D(\mathbf{p}_{e-1},\mathbf{p}_e,\mathbf{p}_e^g)\ =\ \left|\mathbf{p}_{e-1/2}-\mathbf{p}_e^g\right| - \bar{L}_e^g. \quad (20)$$

The rods' diameter change is usually very small, hence we enforce this constraint to prevent it.

Anisotropic Bending-twisting constraint. With these three constraints, C_e^L, C_e^P, and C_e^D, we can define coordinates at the center of edge. Next, we define bending and twisting constraints based on the material frames defined on the orientation element. The constraint tries to minimize the difference of modified Darboux vector (defined in (10)) between rest shape and current shape.

We specify a rod's anisotropic stiffness via parameters $\alpha_i\in[0,1]$ $(i = 1,2,3)$. More specifically, α_1 denotes the resistance to bending around $\mathbf{d}^1$ (the direction from midpoint to ghost point), α_2 denotes the resistance to bending around $\mathbf{d}^2$, and α_3 denotes twisting stiffness. To handle this anisotropic stiffness, we define the *goal Darboux vector* $\Omega_i^{Goal} = (1-\alpha_i)\Omega_i + \alpha_i\bar{\Omega}_i$, with $\alpha_i\in[0,1]$. This is simply an interpolation between the current Darboux vector and that of the rest shape. The bending and twisting constraint can be written as:

$$\mathbf{C}_e^{BT}(\mathbf{p}_{e-1},\mathbf{p}_e,\mathbf{p}_{e+1},\mathbf{p}_e^g,\mathbf{p}_{e+1}^g) = \begin{pmatrix}\Omega_1-\Omega_1^{Goal}\\ \Omega_2-\Omega_2^{Goal}\\ \Omega_3-\Omega_3^{Goal}\end{pmatrix}. \quad (21)$$

Note that if all the α_i are zero (zero stiffness), this constraint $\mathbf{C}$ takes value zero and has no effect on the constraint enforcement. Similarly, if all $\alpha_i = 1$, the constraint tries to match the Darboux vector to the one from the rest shape.

5.1. Endpoint constraints

So far, we have described constraints for bending and twisting deformation of rods. With a few extensions to this formulation, we can constrain the rod's orientation at its endpoints. We consider three cases where a rod is attached to a frame, to a triangle, and to a rigid body.

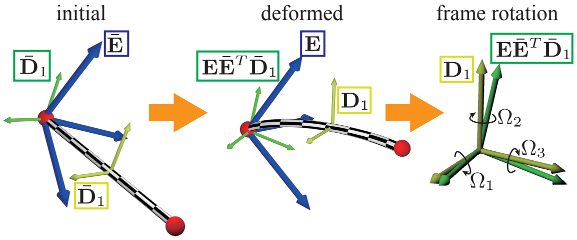

Figure 5: *Configurations of frames on a midpoint of an edge* $\mathbf{D}_1$ *and on a frame* $\mathbf{E}$*. The rod tip's orientation is firmly constrained to the frame and it moves together with frame.*

Frame attachment. Assume a rod's end point $\mathbf{p}_0$ is attached to a frame $\mathbf{E}$. The frame attachment can be interpreted as the rod being attached to an infinitely heavy rigid body. More specifically, this attachment is one-way coupling; the frame rotation is given and is not affected by the rod. As described in § 3.2, the three points $\{\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_1^g\}$ adjacent to the end make a frame $\mathbf{D}_1$. We assume the rod's tip direction is fixed to the frame. Hence, when the frame rotates from $\bar{\mathbf{E}}$ to $\mathbf{E}$, the direction of the tip rotates from $\bar{\mathbf{D}}_1$ to $\mathbf{E}\bar{\mathbf{E}}^T\bar{\mathbf{D}}_1$. In the deformed state, the rod's frame changes from $\mathbf{E}\bar{\mathbf{E}}^T\bar{\mathbf{D}}_1$ to $\mathbf{D}_1$ in the length of $0.5\bar{l}_1$. We compute coordinates of the modified Darboux vector Ω_i between the frame $\mathbf{E}$ and $\mathbf{D}_1$ as

$$\Omega_i(s) = \left(\frac{4}{0.5\bar{l}_1}\right)\frac{\mathrm{vect}(\mathbf{P})_i}{1+\mathrm{tr}\mathbf{P}} \quad i = 1,2,3, \tag{22}$$

where $\mathbf{P} = (\mathbf{E}\bar{\mathbf{E}}^T\bar{\mathbf{D}}_1)^T\mathbf{D}_1$, which encodes the amount of twisting at the tip in the material coordinate of the rod. Note that Ω_i is zero in the initial rest configuration. Thus, the constraint for keeping the relative orientation between the triangle and rod identical can be written as

$$C_f^O(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_1^g) = (\alpha_1\Omega_1, \alpha_2\Omega_2, \alpha_3\Omega_3)^T. \tag{23}$$

where α_i are the same parameter as the rod's stiffness. Note that to represent the initial rest configuration, a rotation matrix $\bar{\mathbf{E}}^T\bar{\mathbf{D}}_1$ is stored for the tip constraints, while three local coordinates of the Darboux vector $\bar{\Omega}_i$ are stored for twisting and bending constraints. The position constraint for a tip of a rod $\mathbf{p}_0$ can be simply written as

$$C_f^T(\mathbf{p}_0^t) = |\mathbf{q} - \mathbf{p}_0|, \tag{24}$$

where the $\mathbf{q} \in \mathbb{R}^3$ is the position where the tip is attached.

Triangle attachment. The attachment to a triangle can be handled similarly to the case of attachment to a frame. Here, we consider two ways coupling between rod and triangle; the triangle's movement is influenced by the rod. We assume the rod tip edge is attached to a triangle t with vertices $\{\mathbf{p}_0^t, \mathbf{p}_1^t, \mathbf{p}_2^t\}$ as shown in Fig. 6. We define coordinate bases $\mathbf{E}_t = \{\mathbf{e}_t^1, \mathbf{e}_t^2, \mathbf{e}_t^3\}$ as

$$\begin{cases} \mathbf{e}_t^1 &= (\mathbf{p}_1^t - \mathbf{p}_0^t)\ /\ \left|\mathbf{p}_1^t - \mathbf{p}_0^t\right|, \\ \mathbf{e}_t^3 &= \mathbf{e}_t^1 \times (\mathbf{p}_2^t - \mathbf{p}_0^t)\ /\ \left|\mathbf{e}_t^1 \times (\mathbf{p}_2^t - \mathbf{p}_0^t)\right|, \\ \mathbf{e}_t^2 &= \mathbf{e}_t^3 \times \mathbf{e}_t^1. \end{cases} \tag{25}$$

The position where the end point $\mathbf{p}_0$ is attached to a triangle can be written as $\mathbf{q}^t = w_0\mathbf{p}_0^t + w_1\mathbf{p}_1^t + w_2\mathbf{p}_2^t$, where w_0, w_1, w_2 are the barycentric coordinates for $\mathbf{p}_0$ in the triangle in its initial configuration. By plugging in $\mathbf{E}^t$ and $\mathbf{q}^t$ into the frame attachment constraint, we can convert it to a triangle attachment constraint. Note that as this is two-way coupling constraint; three triangle vertices $\{\mathbf{p}_0^t, \mathbf{p}_1^t, \mathbf{p}_2^t\}$ and three rod vertices $\{\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_1^g\}$ are involved.

Rigid body attachment. With the triangle constraint, we can attach the end of a rod to an object. To represent networks of rods, such as trees and nets, we put a small rigid body at the joint where multiple rod join together. Each rod is connected to this rigid body to transmit torque between rods. We represent the frame of this rigid body attached at the tip of the rod using two additional ghost points. At the endpoint of the rod $\mathbf{p}_0$, we put ghost points $\{\mathbf{p}_0^{g_1}, \mathbf{p}_0^{g_2}\}$, so that the two vectors $\mathbf{p}_0^{g_1} - \mathbf{p}_0$ and $\mathbf{p}_0^{g_2} - \mathbf{p}_0$ are perpendicular. With this ghost point, the frame of the rigid body $\mathbf{E}_r = \{\mathbf{e}_r^1, \mathbf{e}_r^2, \mathbf{e}_r^3\}$ at the end of a rod can be written as

$$\begin{cases} \mathbf{e}_r^1 &= (\mathbf{p}_0^{g_1} - \mathbf{p}_0)\ /\ \left|\mathbf{p}_0^{g_1} - \mathbf{p}_0\right|, \\ \mathbf{e}_r^3 &= \mathbf{e}_r^1 \times (\mathbf{p}_0^{g_2} - \mathbf{p}_0)\ /\ \left|\mathbf{e}_r^1 \times (\mathbf{p}_0^{g_2} - \mathbf{p}_0)\right|, \\ \mathbf{e}_r^2 &= \mathbf{e}_r^3 \times \mathbf{e}_r^1. \end{cases} \tag{26}$$

By plugging $\mathbf{E}^r$ into the frame attachment constraint, we can attach a rigid body at the tip of the rod. The orthogonality and length constraints for $\mathbf{p}_0^{g_1} - \mathbf{p}_0$ and $\mathbf{p}_0^{g_2} - \mathbf{p}_0$ are enforced as we did for ghost point on edges in (19) and (20).

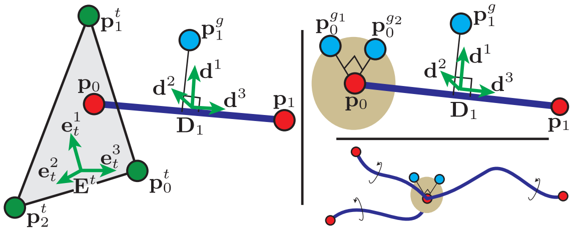

Figure 6: ***Left:*** *configuration of points and frames in a triangle attachment.* ***Upper-right:*** *configuration of points in a small rigid-body attachment.* ***Lower-right:*** *Rigid body attachment is used for a network of rods.*

6. Implementation details

Resolving order-induced instability. In the FEM approach, multiple constraints are simultaneously enforced by solving a linear system. In contrast, position-based frameworks sequentially enforce constraints one-by-one, and as a result, the simulation can be affected by the order of application. Since a rod is one dimensional, a natural ordering is naïve sequential order (i.e., enforcing constraints from one end to the other). However, we found that with this ordering the rod vibrates on its own, even without any external forces and starting the simulation from the rest shape.

In position-based frameworks, each enforcement of a constraint takes the form of internal force. Enforcing the rod's twist constraint in sequential order is similar to an elastic wave propagating inside the rod. The rod has many waving modes where the elastic wave goes back and forth in the strand and the sequential constraint enforcement can excite such vibrations. Note that this artifact related to the constraint order is specific to rod simulation, because in solids or shells the order is generally randomly distributed.

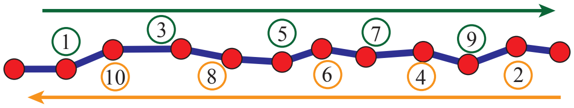

Figure 7: *Bilateral interleaving ordering.*

Our solution to prevent order-dependent instability is to utilize a *bidirectional interleaving order* to apply constraints. As shown in Fig. 7, we interleave the order for both directions of the rod. Applying constraints from two directions at the same time cancels the vibration modes, producing stable results without excessive energy gain.

Applying constraints in a random order also gives a good result that is stable without energy gain. However, we observed a tiny flicker in the high-torsional strain arising from random permutations, especially for curved rods. Generally, the interleaving permutation gives better results.

Initial position of ghost points. The ghost points on edges are placed so that the frame axes $\mathbf{d}^1, \mathbf{d}^2$ correspond to the principal axis of the cross section. For example, if we model bending of a thin planar rod, such as a ribbon, one of the axes is chosen to align with the normal direction of the plate. This will give the expected anisotropic behavior, where the rod is stiff in one direction and soft in the other.

In the case of an isotropic rod (i.e., a rod with a circular cross section), the material frame can be chosen arbitrarily. But ideally, the initial frame should be chosen such that the Darboux vector becomes as small as possible. If the frame rotation between two edges is 180°, the discrete Darboux vector become singular. Hence, we initialize the initial ghost position using parallel transport [BWR*08], so that the material frames become twist-free. If there are loops in the rod topology, we relax the material frame after parallel transport, so that there are no excessive jumps between frames. For this relaxation, we average two material frames at the two edges of an orientation element iteratively for every orientation element inside the rod.

Gravity on ghost points. We assume that material is distributed uniformly around an edge. Hence, the center of gravity of an edge is located at its mid-point and the gravity force should be applied there. However, in our discretization, the ghost point is located off the axis of an edge, and so we need to cancel the spurious rotational moment that would be added to the edge.

We address this problem with a simple heuristic. We tweak the amount of gravitational force on a ghost point such that the gravitational acceleration becomes equal to that of the mid point. We compute acceleration at the mid-point by forward differentiation of its velocities: $\mathbf{a}_m = (\mathbf{v}_m^t - \mathbf{v}_m^{t-\Delta t})/\Delta t$. Here $\mathbf{v}_m^t$ is the velocity of the mid-point at time t, computed as the average velocity of the end points. After enforcing constraints and updating velocities, we compute the ratio $r = (\mathbf{a}_m \cdot \mathbf{g})/|\mathbf{g}|^2$, which indicates how much the mid-point was accelerated by gravity. Finally, we re-distribute the gravitational forces of the ghost points to the edge points such that the total gravitational force applied to the ghost point is in the ratio r against gravity. Please refer to the accompanying supplemental document for more detailed algorithm and the effect of our gravity force modification compared to other approaches.

7. Results

To demonstrate our twist model, we shows an elastic rod pendulum in Fig. 8-left. The rod is connected to a fixed frame on one side (the frame attachment), and is connected to a cube modeled as a triangle mesh on another side (the triangle attachment). After the cube hits another object, the rod is twisted and then recovers. The twist resistance is achieved by the constraints in equations (21) and (23), which are based on Cosserat theory. Fig. 8-right demonstrates stability and resistance to flipping with large twist angles. Please refer to the accompanying video for more detail.

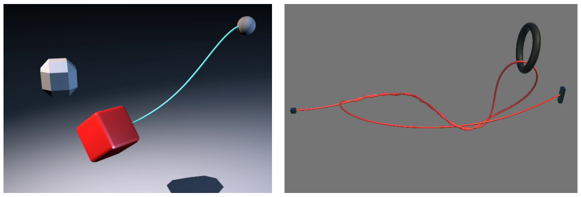

Figure 8: ***Left***: *elastic pendulum.* ***Right***: *rod under large twist.*

In Fig. 9, inspired by [BWR*08], we create a ring and apply twist to the ring. As expected, the ring buckles and twists into a figure-8 pattern, similar to the results shown in [BWR*08]. In this experiment we also tested anisotropic rods, where the cross section is significantly larger in one direction than the other (e.g., a ribbon). As twist is applied to the ring-shaped ribbon, it first buckles similar to the isotropic ring, but then because it can bend only in one direction, it shows a different deformation result.

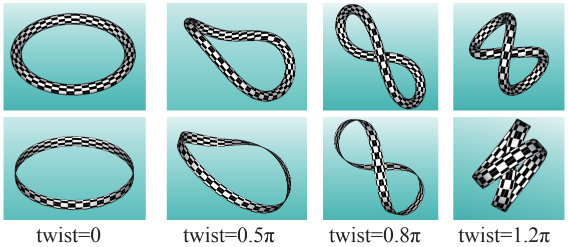

Figure 9: *Twisting ring with (**upper**) isotropic cross-section and (**lower**) anisotoropic cross-section.*

Comparison with previous work. We compare our rod formulation with two previous PBD rod simulation techniques. The first is oriented particles [MC11], where the torsion resistance is modeled by averaging neighboring oriented particles, and boundary twist is given by explicitly specifying rotation matrices at the end points. The second is Chain Shape Matching (CSM) [RKM*12], which applies a twisting force as an external force, on the top of a shape matching deformation. Using these three methods, we simulate twist deformations of a straight and spring rod (discretized with 30 edges) with both ends clamped. In our PBD-based rod, twist and bending constraints are formulated using two neighboring edges. To compare fairly, we choose two neighboring edges as a cluster of shape matching operations in our CSM and oriented particle implementations. We apply the same number of constraint enforcement iterations (ten iterations) for these three methods. Since we wish to compare the deformations originating purely from elasticity, we did not include self-contact and damping in the comparisons.

Fig. 10 shows the comparison result. The oriented particles method gives stable deformation but it lacks a twisting effect because the method doesn't model coupling of twisting and bending. The CSM shows twisting effects but produces unstable deformations and doesn't converge into a static rest shape. This is because the twisting force is applied as an external force, not as a constraint. Generally, the external forces in shape matching work only for simple potential forces such as gravity, and not for elastic forces. Furthermore, CSM is significantly softer than other models because of the singularity in the momentum matrix, which is solved in the oriented particle model by introducing rotational inertia. Our rod deforms stably and plausibly in both examples. We also note that our PBD approach is faster than previous shape matching approaches because PBD doesn't require the costly polar decompositions.

Robustness. Fig. 11 demonstrates the robustness of our rod model. Starting with an configuration with random vertex placement, the simulation quickly recovers, and within a small number of frames, has returned to the rest shape.

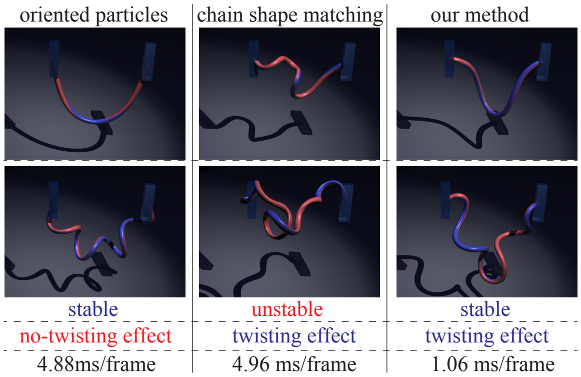

Figure 10: *Twisting (**upper**) straight and (**lower**) spring rod simulated with different techniques. The bottom row gives computational time per frame for each method.*

Fig. 1 and 12 demonstrate some complex examples created with our rod, implemented in a widely-used visual effects framework. These examples include large numbers of rods interacting and colliding with each other. Please see the accompanying video to observe the stability of our model in these complex situations. In the tentacle-ball example (Fig. 1), the sticky behavior is achieved with large friction forces. The core ball itself uses a dynamic position-based cloth model, which is coupled with our rods using the triangle attachment constraint. In the wire-mesh example (Fig. 12), the wire mesh is created by weaving rods in an over/under configuration. With our bending and twisting constraint, the mesh is stiff enough to be prevented from separating or unraveling, maintaining the woven configuration.

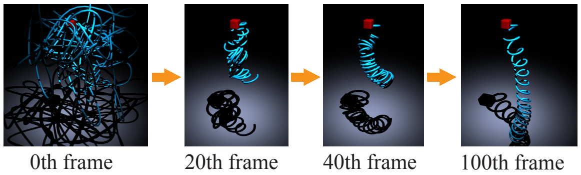

Figure 11: *Starting from random initial configuration, the rod quickly converges to the original shape stably.*

Interactive hair design. Since our PBD simulation is fast and robust, it can be used inside interactive modeling tools. We implemented our PBD model with a simple collision model in a 3D modeling tool, to allow the user to create hair volumes. Because our model is robust, we can allow the user to interactively manipulate rest shape of the hair, such as length and twisting of strands.

In addition to designing hair for virtual applications, this tool is useful to create solid hair for 3D printing purposes. Fig. 13 shows a simple example where we designed a hairstyle for the bunny model, and the 3D-printed result. Please see the video for more examples, including a demon-

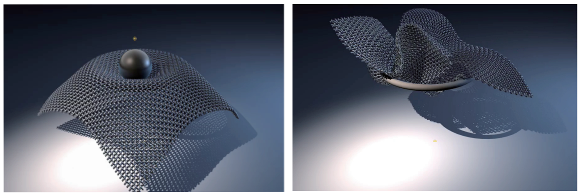

Figure 12: *A wire mesh modeled by our position-based elastic rods. The wire mesh goes through a ring pushed by a sphere. All sequence was solved robust and stably.*

stration showing over 200 hairs, each is modeled with 20 edges, being simulated at interactive rates.

8. Discussion and Limitations

We present position based elastic rods, which can simulate a rod's bending and twisting deformation efficiently using the framework of position based dynamics.

We remove the order-induced instability of constraint enforcement for rods by introducing a special ordering. Furthermore, we remove artificial rotational momentum induced by the gravity on ghost points. While these methods works perfectly for various cases, there are no theoretical proofs. Guaranteeing stability is one of our future works.

Our variation interpretation of constraint enforcement enables a handling multi-dimensional constraint that is specifically useful to modeling deformation of anisotropic materials. We are planning to apply this concept for anisotropic solid and shell deformation simulation. Furthermore, handling wider variety of constraints such as ones shown in [SJLP11] is an interesting future work.

Although our bending and twisting constraints prevent over 180° twisting via the Rodriguez parameter, under high torsional circumstances the rod may still exceed this angle limit. This is because we apply partial constraints sequentially. In future work, we will explore additional constraints to prevent this flip completely.

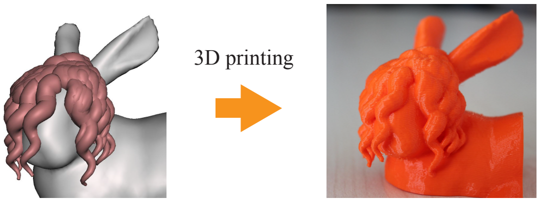

Figure 13: ***Left:*** *Designed bunny hair.* ***Right:*** *3D print.*

Acknowledgement. We thank Duncan Brinsmead for helping us preparing animations.

References

[Ant05] ANTMAN S. S.: *Nonlinear problems of elasticity*, vol. 107. Springer, 2005. 3

[BAC*06] BERTAILS F., AUDOLY B., CANI M.-P., QUERLEUX B., LEROY F., LÉVÊQUE J.-L.: Super-helices for predicting the dynamics of natural hair. *ACM TOG 25*, 3 (2006). 2, 4

[BAV*10] BERGOU M., AUDOLY B., VOUGA E., WARDETZKY M., GRINSPUN E.: Discrete viscous threads. *ACM TOG 29*, 4 (2010). 2

[BBB07] BATTY C., BERTAILS F., BRIDSON R.: A fast variational framework for accurate solid-fluid coupling. *ACM TOG 26*, 3 (2007). 5

[BMOT13] BENDER J., MÜLLER M., OTADUY M. A., TESCHNER M.: Position-based methods for the simulation of solid objects in computer graphics. In *EG STARs* (2013). 3

[BWR*08] BERGOU M., WARDETZKY M., ROBINSON S., AUDOLY B., GRINSPUN E.: Discrete elastic rods. *ACM TOG 27*, 3 (2008). 1, 2, 3, 5, 8

[CBD13] CASATI R., BERTAILS-DESCOUBES F.: Super space clothoids. *ACM TOG 32*, 4 (2013). 2

[CC07] COSSERAT E., COSSERAT F.: Sur la statique de la ligne déformable. *Académie des Sciences 145* (1907), 1409–1412. 2

[GJS01] GOLDSTEIN H., JR. C. P. P., SAFKO J. L.: *Classical Mechanics (3rd Edition)*, 3 ed. Addison-Wesley, 6 2001. 5

[Jak01] JAKOBSEN T.: Advanced character physics. In *Game Developers Conference* (2001). 3

[KJM10] KALDOR J. M., JAMES D. L., MARSCHNER S.: Efficient yarn-based cloth with adaptive contact linearization. *ACM TOG 29*, 4 (2010). 2

[MC11] MÜLLER M., CHENTANEZ N.: Solid simulation with oriented particles. *ACM TOG 30*, 4 (2011). 3, 9

[MHHR07] MÜLLER M., HEIDELBERGER B., HENNIX M., RATCLIFF J.: Position based dynamics. *Visual Communication and Image Representation 18*, 2 (2007). 1, 3, 5

[MHTG05] MÜLLER M., HEIDELBERGER B., TESCHNER M., GROSS M.: Meshless deformations based on shape matching. *ACM TOG 24*, 3 (2005). 3

[MKC12] MÜLLER M., KIM T.-Y., CHENTANEZ N.: Fast simulation of inextensible hair and fur. In *VRIPHYS* (2012). 3

[MLSS94] MURRAY R. M., LI Z., SASTRY S. S., SASTRY S. S.: *A Mathematical Introduction to Robotic Manipulation*, 2nd ed. CRC Press, 3 1994. 5

[MM13] MACKLIN M., MÜLLER M.: Position based fluids. *ACM TOG 32*, 4 (2013). 3

[MSW*09] MCADAMS A., SELLE A., WARD K., SIFAKIS E., TERAN J.: Detail preserving continuum simulation of straight hair. *ACM TOG 28*, 3 (2009). 2

[Pai02] PAI D. K.: Strands: Interactive simulation of thin solids using cosserat models. *CGF 21*, 3 (2002). 2

[RKM*12] RUNGJIRATANANON W., KANAMORI Y., METAAPHANON N., BANDO Y., CHEN B.-Y., NISHITA T.: Animating strings with twisting, tearing and flicking effects. *Computer Animation and Virtual Worlds 23*, 2 (2012). 3, 9

[SJLP11] SUEDA S., JONES G. L., LEVIN D. I. W., PAI D. K.: Large-scale dynamic simulation of highly constrained strands. *ACM TOG 30*, 4 (2011). 10

[SLF08] SELLE A., LENTINE M., FEDKIW R.: A mass spring model for hair simulation. *ACM TOG 27*, 3 (2008). 2

[ST07] SPILLMANN J., TESCHNER M.: CoRdE: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects. In *ACM SCA* (2007). 1, 2, 4

[Sta09] STAM J.: Nucleus: Towards a unified dynamics solver for computer graphics. In *CAD/Graphics 2009* (2009), pp. 1–11. 3