

Fast Physics-Based Modeling of Knots and Ties Using Templates

DEWEN GUO, Peking University, China and Style3D Research, China

ZHENDONG WANG*, Style3D Research, China

ZEGAO LIU, Style3D Research, China

SHENG LI, Peking University, China

GUOPING WANG*, Peking University, China

YIN YANG, University of Utah, USA

HUAMIN WANG, Style3D Research, China



(a) Trench coat

(b) Hoodie

(c) Knotted blouse

(d) Knotted shirt & ribbon

(e) Knotted shirt

(f) Dress shirt

Fig. 1. **Garment examples.** We use these six garment examples to evaluate our system. While some examples, as shown in (a), (b), and (f), feature knots and ties made from fabric patches separate from the garments, others, illustrated in (c), (d), and (e), showcase knots created directly on the garments.

Knots and ties are captivating elements of digital garments and accessories, but they have been notoriously challenging and computationally expensive to model manually. In this paper, we propose a physics-based modeling system for knots and ties using templates. The primary challenge lies in transforming cloth pieces into desired knot and tie configurations in a controllable, penetration-free manner, particularly when interacting with surrounding meshes. To address this, we introduce a pipe-like parametric knot template representation, defined by a Bézier curve as its medial axis and an adaptively adjustable radius for enhanced flexibility and variation. This representation enables customizable knot sizes, shapes, and styles

while ensuring intersection-free results through robust collision detection techniques. Using the defined knot template, we present a mapping and penetration-free initialization method to transform selected cloth regions from UV space into the initial 3D knot shape. We further enable quasistatic simulation of knots and their surrounding meshes through a fast and reliable collision handling and simulation scheme. Our experiments demonstrate the system's effectiveness and efficiency in modeling a wide range of digital knots and ties with diverse styles and shapes, including configurations that were previously impractical to create manually.

CCS Concepts: • Computing methodologies → Physical simulation; Mesh geometry models.

Additional Key Words and Phrases: cloth simulation, intersection-free, physics-based modeling

ACM Reference Format:

Dewen Guo, Zhendong Wang, Zegao Liu, Sheng Li, Guoping Wang, Yin Yang, and Huamin Wang. 2025. Fast Physics-Based Modeling of Knots and Ties Using Templates. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25)*, August 10–14, 2025, Vancouver, BC, Canada. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3721238.3730622>

*Corresponding authors

Authors' Contact Information: Dewen Guo, Peking University, Beijing, China and Style3D Research, Hangzhou, China, guowen@pku.edu.cn; Zhendong Wang, Style3D Research, Hangzhou, China, wang.zhendong.619@gmail.com; Zegao Liu, Style3D Research, Hangzhou, China, liuzegao@linctex.com; Sheng Li, Peking University, Beijing, China, lisheng@pku.edu.cn; Guoping Wang, Peking University, Beijing, China, wgp@pku.edu.cn; Yin Yang, University of Utah, Salt Lake City, USA, yangzzzy@gmail.com; Huamin Wang, Style3D Research, Hangzhou, China, wanghmin@gmail.com.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGGRAPH Conference Papers '25, Vancouver, BC, Canada

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 979-8-4007-1540-2/2025/08
<https://doi.org/10.1145/3721238.3730622>

1 Introduction

Knots and ties are prominent elements in garments and accessories, valued for both their functional and decorative roles. They are formed by intertwining one or more strips of fabric, resulting in intricate interactions and intensive collision handling at their core. Even

when based on the same fundamental structure, such as the bow knot shown in Fig. 12, knots and ties can exhibit diverse shapes due to variations in tightness and allowances. This inherent complexity and variability pose challenges in crafting visually appealing knots and ties, even when done manually in the real world. Replicating this process virtually is even more demanding.

A common approach in virtual design is to pre-model a library of digital knots and ties, which can then be stitched onto garments as needed. However, such datasets are inherently limited in capturing the vast range of possible variations or enabling the dynamic creation of knots directly on garments. For instance, an overhand knot, as shown in Fig. 1d, formed directly from the bottom edge of a T-shirt, cannot be pre-modeled independently.

While recent advancements in collision handling [Lan et al. 2024; Li et al. 2021] have enabled modern simulators to model knots and ties without penetration, the challenge of shaping these elements into visually appealing forms from the outset remains largely unresolved. Unlike the tactile manipulation of fabric with fingers in the physical world, using a cursor to handle digital cloth and ropes is significantly more cumbersome. Even with additional aids like artificial fixing pins, designers often spend considerable time—ranging from minutes to hours—to achieve satisfactory results. Moreover, the simulation process is fraught with challenges: without penetration-free simulation, self-intersections occur frequently; with it, computational costs can be prohibitive. As one designer lamented on YouTube [CLO 2025]:

This is so difficult to achieve. It took me hours to get a decent result, and the knot fell apart so many times that I was close to despair...

The demand for efficient, intuitive, and reliable tools to model and simulate knots and ties is significant and pressing.

To enable the efficient and intuitive creation of digital knots and ties, we adopt a pipe-like template, as shown in Fig. 3b, with its medial axis defined by a Bézier curve representing the intertwining structure. While similar representations have proven effective in yarn-level simulation of knit fabrics [Cirio et al. 2014; Kaldor et al. 2008], knots and ties pose unique challenges. Unlike knit fabrics, which are composed of yarns with circular cross-sections, knots and ties are created from a wide variety of fabric pieces—many of which are wide, irregular, and drastically different from yarns. This complexity makes developing a modeling system particularly challenging, especially when ensuring the knots and ties are created in a penetration-free manner.

Inspired by the template-based approach and mindful of the associated challenges, we propose a physics-based system for modeling digital knots and ties using templates. The bijective mapping from the fabric piece to the knot template ensures intersection-free mapping within the template. Our research places particular emphasis on preventing the occurrence of new intersections elsewhere in the system. Specifically, our system incorporates the following technical contributions:

- **Parametric knot templates:** We introduce a novel knot template model based on a Bézier curve, incorporating variable radii. This parametric model allows users to customize



Fig. 2. Using our physics-based system for modeling knots and ties with templates, users can efficiently create a variety of knot examples, such as the bow knots (c and d) and pretzel knot (c), in just minutes. In contrast, manually arranging digital fabrics into knots is time-consuming—often taking hours—and prone to penetration issues.

the template for different knot sizes, shapes, and styles. During template adjustments, continuous collision detection is employed to identify and resolve potential self-collisions, ensuring a robust and adaptable design process.

- **Mapping and initialization:** With the tessellated knot template, we propose a mapping and initialization method to transform a selected cloth region from its UV space into the desired 3D knot shape. To address potential collisions between the mapped knot mesh and the surrounding cloth or body meshes, we introduce a gradient-based correction scheme for efficient resolution.
- **Penetration-free simulation:** To efficiently and safely simulate the knot alongside the surrounding meshes into a quasistatic state, we propose a fast intersection resolution and collision handling approach based on contour minimization. This is complemented by an IPC-based fallback scheme to ensure a penetration-free guarantee.

We implemented our system and evaluated its performance on garment and non-garment examples, as shown in Fig. 1, 2 and 10. These examples show that our system enables users to create a wide variety of knots and ties with customizable shapes, sizes, and styles through template parameters. The experiments validate the efficiency of our system in modeling knots and ties, including some that were previously difficult or impractical to create by hand. We will release our data upon the acceptance of this paper.

2 Related Work

Cloth Simulation. Physics-based cloth simulation has been a significant research focus in computer graphics since the seminal work by Baraff and Witkin [1998]. Simulation techniques are typically categorized based on how cloth is modeled: continuum models [Baraff and Witkin 1998; Narain et al. 2012], mass-spring models [Choi and Ko 2002; Liu et al. 2013], and yarn-level models [Cirio et al. 2014; Kaldor et al. 2010]. Regardless of the modeling approach, the central challenge lies in solving the underlying dynamic system over time. Martin et al. [2011] proposed formulating cloth simulation as a nonlinear optimization problem, which was later extended by Liu et

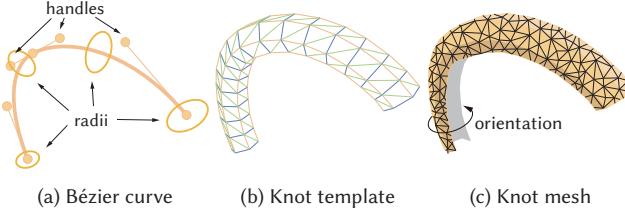


Fig. 3. Bézier curve, tessellated knot template, and knot mesh after applying mapping. This figure illustrates the key concepts used in our system.

al.[2013] and Bouaziz et al.[2014] to develop the projective dynamics framework. Narain et al.[2016] further interpreted projective dynamics using an ADMM-based formulation. Macklin et al. [2016] established the connection between position-based constraints and physics-based simulation. In recent years, research on fast cloth simulation has focused on multigrid solvers [Tamstorf et al. 2015; Wang et al. 2018; Xian et al. 2019], preconditioning techniques [Wang and Yang 2016; Wu et al. 2022], and GPU-based implementations [Lan et al. 2023, 2024; Wang 2015].

Among the many topics in cloth simulation, collision handling is particularly challenging, and it becomes even more complex when simulating the creation of knots and ties. Traditional collision handling techniques, whether based on discrete collision detection (DCD) [Baraff et al. 2003; Volino and Magnenat-Thalmann 2006; Wicke et al. 2006] or continuous collision detection (CCD) [Bridson et al. 2002; Tang et al. 2011; Wang et al. 2015], often struggle to ensure penetration-free simulations. Recent research on barrier-based collision responses has introduced novel approaches [Harmon et al. 2009; Li et al. 2020, 2021; Wang 2021] for achieving penetration-free cloth simulations, but at a significant computational cost. While our research leverages advancements in collision handling technology [Guo et al. 2024; Lan et al. 2020, 2022], it also provides a fast and penetration-free initialization of knots and ties, serving as a foundation for subsequent simulations.

Digital garment creation. Our work is also closely tied to the field of digital garment creation, a research domain that remains relatively underexplored due to the inherent complexity of digital garments. Recent advancements in physics-based cloth simulation have made efficient and automated garment creation increasingly feasible [Liu et al. 2024]. However, intricate garment-making techniques such as smocking [Ren et al. 2024; Segall et al. 2024] and kirigami [Leake et al. 2021], which involve detailed stitching and folding, remain challenging to replicate digitally. Tools like FoldSketch [Li et al. 2018] and platforms such as Crane [Suto et al. 2023] support fold modeling and 3D surface creation, while progress in fabric-thread interaction [Zhou et al. 2024] and garment arrangement [Liu et al. 2024] has improved the efficiency of detailed designs. In contrast to folds, knots and ties are more difficult to model and remain largely unexplored due to their complex structures and intensive collision requirements.

3 System Overview

The use of our system can be broken down into three steps: (i) mapping and initialization, (ii) template parameter adjustment, and (iii) fast simulation to achieve an intersection-free equilibrium.

Knot mapping and initialization. To begin, the user selects regions in the UV space of fabric pieces that will contribute to the knot. Depending on the knot type, this may involve selecting one or many regions. Once selected, these regions are mapped onto the tessellated surfaces of our knot template as shown in Fig. 3b to form a knot mesh as shown in Fig. 3c, which is then initialized to the desired position and orientation in 3D.

Template parameter adjustment. Our knot template provides a wide range of parameters for users to customize, allowing modifications to the knot’s size, shape, and style, as illustrated in Fig. 12. This is achieved by applying transformations to the control points in different parts of the template. Once the template is modified, our system enables the knot mesh to be regenerated and reinitialized in real time, allowing for quick visual feedback.

Penetration-free simulation. After the knot mesh has been mapped and initialized, we perform quasistatic simulation to settle its shape alongside the surrounding meshes. This process is susceptible to collision failures that may untangle the knot. To address this issue without relying on non-physics-based methods such as artificial fixing pins, we propose a fast primary simulator, supplemented by an IPC-based fallback to ensure penetration-free results.

Our system allows users to modify and resimulate knots in real time by repeating the above steps until they are satisfied. This capability is essential for designing digital knots, which was previously impractical due to the challenges of manual manipulation.

4 Parametric Knot Template

We begin by defining a knot template as a *pipe* and outlining the parameters required to shape knots into various configurations in Section 4.1. We will then describe the tessellation of the pipe into a triangular mesh in Section 4.2, essential for collision avoidance in Section 4.3 and mapping in Section 5.1.

4.1 Template Definition

Inspired by yarn-level simulations [Cirio et al. 2014; Kaldor et al. 2010], we represent the structure of a knot using a Bézier curve. We assume this curve has been manually created using third-party modeling software, such as Blender. To enhance flexibility, instead of assuming a constant radius, we discretize the curve into piecewise linear segments, assigning each segment node its own radius variable as shown in Fig. 3a. This approach defines the knot template as a *pipe*, consisting of pipe-like segments with spatially varying radii, with the Bézier curve serving as its medial axis.

To enable customization of template shapes, our system provides several adjustable parameters. First, users can modify the radius variable to directly control the tightness or looseness of the cloth pieces. Second, users can adjust the orientation of the cylinders within the knot template using a global parameter o . As illustrated in Fig. 4, this adjustment sets the direction of the cloth pieces on the knot mesh (Fig. 3c). Lastly, the system divides each knot template

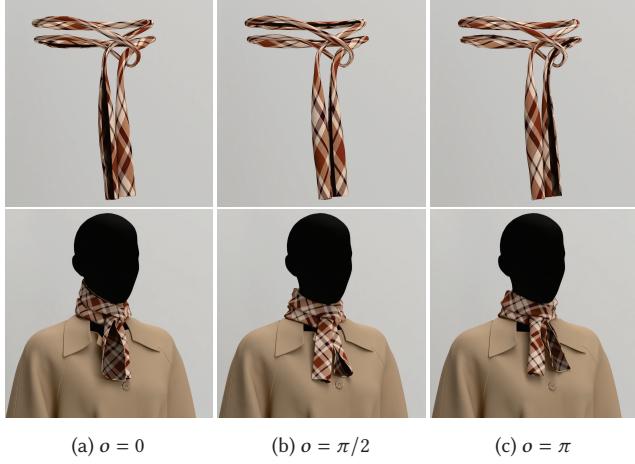


Fig. 4. Pretzel knots created by adjusting the orientation parameter o . Changes in this parameter alter the direction in which the cloth piece on the knot is oriented.



Fig. 5. Bow knots created by adjusting template parameters. These images illustrate the customized knot shapes after mapping and initialization.

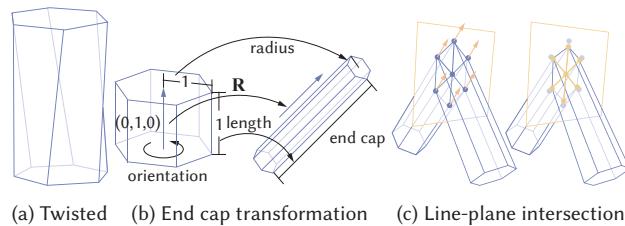


Fig. 6. The illustrated concepts involved in the tessellation of the knot template, i.e., the pipe.

into multiple components, allowing users to apply transformations, namely scaling, rotating, and translation, to the control points of each component. Fig. 5 illustrates the results of multiple bow knots created from the same cloth piece, achieved by adjusting the template parameters of different components.

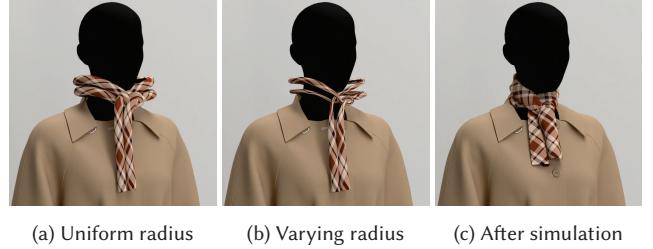


Fig. 7. The outcomes of using uniform versus adaptive radius reduction. When reducing the radius of the knot template to avoid collisions, adaptive radius reduction prevents the formation of small triangles, as seen in (a). This results in faster and more stable simulations, as demonstrated in (c).

4.2 Template Tessellation

Given the continuously defined knot template (i.e., the pipe) in Subsection 4.1, our next goal is to tessellate it into a structured triangle mesh, as illustrated in Fig. 3b. This tessellation results in a sequence of triangulated pipe-like segments. The key challenge lies in ensuring a continuous transition from one segment to the next while minimizing twist artifacts, as shown in Fig. 6a.

This challenge is closely related to the parallel transport problem in the theory of discrete elastic rods (DER) [Bergou et al. 2008]. In DER, parallel transport is typically used to maintain the orientation of rod segments as they bend and twist, with the orientation explicitly stored to address the degeneracy of the segments. This ensures that the local orientation of each rod remains consistent along its length. In contrast, our system applies parallel transport implicitly during the tessellation process. Here, the orientation between pipe segments is maintained through the mesh representation, eliminating the need for explicit storage of orientation at each node.

Specifically, the process begins at one end cap, progressively building the pipe structure. Initially, the first line segment is transformed into a discretized cylinder, ensuring that the cutting plane bisects the angle between adjacent segments.

End cap. Given the radius r , resolution res (e.g., $res = 6$ for hexagonal prism, as shown in Fig. 6), length l of the end cap, and orientation o , the positions are computed by iterating over $i \in [0, 2]$ and $j \in [0, res]$, adding vertices $\mathbf{p} = [r \cos \theta, (i - 0.5)l, -r \sin \theta]$, where $\theta = 2\pi j/res + o$, to the cylinder's vertex list \mathbf{v}_{cy} . The first end cap is created by generating and transforming a unit cylinder. For the first segment defined by nodes \mathbf{v}_0 and \mathbf{v}_1 , the segment direction is $\mathbf{d} = \mathbf{v}_1 - \mathbf{v}_0$, with length $l = \|\mathbf{d}\|$. A rotation matrix \mathbf{R} is then constructed to align the direction vector $\mathbf{d}_y = [0, 1, 0]^\top$ with \mathbf{d} [Möller and Hughes 1999]. Additionally, a translation vector $\mathbf{t} = \mathbf{d}/2$ and a diagonal scaling matrix $\mathbf{S} = \text{diag}(r, l, r)$, where r is the radius at \mathbf{v}_0 . The final positions of the end cap are computed as $\mathbf{v}_{ec} = \mathbf{RSv}_{cy} + \mathbf{t}$ (Fig. 6b) and added to the pipe's vertex list.

Parallel Transport. For each consecutive pair of line segments, \mathbf{l}_a and \mathbf{l}_b , parallelism must be checked. If they are parallel, the normal \mathbf{n} of the cutting plane is determined by either of the line segments. Otherwise, $\mathbf{n} = \text{normalize}(\mathbf{l}_a \times \mathbf{l}_b)$. Given any point \mathbf{P} on the plane, the equation of the cutting plane is then determined by: $n_x x + n_y y + n_z z - \mathbf{n}^\top \mathbf{P} = 0$. Parallel transport is computed through

the line-plane intersection (Fig. 6c). For a point Q , a vector v and a plane $ax + by + cz + d = 0$, the intersection point is computed as: $P_{int} = Q + t\hat{v}$, where $t = \frac{aP_x + bP_y + cP_z + d}{av_x + bv_y + cv_z}$. The intersection point is then adjusted to maintain consistency in the radius at each segment node, while remaining on the cutting plane (Fig. 6c). Given the radius r at the line segment node l , the intersection point is $P_{int} = l + r \cdot \text{normalize}(P_{int} - l)$.

4.3 Collision Handling during Customization

The customization of the knot shape can lead to self-collisions in the tessellated knot template. These collisions may occur between two consecutive segments due to an excessively bent Bézier curve or between disjoint segments resulting from overly modified template parameters. In this subsection, we outline our approach to detecting and resolving these collisions.

4.3.1 Continuous collision detection (CCD). The tessellation of the knot template enables convenient continuous collision detection on the resulting triangle mesh, as illustrated in Fig. 3b. Let \mathcal{K}_0 represent the initial state of the knot template with no self-collisions, and \mathcal{K}_1 represent the modified knot template, which may contain potential self-collisions. Assuming the transition from \mathcal{K}_0 to \mathcal{K}_1 is defined by linear interpolation, $(1-t)\mathcal{K}_0 + t\mathcal{K}_1$, the goal of CCD is to determine if any collision occurs within $t \in (0, 1)$ and calculate the corresponding t , referred to as the time of impact (TOI).

When a user performs a series of modifications to the knot template, a natural idea is to simply treat the last (penetration-free) template state as \mathcal{K}_0 and the current template state as \mathcal{K}_1 . However, if \mathcal{K}_0 is already near a collision state, the CCD problem can become ill-conditioned and prone to numerical errors. This can result in an infinitesimally small t , preventing the CCD process from successfully terminating.

To solve this issue, we propose to define \mathcal{K}_0 as the default state of the knot template instead, which is carefully created to be free of penetrations at the first place. Doing so ensures that the CCD process is well defined and the TOI t stays sufficiently away from zero, if collisions do occur. Specifically, we utilize additive continuous collision detection (ACCD) [Li et al. 2021], which incorporates a controlled thickness parameter ($2r_{min}$) designed for codimensional models. This approach enables precise collision handling for thin structures, making ACCD ideal for managing user-driven adjustments to the Bézier curve parameters.

4.3.2 Collision resolution by radius adjustment. Once collisions are found, the next question is: what should happen once a collision is detected by ACCD? We opt not to perform runtime simulation on the Bézier curve and template, as is done in yarn-level simulations, because doing so would distort the already customized knot shape and confuse the user experience. Similarly, we avoid simply restoring and freezing the template parameters, as this imposes overly stringent restrictions—especially for complex knots with limited room for collision-free adjustments.

Instead, we propose reducing the radius of the template until all penetrations are resolved. Since the template's radius is defined at each discretized segment node along the Bézier curve, we can adaptively adjust the radius only where necessary to eliminate collisions.

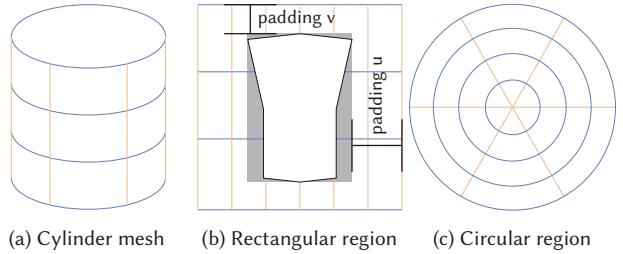


Fig. 8. Cylinder mesh and selected regions. Our system supports two mapping methods: rectangular mapping, which maps a rectangular region from (b) onto the surface mesh in (a), and circular mapping, which maps a circular region from (c) onto the same surface mesh.

In contrast, using a globally uniform radius adjustment would lead to unnecessary changes, creating small, unfriendly triangles and compromising the overall geometry of the knot (Fig. 7).

5 Knot Mapping and Initialization

Here we describe the construction of a knot mesh by mapping a selected region from the UV space of a cloth piece onto the tessellated pipe mesh surface. We also explain how to initialize its 3D position and orientation while ensuring it remains intersection-free with its surroundings.

5.1 Knot Mapping

Since we construct the tessellated pipe mesh homomorphically to a cylinder mesh, as shown in Fig. 8a, we can simplify the knot mapping problem to building a mapping from the selected region onto the cylinder surface. Our system supports two mapping methods:

- **Rectangular Mapping:** This method assumes the selected region is rectangular and wraps it around the cylinder surface, as illustrated in Fig. 8b. It is primarily used when the knot is made from stripe-like fabric pieces.
- **Circular Mapping:** This method assumes the selected region is circular, mapping each circular sector to a rectangle on the cylinder surface, as shown in Fig. 8c. It is suitable for knots formed by pulling two tips out of a fabric piece, such as in the knotted shirt examples in Fig. 1d and 1e.

Other mapping methods can be added into the system if necessary.

To introduce more flexibility in knot mapping, we do not restrict the selected region to the exact bounding box (or circle) of the fabric piece. Instead, we allow the selected region to include additional padding, as illustrated in Fig. 8b. This padding provides control over the mapped knot shape, enabling transitions from a cylinder-like structure to a nearly planar form. In the shirt example shown in Fig. 9, this feature is crucial for the system to create a neat tie.

5.2 Knot Initialization

After mapping the selected region(s) to form the knot mesh, we compute its 3D position and rotation. If the region spans an entire cloth piece, manual posing is required. For partial regions, a gap forms between the knot and remaining cloth. Let x^m and x^f denote boundary vertices and their 3D counterparts, leading to the shape

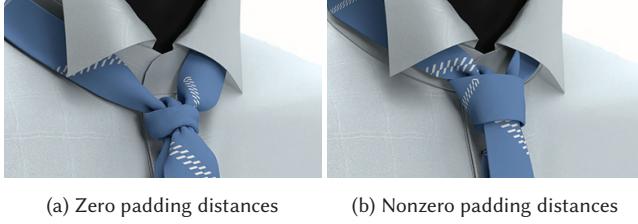


Fig. 9. The outcomes with zero and nonzero padding distances. Padding distances allow users to control the resulting knots and ties, ranging from a cylindrical shape, as shown in (a), to a nearly planar shape, as shown in (b).

matching problem: $\{\mathbf{R}, \mathbf{t}\} = \arg \min \sum \|\mathbf{R}\mathbf{x}_k^m + \mathbf{t} - \mathbf{x}_k^f\|^2$, solvable via standard formulas. Since \mathbf{R} and \mathbf{t} may cause mesh collisions, we dynamically adjust \mathbf{t} using real-time intersection checks. Inspired by [Lan et al. 2024], we weight collision gradients G_i by their lifespan K_i (e.g., $G = \sum_i 1.2^{K_i} G_i$), preserving constraint history for stable convergence. While not collision-free in all cases, this method enables efficient real-time feedback and a robust initial resolution.

6 Penetration-Free Simulation

With the knot mapped and initialized in 3D, the final step is simulating it with the surrounding meshes to achieve a quasistatic state for further demonstrations or simulations. The main challenge is efficiently resolving existing penetrations and preventing new ones. While our system addresses penetrations within the knot mesh itself (as detailed in Sections 4 and 5), issues may still arise from misaligned orientations between the knot and surrounding meshes, pre-existing mesh defects, or false positives from small triangles.

To address this, we need a primary simulator capable of efficiently handling collisions and resolving existing penetrations. Additionally, a fallback simulator is needed to strictly prevent new penetrations if the primary simulator fails to resolve them within a reasonable timeframe. Notably, a fallback simulator based on continuous collision detection relies on existing penetrations being resolved first—task our primary simulator handles swiftly and effectively.

6.1 A Primary Simulator

Our simulator unifies intersection resolution, collision handling, friction, and cloth dynamics in an optimization-based framework. Using a Gauss-Newton approximation for force Jacobians, we address collision gradients—similar to contact forces—without requiring a well-defined Hessian. Each timestep applies a single Newton method for quasistatic simulation, incorporating dynamics, contacts, collisions, and friction. With physics-based knot modeling, a friction coefficient of $\mu = 0.1$ ensures structural stability.

We use BVH for broad-phase culling, then detect edge-face intersection and vertex-face / edge-edge contacts. Our collision resolution minimizes intersections via contour minimization gradient G_{col} [Volino and Magnenat-Thalmann 2006], with stencil Hessian: $H_{col} = G_{col}G_{col}^\top$. Contact forces follow [Harmon et al. 2008]: $G_{con}^{vf} = (-\beta_0 N, -\beta_1 N, -\beta_2 N, N)$ and $G_{con}^{ee} = (-\beta_0 N, -\beta_1 N, \beta_2 N, N)$ where N is contact normal and β are barycentric coordinates. The GN Hessian is: $H_{con} = G_{con}G_{con}^\top$.

For linear subproblems in GN loops, we use Block-Jacobi PCG with 3×3 diagonal block preconditioners. The GN Hessian approximation avoids semi-positive definite projections, enabling real-time computation. Unlike post-processing methods, our unified framework combines collision handling and cloth dynamics, preventing inconsistencies from separate resolution steps and improving numerical stability. This approach supports larger timesteps, enhancing real-time performance.

6.2 An IPC-Based Fallback

While our primary simulator effectively eliminates and prevents penetrations, a fallback simulator is necessary to ensure strict collision safety if the primary simulator fails to prevent new penetrations. Although a standard IPC [Li et al. 2020] can be applied in this context, it proves inefficient for simulating knots and ties due to the frequent truncated linear searches. The initialized shape can generate millions of active contacting pairs with a small positive offset like 10^{-3} for constraint activation. Achieving collision-free equilibrium does not always require line searches for stability, particularly in cloth dynamics where overshooting is rare. Instead of precise line searches, we use intersection detection to avoid costly root-finding for cubic or higher-order polynomials. If a collision is detected, we backtrack by applying a scaling factor (e.g., 0.9) until no collision is found. Intersection detection is more efficient as it only requires lightweight geometric computation, returning a boolean result without iterative time-of-impact calculations.

7 Results and Discussions

We implement the mapping, initialization, and parameter adjustment components of our system on the CPU, while the subsequent quasistatic simulation component runs on the GPU. Our evaluation is conducted on an Intel® Core™ i9-13900K 3.00 GHz CPU and an NVIDIA® GeForce RTX™ 4090 GPU. The experiment includes both garment and non-garment examples, as illustrated in Fig. 1 and 10. These examples span a wide variety of knot and tie types, including the double knot, pretzel knot, bow knot, double bow knot, overhand knot, noose knot, and Windsor knot. Table 1 presents the statistics and timings for our examples, demonstrating that our system significantly reduces the time required to model these examples—some of which were previously impossible to create through manual manipulation.

7.1 Simulation Performance Comparisons

In this subsection, we evaluate the performance of our primary simulator, IPC-based fallback simulator, and a standard IPC simulator in simulating examples toward a quasistatic equilibrium. One challenge is that their convergence speeds cannot be directly compared as their potential objectives and resulting force gradients differ. Instead, we use the number of contacts as a metric, assuming this number stabilizes once the simulation reaches a quasistatic state.

Fig. 11 shows the number of contacts and collisions resolved using both methods. Both methods achieve reasonable time costs, with all collisions resolved through the primary while preserving the knot structures before equilibrium is reached. However, the



Fig. 10. The handbag example. Our system can also be applied to model knots and ties on accessories, such as the handbag shown in this example.

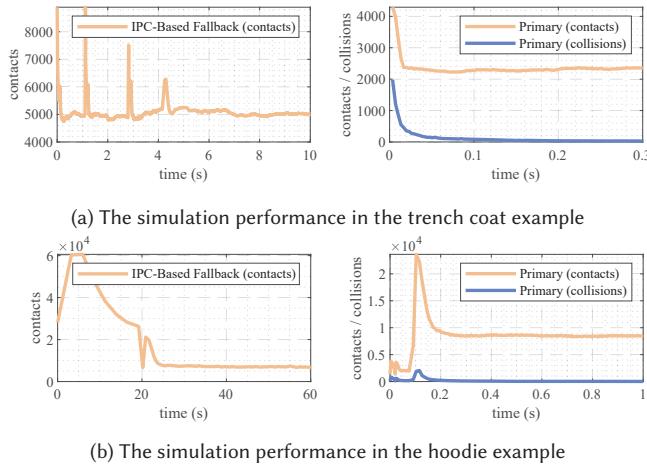


Fig. 11. The performance of our primary simulator and IPC-based fallback simulator. While the primary simulator does not strictly prevent new penetrations during simulations, it eliminates all penetrations by the end and is significantly more efficient. When treating the stabilized number of contacts as a convergence metric, the primary simulator is about 20 times faster than the fallback simulator.

primary simulator is exceptionally fast, making it ideal for real-time applications. While IPC-based fallback is also available as an option and it reaches equilibrium within a reasonable time frame, the efficiency makes it an uneconomic choice for these examples.

When using the GPU-based standard C-IPC [Li et al. 2021], the initial state in the first scenario of the benchmark (Fig. 5b & 11a) activates 6,006,744 contact pairs with a threshold of 10^{-3} , which requires over an hour to solve the first time step. This issue arises from the inherent dilemma in the original IPC formulation. A small positive offset must be applied to the barrier energy to make the method feasible. Smaller offsets can introduce numerical instability and worsen system conditions, while larger offsets result in many more active pairs, increasing the burden on optimization. Therefore, while IPC is accurate, it is impractical for production use. In contrast, our method offers users multiple solutions. The primary

Table 1. The statistics and timings of our examples. This table offers a rough estimation of the time required by our system compared to manual manipulation, as actual times may vary depending on the user’s experience.

Example	Knot type	#verts, #tris	Time (ours)	Time (manual)
Trench coat	Bow knot	2.9K, 5.3K	7 mins	3 hrs
	Pretzel knot	17K, 33K	1 min	37 mins
	Bow knot	12K, 22K	1 min	2 hrs
Hoodie	Double knot	49K, 97K	< 1 min	1 hr
	Bow knot	11K, 22K	< 1 min	2 hrs
Knotted blouse	Double knot	54K, 106K	12 mins	2 hrs
	Overhand knot	115K, 230K	17 mins	∞
Knotted shirt	Double knot	17K, 33K	17 mins	∞
	Overhand knot	115K, 230K	17 mins	∞
Dress shirt	Windsor knot	10K, 20K	< 1 min	4 hrs
	Noose knot	23K, 46K	10 mins	3 hrs
Handbag	Double bow knot	86K, 172K		

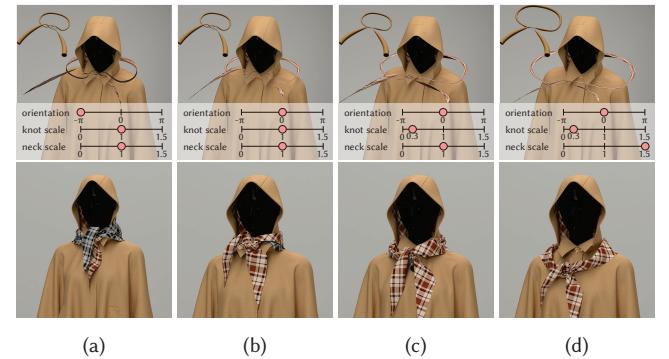


Fig. 12. Scarf customization on a hoodie. Top row: template and mapping results given by the parameter sliders; Bottom row: simulation.

simulator provides a high rate of success, although passing-through of the structure may still occur. Our IPC-based fallback guarantees safety and achieves speedups of over 100x, as demonstrated in the benchmark example.

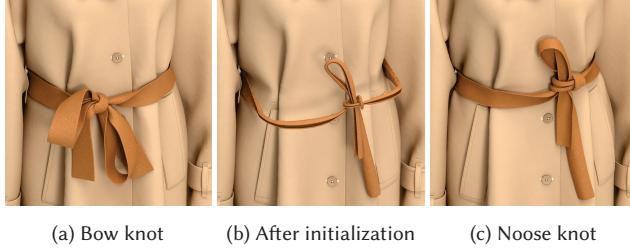
7.2 Variation

We present a scarf customization case study on a hoodie to demonstrate parametric control of double knot configurations, with key parameters including orientation, knot scale, and neck scale. Figs. 12a and 12b illustrate how a $-\pi$ rotation inverts the scarf’s inner layer in the simulated topology. Contracting the knot scaling coefficient (Figs. 12c and 12d) achieves tightening of the knot structure. Fig. 12d further demonstrates compatibility with expanded neck dimensions via increased neck scale parameter.

7.3 Comparison

Due to the understudied nature of knot and tie modeling, we benchmark our template-driven approach exclusively against the Interference-Aware Geometric Modeling (IAGM) [Harmon et al. 2011], which can be used to eliminate self-intersections in the modeling process of thickness-endowed spline curves.

IAGM contracts tessellated meshes until intersection-free, similar to our uniform-radius mode (Fig. 7a), but differs in two ways: (1) IAGM detects collisions on surface meshes (vertex/edge/face),



(a) Bow knot (b) After initialization (c) Noose knot

Fig. 13. Interactive design of knots on digital garments. Our system enables users to conveniently and interactively explore different knot categories without restarting the modeling process from scratch.

while ours uses medial-axis representation; (2) Our method avoids per-element checks, maintaining constant complexity, while IAGM slows with tessellation density. In tight configurations (Fig. 5b), ours runs at > 100 FPS vs. IAGM’s < 10 FPS — over $10\times$ faster.

Moreover, our adaptive radii strategy delivers two benefits against uniform contraction in IAGM: (i) It preserves geometric fidelity by preventing excessive shrinkage in low-curvature regions (e.g., far-knot segments), thereby eliminating artificial distortion from uniform contraction; and (ii) Adaptive radii modulation maintains mesh integrity by suppressing degenerate triangle generation in peripheral zones.

7.4 Interactive Knot Design

Our system allows users to interactively customize the shape and style of a knot and explore different knot categories without restarting the entire modeling process. For instance, a user can start with a bow knot in Fig. 13a and, if dissatisfied, quickly remap and reinitialize it to a different knot in Fig. 13b. After a brief simulation, the user can review the result in Fig. 13c and continue designing and editing until fully satisfied. This seamless and controllable interactive process is made possible only by our system.

7.5 Limitations

The system’s reliance on pre-defined templates for generating knots and ties presents challenges, as creating these templates can be complex and falls beyond the scope of this work. While the method is efficient and scalable for tested examples, its applicability to more intricate or unconventional knot structures remains unexplored, limiting its ability to handle entirely novel configurations without external modeling tools. While the system prevents penetrations within the knot, it cannot strictly prevent penetrations among the elements connecting the knot and surrounding meshes, due to misalignment.

Low-resolution tessellated meshes can introduce interpenetrations during geometric mapping. These unresolved collisions propagate through the simulation pipeline, leading to persistent pass-through artifacts even after the primary solver’s intersection-resolution stage. Fig. 14a illustrates a failure case in the scarf simulation. While the final output appears intersection-free, the simulated geometry fails to preserve the intended structure defined by the template



(a) Low resolution (b) High resolution

Fig. 14. This figure highlights a mesh pass-through artifact caused by low-resolution tessellation (a). Increasing the mesh resolution (b) effectively resolves this issue. (Target template: Fig. 12c.)

(Fig. 12c). By employing higher mesh resolution (Fig. 14b), our approach successfully resolves these artifacts while maintaining structural fidelity. These factors pose challenges for our simulators. In rare cases, existing penetrations may persist without assistance from an inactive fallback simulator. However, such instances are exceedingly rare and were not observed in our experiments.

8 Conclusions and Future Work

In conclusion, we proposed a novel method for efficiently modeling knots and ties using Bézier curve templates, ensuring intersection-free results and reducing the need for manual adjustments. With robust collision handling and real-time feedback, our approach streamlines the design process and demonstrates significant performance improvements over existing methods.

Future work could extend our method to handle additional fabric interactions, such as pleating, ruffling, or stitching, broadening its utility in digital fashion design. Integrating these elements into a unified framework would create a more comprehensive solution for garment simulation. Additionally, incorporating a flexible, algorithmic approach to generating knot templates directly within the system could reduce reliance on external tools and expand the range of knots that can be seamlessly modeled.

Acknowledgments

We thank the anonymous reviewers for their constructive feedback. This work was supported by Key R&D Program of Zhejiang (No. 2024C01069) and Shenzhen Science and Technology Program (No.KJZD20230923114114028).

References

- David Baraff and Andrew Witkin. 1998. Large Steps in Cloth Simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’98)*. Association for Computing Machinery, New York, NY, USA, 43–54.
- David Baraff, Andrew Witkin, and Michael Kass. 2003. Untangling Cloth. *ACM Trans. Graph. (SIGGRAPH)* 22, 3 (July 2003), 862–870.
- Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. 2008. Discrete elastic rods. In *ACM SIGGRAPH 2008 papers*. 1–12.
- Sofien Bouaziz, Sebastian Martin, Tianfian Liu, Ladislav Kavan, and Mark Pauly. 2014. Projective Dynamics: Fusing Constraint Projections for Fast Simulation. *ACM Trans. Graph. (SIGGRAPH)* 33, 4, Article 154 (July 2014), 11 pages.

- Robert Bridson, Ronald Fedkiw, and John Anderson. 2002. Robust Treatment of Collisions, Contact and Friction for Cloth Animation. *ACM Trans. Graph. (SIGGRAPH)* 21, 3 (July 2002), 594–603.
- Kwang-Jin Choi and Hyeong-Seok Ko. 2002. Stable but Responsive Cloth. *ACM Trans. Graph. (SIGGRAPH)* 21, 3 (July 2002), 604–611.
- Gabriel Cirio, Jorge Lopez-Moreno, David Miraut, and Miguel A. Otaduy. 2014. Yarn-Level Simulation of Woven Cloth. *ACM Trans. Graph.* 33, 6, Article 207 (nov 2014), 11 pages.
- CLO. 2025. How to Tie a Double Knot in CLO. <https://www.youtube.com/watch?v=Xqq9LRTjjt4>. [Online; accessed 20-Jan-2025].
- Dewen Guo, Minchen Li, Yin Yang, Sheng Li, and Guoping Wang. 2024. Barrier-Augmented Lagrangian for GPU-based Elastodynamic Contact. *ACM Trans. Graph.* 43, 6, Article 225 (Nov. 2024), 17 pages.
- David Harmon, Daniele Panozzo, Olga Sorkine, and Denis Zorin. 2011. Interference-aware geometric modeling. *ACM Transactions on Graphics (TOG)* 30, 6 (2011), 1–10.
- David Harmon, Etienne Vouga, Breannan Smith, Rasmus Tamstorf, and Eitan Grinspun. 2009. Asynchronous Contact Mechanics. *ACM Trans. Graph. (SIGGRAPH)* 28, 3, Article 87 (July 2009), 12 pages.
- David Harmon, Etienne Vouga, Rasmus Tamstorf, and Eitan Grinspun. 2008. Robust Treatment of Simultaneous Collisions. *ACM Trans. Graph. (SIGGRAPH)* 27, 3 (aug 2008), 1–4.
- Jonathan M. Kaldor, Doug L. James, and Steve Marschner. 2008. Simulating Knitted Cloth at the Yarn Level. *ACM Trans. Graph. (SIGGRAPH)* 27, 3 (Aug. 2008), 1–9.
- Jonathan M. Kaldor, Doug L. James, and Steve Marschner. 2010. Efficient Yarn-based Cloth with Adaptive Contact Linearization. *ACM Trans. Graph. (SIGGRAPH)* 29, 4, Article 105 (July 2010), 10 pages.
- Lei Lan, Minchen Li, Chenfanfu Jiang, Huamin Wang, and Yin Yang. 2023. Second-Order Stencil Descent for Interior-Point Hyperelasticity. *ACM Trans. Graph. (SIGGRAPH)* 42, 4, Article 108 (jul 2023), 16 pages.
- Lei Lan, Zixuan Lu, Jingyi Long, Chun Yuan, Xuan Li, Xiaowei He, Huamin Wang, Chenfanfu Jiang, and Yin Yang. 2024. Efficient GPU Cloth Simulation with Non-distance Barriers and Subspace Reuse. *ACM Trans. Graph. (SIGGRAPH Asia)* 43, 6, Article 226 (Nov. 2024), 16 pages.
- Lei Lan, Ran Luo, Marco Fratarcangeli, Weiwei Xu, Huamin Wang, Xiaohu Guo, Junfeng Yao, and Yin Yang. 2020. Medial Elastics: Efficient and Collision-Ready Deformation via Medial Axis Transform. *ACM Trans. Graph.* 39, 3, Article 20 (April 2020), 17 pages.
- Lei Lan, Guanqun Ma, Yin Yang, Changxi Zheng, Minchen Li, and Chenfanfu Jiang. 2022. Penetration-Free Projective Dynamics on the GPU. *ACM Trans. Graph. (SIGGRAPH)* 41, 4, Article 69 (jul 2022), 16 pages.
- Mackenzie Leake, Gilbert Bernstein, Abe Davis, and Maneesh Agrawala. 2021. A Mathematical Foundation for Foundation Paper Pieceable Quilts. *ACM Trans. Graph. (SIGGRAPH)* 40, 4, Article 65 (July 2021), 14 pages.
- Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M. Kaufman. 2020. Incremental Potential Contact: Intersection-and Inversion-Free, Large-Deformation Dynamics. *ACM Trans. Graph.* 39, 4, Article 49 (jul 2020), 20 pages.
- Minchen Li, Danny M. Kaufman, and Chenfanfu Jiang. 2021. Codimensional Incremental Potential Contact. *ACM Trans. Graph.* 40, 4, Article 170 (jul 2021), 24 pages.
- Minchen Li, Alla Sheffer, Eitan Grinspun, and Nicholas Vining. 2018. Foldsketch: Enriching Garments with Physically Reproducible Folds. *ACM Trans. Graph. (SIGGRAPH)* 37, 4, Article 133 (July 2018), 13 pages.
- Chen Liu, Weiwei Xu, Yin Yang, and Huamin Wang. 2024. Automatic Digital Garment Initialization from Sewing Patterns. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–12.
- Tiantian Liu, Adam W. Bargteil, James F. O'Brien, and Ladislav Kavan. 2013. Fast Simulation of Mass-spring Systems. *ACM Trans. Graph. (SIGGRAPH Asia)* 32, 6, Article 214 (Nov. 2013), 7 pages.
- Miles Macklin, Matthias Müller, and Nuttapong Chentanez. 2016. XPBD: Position-Based Simulation of Compliant Constrained Dynamics. In *Proceedings of the 9th International Conference on Motion in Games (Burlingame, California) (MIG '16)*. Association for Computing Machinery, New York, NY, USA, 49–54.
- Sebastian Martin, Bernhard Thomaszewski, Eitan Grinspun, and Markus Gross. 2011. Example-based Elastic Materials. *ACM Trans. Graph. (SIGGRAPH)* 30, 4, Article 72 (July 2011), 8 pages.
- Tomas Möller and John F. Hughes. 1999. Efficiently Building a Matrix to Rotate One Vector to Another. *J. Graph. Tools* 4, 4 (Dec. 1999), 1–4.
- Rahul Narain, Matthew Overby, and George E. Brown. 2016. ADMM \supseteq Projective Dynamics: Fast Simulation of General Constitutive Models. In *Proceedings of SCA* (Zurich, Switzerland). 21–28.
- Rahul Narain, Armin Samii, and James F. O'Brien. 2012. Adaptive Anisotropic Remeshing for Cloth Simulation. *ACM Trans. Graph. (SIGGRAPH Asia)* 31, 6, Article 152 (Nov. 2012), 152:1–152:10 pages.
- Jing Ren, Aviv Segall, and Olga Sorkine-Hornung. 2024. Digital Three-dimensional Smocking Design. *ACM Trans. Graph.* 43, 2, Article 14 (Jan. 2024), 17 pages. doi:10.1145/3631945
- Aviv Segall, Jing Ren, Amir Vaxman, and Olga Sorkine-Hornung. 2024. Fabric Tessellation: Realizing Freeform Surfaces by Smocking. *ACM Trans. Graph. (SIGGRAPH)* 43, 4, Article 89 (July 2024), 20 pages.
- Kai Suto, Yuta Noma, Kotaro Tanimichi, Koya Narumi, and Tomohiro Tachi. 2023. Crane: An Integrated Computational Design Platform for Functional, Foldable, and Fabricable Origami Products. *ACM Trans. Comput.-Hum. Interact.* 30, 4, Article 52 (Sept. 2023), 29 pages.
- Rasmus Tamstorf, Toby Jones, and Stephen F. McCormick. 2015. Smoothed Aggregation Multigrid for Cloth Simulation. *ACM Trans. Graph. (SIGGRAPH Asia)* 34, 6, Article 245 (Oct. 2015), 13 pages.
- Chen Tang, Sheng Li, and Guopin Wang. 2011. Fast Continuous Collision Detection Using Parallel Filter in Subspace. In *Symposium on Interactive 3D Graphics and Games*. 71–80.
- Pascal Volino and Nadia Magnenat-Thalmann. 2006. Resolving Surface Collisions Through Intersection Contour Minimization. *ACM Trans. Graph. (SIGGRAPH)* 25, 3 (July 2006), 1154–1159.
- Huamin Wang. 2015. A Chebyshev Semi-iterative Approach for Accelerating Projective and Position-Based Dynamics. *ACM Trans. Graph. (SIGGRAPH Asia)* 34, 6, Article 246 (Oct. 2015), 9 pages.
- Huamin Wang. 2021. GPU-Based Simulation of Cloth Wrinkles at Submillimeter Levels. *ACM Trans. Graph. (SIGGRAPH)* 40, 4, Article 169 (jul 2021), 14 pages.
- Huamin Wang and Yin Yang. 2016. Descent Methods for Elastic Body Simulation on the GPU. *ACM Trans. Graph. (SIGGRAPH Asia)* 35, 6, Article 212 (Nov. 2016), 10 pages.
- Zhendong Wang, Min Tang, Ruofeng Tong, and Dinesh Manocha. 2015. TightCCD: Efficient and Robust Continuous Collision Detection Using Tight Error Bounds. *Computer Graphics Forum (Pacific Graphics)* 34, 7 (2015), 289–298.
- Zhendong Wang, Longhua Wu, Marco Fratarcangeli, Min Tang, and Huamin Wang. 2018. Parallel Multigrid for Nonlinear Cloth Simulation. *Computer Graphics Forum (Pacific Graphics)* 37, 7 (2018), 131–141.
- Martin Wicke, Hermes Lanker, and Markus Gross. 2006. Untangling Cloth with Boundaries. In *Proc. of Vision, Modeling, and Visualization*. 349–356.
- Botao Wu, Zhendong Wang, and Huamin Wang. 2022. A GPU-Based Multilevel Additive Schwarz Preconditioner for Cloth and Deformable Body Simulation. *ACM Trans. Graph.* 41, 4, Article 63 (jul 2022), 14 pages.
- Zangyueyang Xian, Xin Tong, and Tiantian Liu. 2019. A Scalable Galerkin Multigrid Method for Real-Time Simulation of Deformable Objects. *ACM Trans. Graph. (SIGGRAPH Asia)* 38, 6, Article 162 (nov 2019), 13 pages.
- Ningfeng Zhou, Jing Ren, and Olga Sorkine-Hornung. 2024. Computational Smocking through Fabric-Thread Interaction. In *Computer Graphics Forum*. Wiley Online Library, e15030.