



Efficient frictional contacts for soft body dynamics via ADMM

Siyan Zhu¹ · Cheng Fang¹ · Peng Yu¹ · Xiao Zhai² · Aimin Hao¹ · Junjun Pan¹

Accepted: 29 April 2024

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2024

Abstract

This paper addresses the longstanding challenge of soft body dynamics with frictional contact through a novel combination of Projective Dynamics for elasticity simulation and Alternating Direction Method of Multipliers for frictional contact handling. The approach utilizes parallel local strain projection for deformable bodies and nonlinear Projected Gauss-Seidel for contact with Coulomb friction, consolidated by a pre-factorized global strain propagation step. Integration of contact stabilization, Matchstick anisotropic friction, and Rayleigh damping enhances reliability and usability. Effectiveness, accuracy, and computational efficiency are demonstrated in challenging cases, including multi-layer and persistent contacts. With a CPU-based parallel implementation, our method achieves visually plausible and stable simulation results at an interactive framerate in moderate-scale scenes, showcasing its applicability across various graphics applications.

Keywords Physics-based animation · Elastic Dynamics · Frictional Contact handling

1 Introduction

The significance of physically based soft body simulation in enhancing realism within applications such as video games, virtual surgery, and fashion design cannot be overstated. Despite its pivotal role, challenges remain in attaining the delicate balance between visual plausibility and computational efficiency, particularly in scenarios demanding accurate modeling of frictional contacts.

Numerous methods in computer graphics, such as Position Based Dynamics (PBD) [15, 28, 30], effectively handle soft body dynamics and contacts. However, challenges like slow convergence and jittering artifacts persist, especially in large-scale scenarios [10]. Alternative methods, e.g., the penalty force [26, 37] or the hard constraint [33, 42], struggle to meet both simulation quality and computational efficiency requirements, particularly in scenarios involving frictional contacts.

In the domain of elastic dynamics, Projective Dynamics (PD) [6] stands out for its efficiency and quality, leveraging pre-factorization for the global solve upfront. Meanwhile, recent advancements addressing frictional collisions [10, 11, 20, 39] use the Alternating Direction Method of Multipliers (ADMM) to decouple contact dynamics and elasticity, introducing a robust iteration framework.

The concept of integrating PD for simulating elasticity and ADMM for managing frictional contacts, with the objective of achieving both computational speed and visually plausible outcomes, is intriguing. Although it has been demonstrated that PD and ADMM can be combined for general deformable simulations [32], the extension to frictional contacts remains unexplored. Motivated by these considerations, this paper embarks on an exploration of this novel combination. Specifically, the elasticity is resolved using parallel local strain projection, while frictional contacts are addressed through the Projected Gauss-Seidel (PGS) method. A global strain propagation step, akin to the pre-factorized global step in PD, consolidates the contributions from these individual steps.

Furthermore, we enhance the utility of the existing simulation framework by incorporating anisotropic friction through the Matchstick Model. This augmentation is complemented by the inclusion of contact stabilization and Rayleigh damping, collectively contributing to an overall improvement in realism and reliability. Our method is inherently parallelizable and typically achieves visually plausible, stable, and

✉ Peng Yu
yupeng@buaa.edu.cn

✉ Xiao Zhai
zhaixiao43@gmail.com

✉ Junjun Pan
pan_junjun@buaa.edu.cn

¹ State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing, China

² Weta FX, Wellington, New Zealand

accurate results with only a few ADMM and PGS iterations at each timestep. This makes it well-suited for various applications, especially in interactive scenarios with moderate-scale scenes (around 20K DoFs).

Our contributions can be summarized as follows:

- Introducing a novel decoupled scheme for soft body dynamics with frictional contacts, combining the efficient elasticity solver from PD with the robust frictional contacts handling from ADMM.
- Integrating Matchstick anisotropic friction, contact stabilization, and Rayleigh elastic damping to enhance both reliability and realism.
- Demonstrating the effectiveness, accuracy, and computational efficiency of our solver, showcasing interactive framerates in moderate-scale scenes. Notably, our approach excels even in scenarios involving complex and persistent contacts.

2 Related works

Our work is built in the framework of a local–global solver for elastic dynamics and employs the operator splitting contact handling method within the ADMM scheme. In this section, we provide a concise overview of both these topics, focusing on methods designed for interactive applications.

2.1 Elastic deformation and frictional contact modeling

Drawing from works like [2, 10, 26], methods for soft body dynamics and contacts are broadly categorized into three types: *primal-based*, which treats both elasticity and contacts as energy optimization problems, *dual-based*, which treats elasticity and contacts as constraints, and *hybrid* approaches, which formulates a constrained optimization problem and treats frictional contacts as hard constraints.

2.1.1 Primal-based methods

Pioneered by [40], primal-based methods initially use explicit time integration to solve elasticity. Later developments, such as the backward Euler time integrator introduced by [4], demonstrated the versatility of this approach by utilizing penalty forces and velocity-level stiff penalties to handle collisions. For greater stability, [16] redefined backward Euler as an energy minimization problem, and [26] employed a penalty-based energy to model collision responses. For interactive simulations, numerous *first-order descent* methods have been introduced due to their parallelizability, exploiting the capabilities of modern GPUs [26, 43]. However, these methods face convergence challenges in extreme scenar-

ios [23], often resulting in penetration and slipping artifacts. In contrast, [38] opted for the projected Newton method, aligning it with a *second-order descent method*. Despite the improved stability, it also comes with computational overhead. Notably, [23] introduced Incremental Potential Contact (IPC), achieving a penetration-free projected Newton solver with accurate frictional contacts but demanding substantial computation.

2.1.2 Dual-based methods

Deformable dynamics can be effectively modeled by treating each finite element as a *soft constraint*. Initially explored for interactive applications by [36], subsequent developments, such as [41], introduced strain limiting and geometric stiffness. PBD, pioneered by [30] and refined by XPBD [28], employs a fast Gauss-Seidel method for iterative constraint solving. Collision methods like the unilateral constraints, the vertex-pushing technique [29] and the box Linear Complementarity Problem (LCP)-based frictional contact solver integration [15] demonstrate its versatility. [12, 31] introduced a frictional contact projection method rigid body PBD. However, these PBD-based methods exhibit issues with slow strain propagation [10], and their effectiveness heavily depends on iteration counts, making them less efficient when dealing with a relatively large number of positional degrees of freedom (DoFs). Additionally, artificial energy problems in contact scenarios [18] and unwanted impacts due to the convexification of friction cones [15] pose challenges for methods following the PBD paradigm.

2.1.3 Hybrid methods

Initially proposed by [3], hybrid methods leverage hard constraints to address contacts in deformable body dynamics, ensuring compliance with Signorini’s condition and Coulomb’s law. Later, [2] combined the hard constraints and the FEM elastic energy as a complementarity problem. The nonlinear complementarity problem (NCP) can be effectively tackled using the Newton’s method [5] or its conjugate-residual-based variant [27]. While accurate and featuring second-order convergence, these methods exhibit notable computation overhead and convergence challenges when the ratio of number of contact to DoFs is high [10]. Linearizing the NCP into the mixed linear complementarity problem (MLCP) [42], one can further employ the Shur complement and Delassus operator to convert MLCP to LCP. However, the LCP-based formulation with operator splitting, popular for simulating rigid bodies [2], proves inefficient for deformable bodies due to the dense nature of the Delassus operator and the challenges in its construction [10]. Iterative Constraint Anticipation (ICA) [33] solves the issue by using nested iterations. However, the demand on small time steps

and many collision detection passes makes it less suitable for interactive applications.

2.2 ADMM for deformable simulation with contact

ADMM originates from an Augmented Lagrangian Multiplier (ALM) and is designed to address optimization problems with constraints [7]. Their methods were reported to perform well in many fields, such as 3D reconstruction [1] and image restoration [8]. In recent years, ADMM has gained prominence in computer graphics [10, 11, 14, 32, 35, 37]. By judiciously decomposing the original problem into sub-problems and introducing auxiliary variables with linear constraints, ADMM yields a solver with a convergence rate of $O(1/k)$ through iterative local–global steps, where k represents the iteration count.

In their work, [32] leverage ADMM to expand the local–global elastic solver PD, accommodating generalized hyperelastic materials. This method comprises an independent local strain projection step and a global propagation step, facilitating matrix prefactorization and rapid forward solving. However, this approach may introduce jittering artifacts when dealing with contacts, a challenge addressed later by [25] in cloth simulation. Their solution involves reformulating PD at the velocity level and handling contacts with Coulomb friction.

In a recent advancement, Daviet [10] presented an ADMM-based technique that separates the PGS frictional contact solver from FEM elastic propagation. This method has been applied to real-time virtual hair editing, employing a Jacobi-style contact solver on GPU [11]. By directly addressing primal variables during contact resolution, this approach achieves a faster convergence rate and delivers visually compelling results within constrained computational time. Importantly, the contact solver in our method aligns with [10].

Contact resolution can be efficiently parallelized through suitable splitting methods. [39] disentangles the complementarity problem based on the associated friction law, separating it into an independent contact force projection step and a global constraint-based system-solving step. Subsequently, [9] extends this approach to a unified rigid body simulator. However, the drawback lies in the computational overhead incurred while solving a global system that undergoes changes at each ADMM iteration, making it less suitable for interactive applications. In an alternative approach, [20] proposes a double splitting scheme that breaks down FEM into numerous small subsystems, resulting in a system with many small, independent tasks. Nonetheless, this method requires a relatively small timestep and in-time factorization for each subsystem.

3 Background

This section provides a brief overview of constraint-based dynamics and ADMM to make the paper self-contained.

3.1 Constraint-based dynamics

Consider a soft body system with vertex positions \mathbf{x} , velocities \mathbf{v} and displacement χ . The equation of motion, accounting for both elastic forces and contact, is expressed as

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{K}\chi = \mathbf{f}_{\text{ext}} + \mathbf{f}_c, \quad (1)$$

where \mathbf{M} is the diagonal lump mass, \mathbf{K} is the elastic stiffness matrix, and \mathbf{f}_{ext} and \mathbf{f}_c are the external force and contact force, respectively. Employing finite difference time integration over timestep h , as discussed in Andrews et al. [2], the velocity can be updated through a system of the form $\mathbf{A}\mathbf{v} = \mathbf{b} + \mathbf{f}_c$, where $\mathbf{A} = \mathbf{M}/h + h\mathbf{K}$. The remaining terms are collectively represented by \mathbf{b} .

To express the relative velocity of contact points, we introduce the Jacobian matrix \mathbf{J} as $\mathbf{u} = \mathbf{J}\mathbf{v}$, where \mathbf{u} is the relative velocity, and the contact force $\boldsymbol{\gamma}$ is reported to nodes by $\mathbf{f}_c = \mathbf{J}^T \boldsymbol{\gamma}$. Combining these equations and using the Shur Complement technique yields a reduced equation as

$$\begin{aligned} \mathbf{u} &= \mathbf{J}\mathbf{A}^{-1}\mathbf{b} + \mathbf{J}\mathbf{A}^{-1}\mathbf{J}^T \boldsymbol{\gamma}, \\ \text{s.t. } \begin{cases} \mathbf{0} \leq \mathbf{u}_N \perp \boldsymbol{\gamma}_N \geq \mathbf{0} \\ \boldsymbol{\gamma} \in K_\mu, \quad \boldsymbol{\gamma}_T = -\mu \boldsymbol{\gamma}_N \frac{\mathbf{u}_T}{\|\mathbf{u}_T\|} \text{ if } \mathbf{u}_T \neq \mathbf{0}, \end{cases} \end{aligned} \quad (2)$$

where μ is the friction coefficient, K_μ denotes the Coulomb friction cone $\{\|\boldsymbol{\gamma}_T\| \leq \mu \boldsymbol{\gamma}_N, \boldsymbol{\gamma} \in \mathbb{R}^3\}$, and the N and T subscripts denotes the normal and tangential directions, respectively. Signorini's law is employed as a complementarity problem to define correct contact behavior, and Coulomb's law is integrated to delineate the friction response.

To simulate elastic dynamics with frictional contacts, we could solve Eq. (2) with unknown variables \mathbf{u} and $\boldsymbol{\gamma}$. Conventional methods to solve this complementarity problem, such as PGS, often require an expensive evaluation of \mathbf{A}^{-1} at each iteration. Our proposed method is motivated to address the computational challenges associated with both computing \mathbf{A}^{-1} and solving a system containing the dense matrix $\mathbf{J}\mathbf{A}^{-1}\mathbf{J}^T$ using the ADMM approach.

3.2 ADMM

Derived from the Augmented Lagrangian Method (ALM), ADMM is designed for solving convex optimization problems of the form

$$\min_{\mathbf{x}, \mathbf{z}} f(\mathbf{x}) + g(\mathbf{z}), \text{ s.t. } \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} + \mathbf{c} = \mathbf{0}, \quad (3)$$

Algorithm 1: Method Overview

Data: Mass, timestep h , ADMM parameters, physics parameters
Input: Simulation ready mesh
Output: Vertices displacement sequence from t_0 to t_n

```

/* Section. 4.1, 4.2 */
1 Build and prefactorize the sparse system matrix ;
2 for from  $t \leftarrow t_0$  to  $t_n$  do
3   foreach element do Elastic strain projection /* Section.
      5.1, 6.1 */
4   Frictional contact projection with optional stabilization;
      /* Section. 4.2 */
5   Global strain propagation with Rayleigh damping;
6 end

```

where $\mathbf{x} \in \mathbf{R}^{n_1}$ and $\mathbf{z} \in \mathbf{R}^{n_2}$ are primal optimization variables, $\mathbf{A} \in \mathbf{R}^{m \times n_1}$, $\mathbf{B} \in \mathbf{R}^{m \times n_2}$ and $\mathbf{c} \in \mathbf{R}^m$ constitute the constraint, and f and g are convex functions, not necessarily smooth [7]. Introducing dual variables $\boldsymbol{\lambda} \in \mathbf{R}^m$, the augmented Lagrangian is expressed as

$$L(\mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}) = f(\mathbf{x}) + g(\mathbf{z}) + \frac{\rho}{2} \|\mathbf{Ax} + \mathbf{Bz} + \mathbf{c} + \boldsymbol{\lambda}\|^2 + \mathbf{k}. \quad (4)$$

Here, we employ the scaled form of ADMM [7] as the constant term \mathbf{k} does not affect the optimum. The iteration scheme is as follows. For iteration $l + 1$,

$$\begin{aligned} \mathbf{x}^{l+1} &:= \arg \min_{\mathbf{x}} L(\mathbf{x}, \mathbf{z}^l, \boldsymbol{\lambda}^l), \\ \mathbf{z}^{l+1} &:= \arg \min_{\mathbf{z}} L(\mathbf{x}^{l+1}, \mathbf{z}, \boldsymbol{\lambda}^l), \\ \boldsymbol{\lambda}^{l+1} &:= \boldsymbol{\lambda}^l + \rho(\mathbf{Ax}^{l+1} + \mathbf{Bz}^{l+1} + \mathbf{c}). \end{aligned} \quad (5)$$

ADMM uses a Gauss-Seidel type update to solve Eq. 3 and is nearly as robust as ALM but with the advantages of variable splitting and problem decomposition. It is noteworthy that in practice, ADMM may still converge for some non-convex objective functions [7, 44].

4 Method overview

This section provides an overview of our simulation framework, outlined in Algorithm 1. Aligned with the ADMM iterative scheme, each optimization iteration is decomposed into a series of steps. This and the next section will elaborate on the solution process within each step.

4.1 Problem decomposition

The implicit Euler time integration of soft body dynamics can be formulated as the optimization problem

$$\mathbf{x}_{t+1} = \arg \min_{\mathbf{x}} \frac{1}{2h^2} \left\| \mathbf{M}^{\frac{1}{2}}(\mathbf{x} - \bar{\mathbf{x}}) \right\|^2 + U(\mathbf{x}) + F(\mathbf{x}, \mathbf{v}), \quad (6)$$

where $\bar{\mathbf{x}} = \mathbf{x}_t + h\mathbf{v}_t + h^2\mathbf{M}^{-1}\mathbf{f}_{\text{ext}}$ is the predicted position with \mathbf{f}_{ext} being the external force, $U(\mathbf{x})$ is the hyperelastic energy and $F(\mathbf{x}, \mathbf{v})$ is the potential energy caused by contacts.

To decouple the position DoFs from the local elastic strain, we introduce an auxiliary variable $\mathbf{z} \in \mathbf{R}^{n_e \times 3}$, where n_e denotes the number of elastic elements, and equality constraint $\mathbf{z}_i = \mathbf{D}_i\mathbf{x}$ for the i -th element. Here, \mathbf{D}_i represents the vertex weight matrix for each Finite Element, and these matrices can be concatenated into $\mathbf{D} = [\mathbf{D}_1^T, \mathbf{D}_2^T, \dots, \mathbf{D}_{n_e}^T]^T$. The local elastic energy of each element is defined as the quadratic form of a constraint C_i , scaled by stiffness k_i and volume v_i . The total elastic energy is then obtained by integrating this over the entire scene $U(\mathbf{x}) = \sum_{i=1}^{n_e} U_i(\mathbf{z}_i) = \sum_{i=1}^{n_e} \frac{1}{2}k_i v_i C_i^2(\mathbf{z}_i)$. In our solver, we employ four types of elastic constraints, namely: spring attachment, curvature bending constraint, and corotational-like elastic constraints on triangle and tetrahedron elements [6].

To decouple the nodal velocity \mathbf{v} and contact force $\boldsymbol{\gamma}$, related through the Delassus operator $\mathbf{JA}^{-1}\mathbf{J}^T$ in Eq. (2), we introduce an auxiliary virtual velocity \mathbf{p} and equality constraint $\mathbf{p} = \mathbf{v}$. Similar to Daviet [10], we define an indicator function $F_i(\mathbf{p})$ which is zero if the virtual velocity \mathbf{p}_i lies within the feasible manifold from Eq. (2) or $+\infty$ when \mathbf{p}_i violates the friction and penetration restrictions. Like the elastic energy, the contact potential is also integrated over the entire scene $F(\mathbf{p}) = \sum F_i(\mathbf{p})$. To unify the type of DoFs, we relate velocities \mathbf{v} and positions \mathbf{x} through $\mathbf{v} = (\mathbf{x} - \mathbf{x}_t)/h$, or $\mathbf{x} - \mathbf{x}_t - h\mathbf{p} = \mathbf{0}$ for the virtual velocities.

Plugging them into Eq. 6, we have

$$E(\mathbf{x}) = \frac{1}{2h^2} \left\| \mathbf{M}^{\frac{1}{2}}(\mathbf{x} - \bar{\mathbf{x}}) \right\|^2 + \sum_{i=1}^{n_e} U(\mathbf{z}_i) + F(\mathbf{p}), \quad (7)$$

$$s.t. \quad \mathbf{D}\mathbf{x} = \mathbf{z}, \quad \mathbf{x} - \mathbf{x}_t - h\mathbf{p} = \mathbf{0}.$$

There are two primal variables in Eq. (7), which may not appear perfectly aligned with the ADMM scheme in Eq. (3). Therefore, we concatenate \mathbf{z} and \mathbf{p} into a single variable $\mathbf{y} = [\mathbf{z}^T, \mathbf{p}^T]^T$. Additionally, we introduce diagonal matrices $\mathbf{W}_e = \text{diag}([k_{e0}, k_{e1}, \dots, k_{ei}, \dots]\rho)$ and $\mathbf{W}_c = k_c\rho\mathbf{I}_n$, where k_{ei} represents the stiffnesses of the i -th elastic constraints, k_c is a user-defined contact intensity, n is the number of contacts, and ρ denotes the ADMM penalty parameter. The constraint of form $\mathbf{Ax} + \mathbf{By} + \mathbf{c} = \mathbf{0}$ then becomes

$$\begin{bmatrix} \mathbf{W}_e\mathbf{D} \\ \frac{1}{h}\mathbf{W}_c \end{bmatrix} \mathbf{x} + \begin{bmatrix} -\mathbf{W}_e & \mathbf{0} \\ \mathbf{0} & -\mathbf{W}_c \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \mathbf{p} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ -\mathbf{W}_c \frac{1}{h}\mathbf{x}_t \end{bmatrix} = \mathbf{0}. \quad (8)$$

After introducing dual variables $\boldsymbol{\lambda}_c$ and $\boldsymbol{\lambda}_e$ associated with constraints $\mathbf{D}\mathbf{x} = \mathbf{z}$ and $\mathbf{x} - \mathbf{x}_t + h\mathbf{p} = \mathbf{0}$, the ADMM iterative scheme for solving soft body dynamics with frictional contacts is then formulated as the following 4 steps. At iteration $l + 1$,

1. The *global strain propagation* step $\mathbf{x}^{l+1} \leftarrow \arg \min_{\mathbf{x}} L_1$, where

$$L_1 = \frac{1}{2h^2} \left\| \mathbf{M}^{1/2}(\mathbf{x} - \bar{\mathbf{x}}) \right\|^2 + \frac{1}{2} \left\| \mathbf{W}_e^{1/2}(\mathbf{D}\mathbf{x} - \mathbf{z}^l + \boldsymbol{\lambda}_e^l) \right\|^2 + \frac{1}{2} \left\| \mathbf{W}_c^{1/2}(\mathbf{x} - h\mathbf{p}^l - \mathbf{x}_t + \boldsymbol{\lambda}_c^l) \right\|^2.$$

2. The *local elasticity projection* step $\mathbf{z}^{l+1} \leftarrow \arg \min_{\mathbf{z}} L_2$, where

$$L_2 = \sum_i U(\mathbf{z}_i) + \frac{1}{2} \left\| \mathbf{W}_e^{1/2}(\mathbf{D}\mathbf{x}^{l+1} - \mathbf{z} + \boldsymbol{\lambda}_e^l) \right\|^2,$$

which can be independently solved for each finite element.

3. The *frictional contact projection* step, aiming to project the virtual velocity \mathbf{p} into a feasible manifold $\mathbf{p}^{l+1} \leftarrow \arg \min_{\mathbf{p}} L_3$, where

$$L_3 = I_{K_\mu}(\mathbf{p}) + \frac{1}{2} \left\| \mathbf{W}_c^{1/2}(\mathbf{x}^{l+1} - h\mathbf{p}^{l+1} - \mathbf{x}_t + \boldsymbol{\lambda}_c^l) \right\|^2.$$

Here we replace indicator function $F(\mathbf{p})$ with $I_{K_\mu}(\mathbf{p})$ to clarify that the feasible set is a friction cone K_μ .

4. The *dual update* step.

$$\begin{aligned} \boldsymbol{\lambda}_c^{l+1} &= \boldsymbol{\lambda}_c^l + \mathbf{D}\mathbf{x}^{l+1} - \mathbf{z}^{l+1}, \\ \boldsymbol{\lambda}_e^{l+1} &= \boldsymbol{\lambda}_e^l + \mathbf{x}^{l+1} - h\mathbf{p}^{l+1} - \mathbf{x}_t. \end{aligned}$$

Step 2 can be executed locally for each element in parallel [32], and step 4 is trivial. We will delve into Step 1 and Step 3 in Sects. 4.2 and 5, respectively.

4.2 Global strain propagation

The global strain propagation aims for convergence by combining contributions from elasticity and contact. Considering $\arg \min_{\mathbf{x}} L_1$ as an unconstrained quadratic optimization problem, we can derive its first-order optimal condition $\nabla_{\mathbf{x}} L_1 = \mathbf{0}$ to obtain a globally minimal solution. This leads to a linear system with a constant left-hand side, which remains consistent despite changes in contact detection information and elastic stress,

$$\begin{aligned} (\mathbf{M} + h^2 \mathbf{D}^T \mathbf{W}_e \mathbf{D} + h^2 \mathbf{W}_c) \mathbf{x} = \\ \mathbf{M}\bar{\mathbf{x}} + h^2 \mathbf{D}^T \mathbf{W}_e (\mathbf{z}^l - \boldsymbol{\lambda}_e^l) + h^2 \mathbf{W}_c (\mathbf{p}^l - \boldsymbol{\lambda}_c^l). \end{aligned} \quad (9)$$

4.2.1 Rayleigh damping

Damping plays a crucial role in simulating realistic dissipative effects, but achieving it is not always straightforward. Applying artificial decay proportional to velocity can eliminate global rigid body motion, while methods like local

damping correction, as in [28], may result in unnaturally rigid behavior of soft bodies [22]. In our approach, we address damping in the global strain step, aiming to achieve a plausible result with minimum overhead.

FEM elastic system often uses the Rayleigh damping $\mathbf{f}_d = -k_d \mathbf{H}(\mathbf{x}_{t+1}) \mathbf{v}_{t+1}$, where k_d is damping stiffness, and $\mathbf{H} = \frac{\partial^2 E}{\partial \mathbf{x}^2}$ is the implicit elastic energy Hessian. However, this does not apply to our system, not only because it is challenging to compute but also because it disrupts the constancy of our left-hand side. Fortunately, our global energy L_1 has quadratic contribution from elasticity, allowing us to simplify the damping force to $\mathbf{f}_d = -k_d \mathbf{D}^T \mathbf{W}_e \mathbf{D}(\mathbf{x}_{t+1}) \mathbf{v}_{t+1}$. To integrate damping implicitly, our final global propagation system becomes

$$\begin{cases} (\mathbf{M} + h^2 \mathbf{D}^T \mathbf{W}_e \mathbf{D} + h^2 \mathbf{W}_c + \mathbf{A}_{\text{damp}}) \mathbf{x} = \\ \mathbf{M}\bar{\mathbf{x}} + h^2 \mathbf{D}^T \mathbf{W}_e (\mathbf{z}^l - \boldsymbol{\lambda}_e^l) + h^2 \mathbf{W}_c (\mathbf{p}^l - \boldsymbol{\lambda}_c^l) + \mathbf{b}_{\text{damp}}, \\ \mathbf{A}_{\text{damp}} = h k_d \mathbf{D}^T \mathbf{W}_e \mathbf{D}, \\ \mathbf{b}_{\text{damp}} = \mathbf{A}_{\text{damp}} \mathbf{x}_t. \end{cases} \quad (10)$$

It's worth noting that the system matrix remains constant across time steps. In practice, our damping method is similar to the Laplacian damping employed in the PD framework [22].

5 Decoupled frictional contact solver

In this section, we present the decoupled frictional contact projection and discuss how it bypasses the expensive Delassus operator $\mathbf{J}\mathbf{A}^{-1}\mathbf{J}^T$ to achieve efficiency.

To solve step 3, we assume that K_μ is a second-order cone [39], and find the first-order optimal conditions to L_3 as

$$\begin{cases} \mathbf{W}_c \mathbf{p} = \mathbf{J}^T \boldsymbol{\gamma} + \frac{1}{h} \mathbf{W}_c (\mathbf{x} - \mathbf{x}_t + \boldsymbol{\lambda}_c), \\ \mathbf{u} = \mathbf{J}\mathbf{p}, \quad (\mathbf{u}, \boldsymbol{\gamma}) \in K_\mu. \end{cases} \quad (11)$$

Note that $\mathbf{J}^T \boldsymbol{\gamma}$ does not directly result from taking the derivative of L_3 with respect to \mathbf{p} . Rather, it is determined by the requirements to satisfy the indicator function I_{K_μ} , which enforces that the velocity \mathbf{p} remains within the feasible manifold defined by K_μ , adhering to the friction laws and Signorini's condition.

This can be explained through the consideration of two cases. In instances where the contacts are separating, \mathbf{p} lies within the cone K_μ and $\mathbf{J}^T \boldsymbol{\gamma}$ is 0. Otherwise, when the contacts are sliding or sticking, $\mathbf{J}^T \boldsymbol{\gamma}$ represents the contact force directing \mathbf{p} toward its projection on the manifold. It is important to note that this holds true under the assumption that K_μ

is a convex cone, a condition we establish earlier by assuming K_μ to be a second-order cone. A detailed derivation can be found in the appendix of [10].

In situations where K_μ is non-convex, like Coulomb friction law, there is no guarantee that Eq. (11) remains valid. Nonetheless, the two types of friction laws exhibit the same behavior when the contact pair is either separating or sticking. The primary divergence emerges in the sliding case: Coulomb friction produces a relative velocity \mathbf{u} precisely aligned with the tangent plane, while convex friction laws generate a separating normal relative velocity $u_N > 0$ [10, 39]. Although theoretically this inconsistency may introduce convergence risks, this substitution works well in a majority of practical cases. Moreover, we introduce a contact stabilization technique later in the paper to further mitigate the potential instability.

5.1 Local contact projection

We consider two types of contacts in the simulation, the edge-edge contacts and the point-triangle contacts. For either type, the gap function can be linearly represented from the positional DoFs using $\phi_c = \sum_{i \in c} \mathbf{J}_{c,i} \mathbf{x}_i$.

By taking Shur complement and denoting the Delassus operator as $\mathbf{Q} = \mathbf{J}\mathbf{W}_c^{-1}\mathbf{J}^T$, the complementarity problem in Eq. (11) becomes

$$\begin{cases} \mathbf{u} = \mathbf{Q}\boldsymbol{\gamma}^l + \frac{1}{h}\mathbf{J}(\mathbf{x}^{l+1} - \mathbf{x}_t + \boldsymbol{\lambda}_c^l), \\ (\mathbf{u}, \boldsymbol{\gamma}) \in K_\mu. \end{cases} \quad (12)$$

In contrast to the dense matrix $\mathbf{J}\mathbf{A}^{-1}\mathbf{J}^T$, the Delassus operator \mathbf{Q} exhibits significant sparsity, which makes it suitable for iterative methods. We therefore solve Eq. (12) using PGS. Denoting the constant term $\frac{1}{h}\mathbf{J}(\mathbf{x}^{l+1} - \mathbf{x}_t + \boldsymbol{\lambda}_c^l)$ as \mathbf{b}^{l+1} , at iteration $k + 1$, each contact g is resolved sequentially by first initializing

$$\mathbf{u}_g^{k+1} = \begin{cases} q_{g,d}\boldsymbol{\gamma}_g^{k+1} + \sum_o q_{g,o}\boldsymbol{\gamma}_o^l + \mathbf{b}^{l+1} & \text{if } k > 0, \\ \mathbf{J}\mathbf{p} & \text{if } k = 0, \end{cases} \quad (13)$$

where the superscript $^{k+1}$ denotes the implicit variables undergoing update, q_g represents the block corresponding to contact g within \mathbf{Q} , and the subscripts d and o distinguish diagonal and off-diagonal elements, respectively. Subsequently, we employ the *enumerative solver* by Daviet [10] to project $(\mathbf{u}_g^{k+1}, \boldsymbol{\gamma}_g^{k+1})$ into K_μ . Notably, the scale matrix $q_{g,d}$ is isotropic, rendering the projection of $(\mathbf{u}_g^{k+1}, q_{g,d}\boldsymbol{\gamma}_g^{k+1})$ into K_μ equivalent to our problem. To elaborate, we classify the contact into three cases based on

$$\mathbf{u}_g^* = \mathbf{u}_g^{k+1} - q_{g,d}\boldsymbol{\gamma}_g^{k+1} = \mathbf{J}\mathbf{p}^{k+1} - q_{g,d}\boldsymbol{\gamma}_g^{k+1}, \quad (14)$$

1. *Separating*. If $\mathbf{u}_g^* \cdot \mathbf{n} > 0$, where \mathbf{n} is the contact normal, the collision is parting and we set $\mathbf{u}_g^{k+1} = \mathbf{u}_g^*$.
2. *Sticking*. In the sticking case, \mathbf{u}_g^{k+1} is set to $\mathbf{0}$. This scenario can be detected when $-\mathbf{u}_g^*$ lies in K_μ .
3. *Sliding*. If neither of the above cases holds, the contact is in a sliding state. Here, \mathbf{u}_g^{k+1} is anticipated to be zero along the normal, and the force $q_{g,d}\boldsymbol{\gamma}_g^{k+1}$ truncated at cone K_μ in the tangential plane. Therefore, we have

$$\mathbf{u}_g^{k+1} = \mathbf{u}_{g,T}^* + \mu \left\| \mathbf{u}_{g,N}^* \right\| \frac{\mathbf{u}_{g,T}^*}{\left\| \mathbf{u}_{g,T}^* \right\|},$$

where $\mathbf{u}_{g,T}^*$ and $\mathbf{u}_{g,N}^*$ are the tangential and normal components of \mathbf{u}_g^* , respectively.

Once \mathbf{u}_g^{k+1} is computed, we can synchronize the update of $\boldsymbol{\gamma}_g^{k+1}$ using either Eqs. (13) or (14) to maintain their compatibility. Subsequently, we immediately adjust the velocity \mathbf{p}^{k+1} of the corresponding four nodes by

$$\mathbf{p}_i^{k+1} = \mathbf{p}_i + \frac{J_{g,i}}{w_i}(\boldsymbol{\gamma}_g^{k+1} - \boldsymbol{\gamma}_g).$$

6 Discussion

The details of our contact solver are summarized in Alg. 2. To maintain alignment with [32], we have reordered step 1 and step 3. In practice, a small number of PGS iterations (1–6 in our solver) within each ADMM iteration are sufficient to achieve a near-optimal solution. This usually yields a penetration-free outcome with visually plausible frictional contact response, even in challenging scenarios such as multi-layer cloth. The fast convergence is attributed to the Gauss-Seidel style iteration scheme, which resembles the fast constraint updates in PBD. Each contact projection step directly updates the virtual velocity \mathbf{p} for the corresponding four nodes, in addition to the contact force $\boldsymbol{\lambda}$. To keep the contact feasible manifold up to date, more than one collision detection is triggered to update the active contacts set within a single time step. This approach shares similarities with ICA [33] and IPC [23], which repeatedly identify newly updated contact manifolds within a time step. The distinction lies in our formulation of contact dynamics, which is fundamentally dual-based and capable of solving primal elastodynamics in a local–global manner.

6.1 Contact stabilization

With persistent and multi-layer contacts, the PGS method usually has two common issues: penetration and jittering.

Algorithm 2: ADMM solver for elastic and contact dynamics from time t to $t + 1$

```

Data: mass matrix  $\mathbf{M}$ ; weight matrix  $\mathbf{W}_e, \mathbf{W}_c$ ; selection matrix  $\mathbf{D}$ , timestep  $h$ ; #element  $n_e$ ; #vertices  $n$ ;
Input: position  $\mathbf{x}_t$ ; velocity  $\mathbf{v}_t$ ; dual variable  $\lambda_e, \lambda_c$ 
Output: position  $\mathbf{x}_{t+1}$ ; velocity  $\mathbf{v}_{t+1}$ ; dual variable  $\lambda_e, \lambda_c$ 
1  $\bar{\mathbf{x}} \leftarrow \mathbf{x}_t + h\mathbf{v}_t$ ;  $\mathbf{b} \in \mathbb{R}^{n \times 3}$ ;  $\mathbf{z} \in \mathbb{R}^{n_e \times 3}$ ;  $\mathbf{x}_{curr} \leftarrow \bar{\mathbf{x}}$ ;
2 for  $l$  from 1 to  $\max ADMM Iterations$  do
3   if collision detection interval reached then
4     Update collision detection ;
5      $k \leftarrow \#contacts$ ,  $\mathbf{Q} \leftarrow \mathbb{R}^{k \times 4}$ ;
6     Precompute  $\mathbf{Q}_c = \mathbf{J}_c^T \mathbf{W}_c \mathbf{J}_c$ ; /* Eq. 12 */
7   end
8   /* Step 2: Elastic local step */
9    $\mathbf{z} \leftarrow \mathbf{D} \mathbf{x}_{curr} + \lambda_e$ ;
10  Solve elastic projection in parallel, update  $\mathbf{z}$ ;
11  /* Step 3: Contact projection step */
12   $\mathbf{p} \leftarrow (\mathbf{x}_{curr} - \mathbf{x}_t + \lambda_c)/h$ ;
13  for  $k$  from 1 to  $\max PGS Iterations$  do
14    | Solve Eq. 13 for every contact, update  $\mathbf{p}$ ;
15  end
16  /* Step 1: Global Propagation, Eq. 9 */
17   $\mathbf{b} \leftarrow \mathbf{M}\bar{\mathbf{x}} + h^2 \mathbf{D}^T \mathbf{W}_e (\mathbf{z} - \lambda_e)$ ;
18   $\mathbf{b} \leftarrow \mathbf{b} + h^2 \mathbf{W}_c (h\mathbf{p} + \mathbf{x}_t - \lambda_c)$ ;
19  Solve the linear system using precomputed LLT decomposition, update  $\mathbf{x}_{curr}$ ;
20  /* Step 4: Dual update */
21   $\lambda_e \leftarrow \lambda_e + \mathbf{D}\mathbf{x}_{curr}$ ;
22   $\lambda_c \leftarrow \lambda_c + \mathbf{x}_{curr} - \mathbf{x}_t - h\mathbf{p}$ ;
23 end

```

These challenges stem from the difficulty to rapidly converge when facing simultaneous but contradicting contacts. In some circumstances, allowing a provisional penetration can be advantageous to resolve contacts in conflict.

To stabilize the contacts, we introduce compliance for these hard constraints, drawing inspiration from the compliance in XPBD. We replace the fraction α of the Lagrange-multiplier-based collision force with a secondary penalty force. Assuming a Hookean spring at the contact point, the penalty force is $\mathbf{f}_{spring} = -k_s \phi_N \mathbf{n}$, where k_s is the spring constant, and ϕ_N is the gap function along the contact normal. Blending the hard contact force $\boldsymbol{\gamma}$ and the spring force, Eq. (11) becomes

$$\begin{bmatrix} \mathbf{W}_c & -\mathbf{J}^T \\ \mathbf{J} & \alpha \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{p}^{l+1} \\ \boldsymbol{\gamma} \end{bmatrix} = \begin{bmatrix} \mathbf{W}_c (\mathbf{v}^{l+1} + \frac{1}{h} \lambda_c^l) \\ \mathbf{u} - k_s h \alpha \Phi_N \end{bmatrix}, \tag{15}$$

$(\mathbf{u}, \boldsymbol{\gamma}) \in K_\mu$,

where Φ_N is the concatenated vector of ϕ_N . This results in a Schur complement equation slightly different than Eq. (12)

$$\begin{aligned} \mathbf{u} &= \mathbf{Q}' \boldsymbol{\gamma} + \mathbf{J} (\mathbf{v}^{l+1} + \frac{1}{h} \lambda_c^l) + k_s h \alpha \Phi_N, \\ \mathbf{Q}' &= \mathbf{Q} + \alpha \mathbf{I}. \end{aligned} \tag{16}$$

However, the local contact projection can be carried out in the same manner as before.

The compliance factor α plays a crucial role in balancing the hard and the soft contacts. Setting $\alpha = 0$ or 1 results in purely hard or soft contact. In practice, employing a very small α and a significantly high k_s is enough to ensure that the combined contact forces do not incur penetration. However, the major drawback of this stabilization scheme, apart from the additional user parameters, lies in the inability to reliably model friction on the supplemental penalty forces.

6.2 Anisotropic friction

Anisotropic friction response is often desired when simulating cloth contacts, especially for fabrics with unique weaving patterns. To achieve this, we incorporate the Matchstick Model [13] into our contact solver.

To capture the anisotropic texture of the surface, we introduce a *structure direction field*, which represents a vector field tangent to the surface. The Matchstick Model employs an *elliptical cone*, where the angles of the axes are determined by the two field vectors on the contacting surfaces, denoted as \mathbf{s}_1 and \mathbf{s}_2 . The model is defined as follows

$$\begin{aligned} \mathbf{t} &= \frac{\mathbf{s}_1 + \mathbf{s}_2}{\|\mathbf{s}_1 + \mathbf{s}_2\|}, \quad \mathbf{b} = \mathbf{n} \times \mathbf{t}, \\ d &= 1 - \frac{2}{\pi} \arccos(|\mathbf{s}_1 \cdot \mathbf{s}_2|), \end{aligned} \tag{17}$$

$$\begin{aligned} \mu_{\mathbf{t}} &= d \mu_{\text{aniso1}} + (1 - d) \mu_{\text{iso}}, \\ \mu_{\mathbf{b}} &= d \mu_{\text{aniso2}} + (1 - d) \mu_{\text{iso}}, \end{aligned}$$

where \mathbf{t} and \mathbf{b} denote the tangent and binormal orientations, and μ_{iso} , μ_{aniso1} , and μ_{aniso2} represent user-defined isotropic and anisotropic friction coefficients. Notably, the anisotropy reaches its maximum when \mathbf{s}_1 and \mathbf{s}_2 are parallel, and reduces to isotropic friction when they are perpendicular.

We replace the isotropic friction cone K_μ in Eq. (12) with the anisotropic counterpart $K_{\mu_{\mathbf{t}}, \mu_{\mathbf{b}}}$. To project $(\mathbf{u}, \boldsymbol{\gamma})$ onto the elliptical cone, we truncate \mathbf{u}_t within the tangent space by calculating the intersection points with ellipse P using the parametric equation $P(\theta) = \mu_{\mathbf{t}} \sin(\theta) + \mu_{\mathbf{b}} \cos(\theta)$. This assessment determines if truncation is necessary and identifies the precise truncated position. This minor refinement allows us to achieve visually plausible anisotropic friction behavior.

6.3 Parameter heuristics

One potential drawback of our ADMM scheme is its sensitivity on \mathbf{W}_c and \mathbf{W}_e . Choosing them arbitrarily will negatively affect convergence. In practice, we use $\mathbf{W}_e = 0.5\mathbf{K}$ according to the elastic stiffness \mathbf{K} to achieve best convergence, and heuristically set $\mathbf{W}_{c,v} = g(\eta_v)$, where

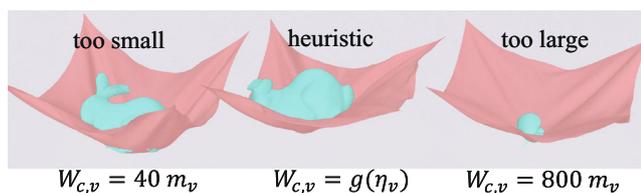


Fig. 1 Influence of different choices of W_c

$g(\eta_v) = \frac{1}{h^2} \text{clamp}(25m_v, 0.001\eta_v, \eta_v)\mathbf{I}$ for the v -th vertex, as suggested by [10], where m_v is its mass, and η_v is the smallest eigenvalue of its elastic matrix's 3×3 block on the diagonal. Due to the isotropy of our global propagation, the three eigenvalues are equal and therefore can be calculated as $\eta_v = m_v + (h^2 \mathbf{D}_v^T \mathbf{W}_{e,v} \mathbf{D}_v)$, where \mathbf{D}_v is the v -th column of \mathbf{D} . Figure 1 illustrates the result of different W_c .

7 Results

We evaluate our method in various scenarios detailed in Table 1. Our solver is implemented in C++ and executed on a desktop PC with an Intel i9-13900k CPU and 32GB RAM. We use Eigen3 for all linear algebra operations and oneTBB to parallelize computations. Additionally, we also integrate a naive BVH for broad-phase contact culling and discrete proximity query for narrow-phase collision detection.

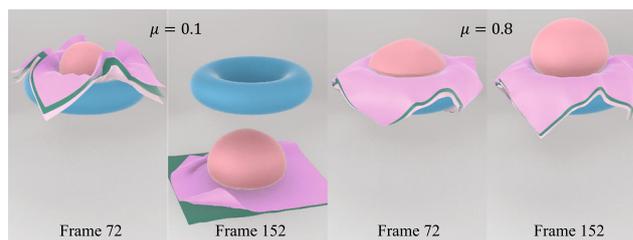
7.1 Effectiveness validation

We present a comprehensive evaluation of our solver, demonstrating its effectiveness across four key dimensions: contacts management in intricate setups, isotropic friction handling, anisotropic friction handling, and convergence properties.

7.1.1 Intricate contact setups

In Fig. 2a, we illustrate the *ball on torus* scenario, where a dense elastic ball falls onto a stationary torus, separated by three layers of cloth. When employing a smaller friction coefficient μ , both the ball and the cloth reach ground. However, a higher μ increases friction, impeding the ball from passing through the opening in the torus. In Fig. 2b, we illustrate the *ball on layers* scene, which features a stack of 10-layer cloths, with one of them pinned on four corners. A heavy ball falls onto the cloths, inducing many frictional contacts. Remarkably, the simulation remains stable and visually plausible, even as the number of contacts exceeds 2.3 times the number for DoFs.

We further assess our solver in a more practical and challenging *dancing* scene, as shown in the top of Fig. 3. Our solver demonstrates the capability to generate high-quality cloth motion at an interactive framerate. It's noteworthy that



(a) Heavy ball falls onto torus separated by 3 layers of cloth.



(b) Elastic ball falls onto 10 layers of cloth.

Fig. 2 Multi-layer frictional contact tests

the characters' cloths and trousers adhere to the moving bodies exclusively through friction, without any explicit pinning. Our solver also scales well, as illustrated at the bottom of Fig. 3, where over 800 elastic donuts with 1.3K vertices each fall on a plane. At timestep $h = 0.025s$ and 40 ADMM iterations per timestep, this stress test is able to produce visually plausible result for over 1 million vertices (over 3 million DoFs), averaging at 33.2s per frame, with a slight performance tweak by carrying out the global step individually for each donut.

When comparing with the state-of-the-art open-sourced IPC [23] and C-IPC [24] methods, our method is able to reproduce the visual result under the same time step and material parameters. As illustrated in Fig. 4, the *twisting rods* scene, adapted from the IPC paper, simulates four rods with intricate intertwining under large stress; the *drape on sphere* scene simulates a drape falling onto a rotating sphere. Our method has an impressive $9.7\times$ (*twisting rods*) and $63.2\times$ (*drape on sphere*) speed-up, respectively. We also noticed that the computing time of our solver is less affected by the collision density compared to the IPC methods. However, in the presence of extreme stress, our solver cannot ensure successful contact, unlike the IPC family methods, particularly when operating within highly constrained time limits.

Table 1 Statistics for the experiments shown in the paper

Scene	ADMM				Runtime					
	#DoFs	#ADMM ¹	W_c	#PGS ²	CDI ³	Max#c ⁴	Avg t/f (ms) ⁵	CD% ⁶	CS/EP% ⁷	GP% ⁸
Ball on layers ($\mu = 0.8$)	11796	25	80.0	2	10	9553	33.01	39.4	26.2	22.7
Multilayered (w/contact stab)	23751	20	400.0	3	5	46807	76.08	37.7	35.1	17.3
Analytical cloth	414	100	80.0	5	10	166	11.32	29.9	46.0	16.4
Masonry arch ($\mu = 0.8$)	600	40	30.0	6	4	170	17.03	68.1	20.0	7.6
Dancing (T-shirt and pants)	37665	40	25.0	3	4	23341	208.32	30.9	24.7	30.4
Twisting rods	6996	20	1000.0	3	2	1978	26.90	50.3	24.7	14.7
Ball on torus ($\mu = 0.1$)	7857	30	425.0	3	5	11056	45.54	43.5	28.9	18.9
House of cards (fine, $\mu = 0.8$)	1719	100	50	5	10	313	23.77	30.8	39.7	20.0
Large scale scene	3.12M	40	800	5	5	995.8K	33191.47	27.7	48.0	14.6
Drape on sphere	51309	50	400	5	5	34797	557.70	27.5	27.9	32.1

^{1,2,3}#ADMM: number of ADMM iterations. #PGS: number of PGS iterations. CDI: collision detection interval

⁴Max#c: max number of contact pairs during simulation

⁵Avg t/f (ms): average time per frame in milliseconds

⁶⁻⁸CD%: percentage of time for collision detection. CS/EP%: percentage of time for PGS contact solving and elastic projection, since the two are executed in parallel for performance consideration, they are clustered into one category. GP%: percentage of time for global strain propagation



Fig. 3 Top: dancing characters from SMPL. The character animation and clothing meshes are from HOOD [17]. Bottom: large-scale scene with over 3 million DoFs

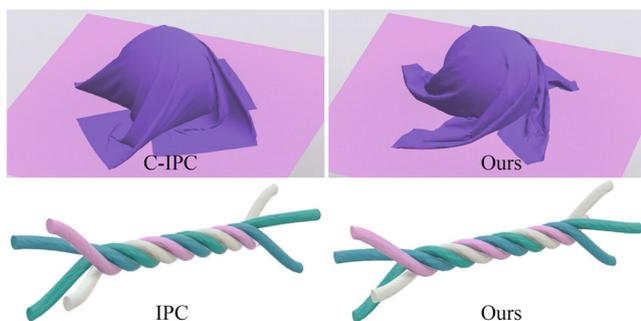


Fig. 4 Top: comparison with C-IPC in *drape on sphere*. Bottom: comparison with IPC in *twisting rods*

7.1.2 Isotropic friction

While the *dancing* scene has a visual validation of the effectiveness of our frictional contact solver, Fig. 5 provides a further examination of its accuracy using the *masonry arch* structure. This scene serves as a challenging case and test benchmark for friction simulation, adapted from Li et al. [23]. The friction coefficient μ plays a crucial role: a high value (0.9) leads to a stable equilibrium, while lower values (0.4, 0.05) may result in the disruption of the equilibrium due to sliding. Note that our solver does not simulate “rigid bodies,” but very stiff elastic pieces as approximation in this scene.

Another demanding friction benchmark is the *house of cards*, inspired by ARGUS [21]. As illustrated in Fig. 6, our solver effectively handles friction in this scene, maintaining

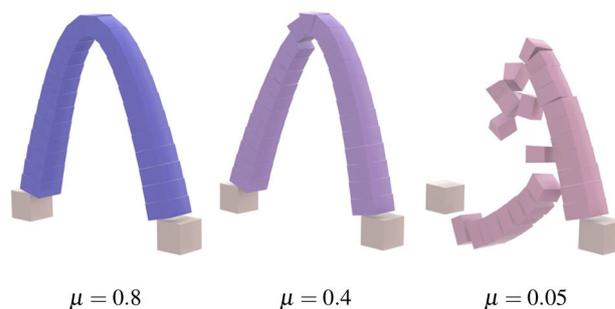


Fig. 5 *Masonry arch* with different friction coefficient

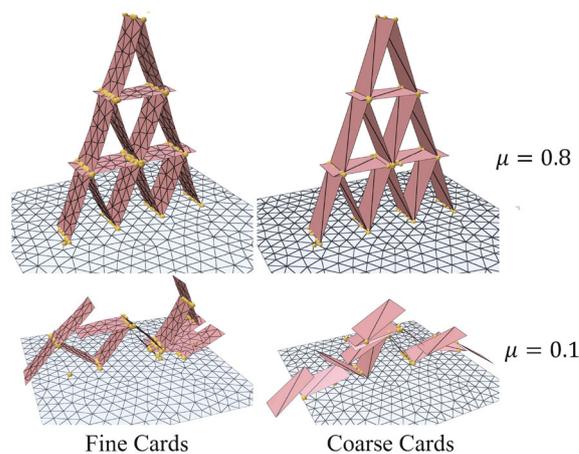


Fig. 6 *House of cards* with different friction coefficient and mesh resolution. The yellow dots indicate contact points

card stability under high friction coefficients and different mesh resolutions.

We further validate our solver analytically on the interactions of a piece of cloth or a box falling onto a slope with varying angles and friction coefficients. The obtained results align with the analytical expectations, as illustrated in Fig. 7. This validation confirms the accuracy of our solver and is consistent with state-of-the-art works [21, 25].

7.1.3 Anisotropic friction

Figure 8 shows the scene *cloth rolling* to validate the anisotropic friction response. We set $\mu_{iso} = 0.3$, $\mu_{aniso1} = 0.05$ and $\mu_{aniso2} = 1.3$. The green plane features a friction structure direction normal to the beige cloth, resulting in isotropic resistance. Conversely, the violet plane, with the same friction structure but placed perpendicularly, exhibits higher friction, impeding the movement of the cloth and eventually causing it to roll.

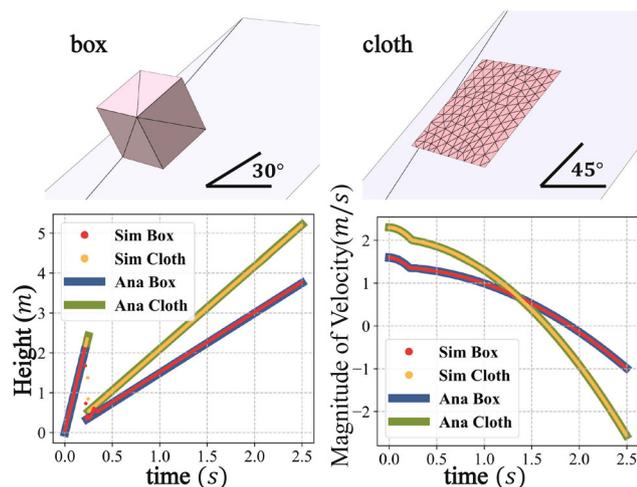


Fig. 7 Analytical validation of our contact simulation. We use slope angles 30 and 45 degrees and friction coefficients 0.4 and 0.7 for the box and cloth, respectively

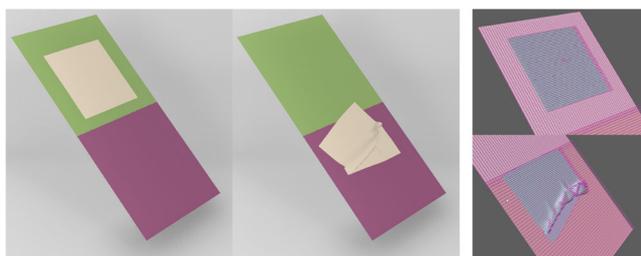


Fig. 8 Anisotropic friction test. Left: the result. Right: the visualization of the friction structure directions

7.1.4 Convergence of ADMM

Figure 9 presents the residual profile throughout ADMM iterations at frame 80 for the *twisting rods* and *ball on layers* scenes. Here, PR is the satisfiability of linear constraint in Eq. (3), while DR represents the overall convergence [7, 39]:

$$PR^{l+1} = \left\| (\mathbf{D} \mathbf{x}^{l+1} - \mathbf{z}) + (\mathbf{x}^{l+1} - \mathbf{x}_t - h\mathbf{p}) \right\|^2$$

$$DR^{l+1} = \left\| \mathbf{D}^T \mathbf{W}_e (\mathbf{z}^{l+1} - \mathbf{z}^l) + \mathbf{W}_c (\mathbf{p}^{l+1} - \mathbf{p}^l) \right\|^2$$

We prioritize PR over DR, as PR reflects the consistency of the final result with the contact projection outcome, which is usually more visually noticeable than the absolute solution accuracy.

The effectiveness of our approach relies on the judicious choice of W_c matrices. Typically, our solver experiences a rapid decrease in residuals within the initial 20 to 30 iterations. Periodic up-and-down jitter during collision detection intervals is a regular occurrence, updating the set of contact constraints and inducing fluctuations in residuals. This

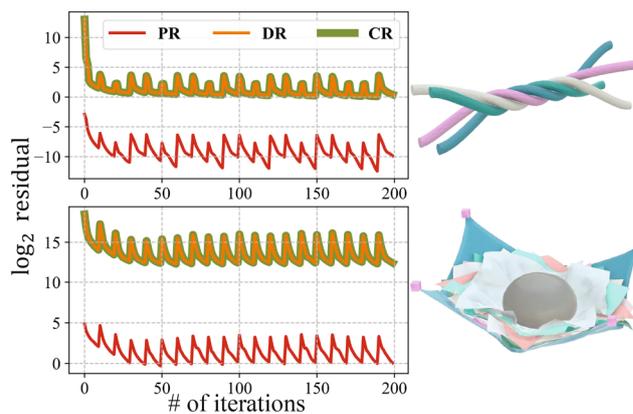


Fig. 9 Residual profile of two scenes at frame 80. DR: dual residual, PR: primal residual, CR: combined residual

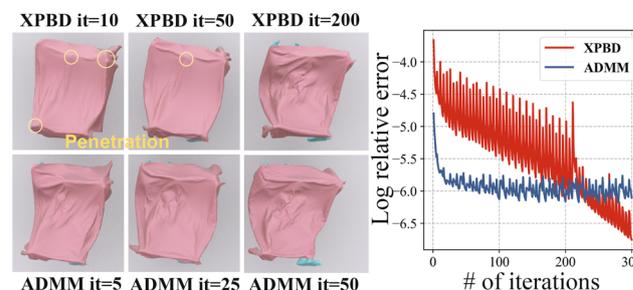


Fig. 10 Left: Curtain falling on Armadillo using XPBD and our method. Right: Log relative error on selected frame

behavior is expected and is also observed in other research, such as [19].

We also compare our solver with XPBD cloth, as depicted in Fig. 10, where a draped curtain interacts with Armadillo. With the identical PGS method for contact resolution applied for both methods, we observe a faster convergence for our method during the initial iterations, achieving a penetration-free state at a substantially earlier stage.

7.2 Ablation experiments

The integration of contact stabilization and Rayleigh damping constitutes a pivotal enhancement under certain circumstances. In the following, we present an evaluation of the efficacy of these incorporated features.

7.2.1 Contact stabilization

Figure 11 illustrates a challenging *multilayered* scene featuring 20 layers of cloth stacked on a static plane. The introduction of contact stabilization, with parameters $\alpha = 10^{-4}$ and $k_s = 10^6$, yields a stable and penetration-free outcome, achieved within a mere 20 ADMM iterations for each timestep lasting 0.033s. In contrast, omitting the stabilization

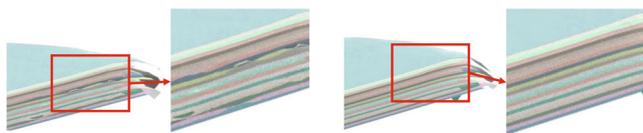


Fig. 11 Ablation experiments for contact stabilization. Left: 20 cloth layers jitter under 20 ADMM iterations due to penetration. Right: With contact stabilization, the cloth stack becomes stable of penetration-free

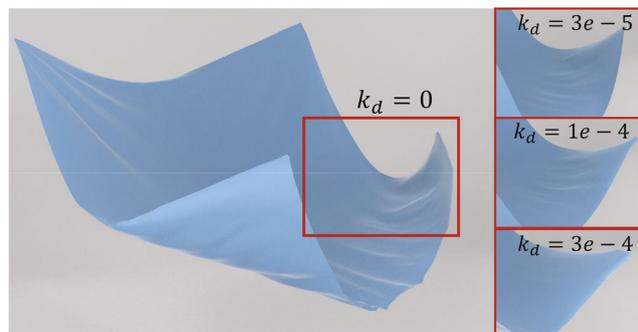


Fig. 12 Experiments on Rayleigh damping. A piece of fabric may exhibit small unnatural wrinkles at the edges in the absence of damping. However, by introducing a modest amount of damping (e.g., $k_d = 3 \times 10^{-5}$, 10^{-4} , 3×10^{-4}), these undesirable wrinkles can be effectively controlled without compromising the overall shape of the fabric, as linear momentum is conserved

technique leads to persistent jittering and penetration issues in the vertical cut view. This highlights the critical importance of our stabilization technique, especially in addressing intricate collisions, as exemplified by the challenging cloth stacking scenario.

7.2.2 Rayleigh damping

Rayleigh damping presents an effective approach for manipulating kinetic energy without incurring significant loss of linear momentum. Figure 12 depicts a scenario where Rayleigh damping can be used to control the formation of wrinkles while preserving the overall shape of the fabric.

7.3 Performance

Table 1 presents an overview of time statistics for the representative scenes, showcasing the computational efficiency of our method. The achievement of interactive speeds in the majority of cases is attributed to the efficiency of our novel ADMM scheme and our lightweight contact solver. Notably, scenes characterized by intricate collisions reveal that the primary time-consuming factors is the PGS step.

It is important to emphasize that all experiments were conducted on a multi-core CPU, utilizing a rudimentary collision detector and PGS solver. The prospect for achieving further acceleration is substantial, including the exploration of GPU

acceleration, the parallelization of the PGS solver through graph partitioning, or the implementation of an enhanced collision detector in subsequent iterations of our work. However, considering that these are non-trivial and beyond the scope of this research, we leave it as future developments.

8 Conclusion and future works

This paper introduces a novel approach to simulate soft body dynamics with frictional contacts through the incorporation of ADMM. Our method strategically combines a locally parallel elastic deformation projection with an iteratively frictional contact-solving method. The efficiency of our approach is underscored by the rapid convergence of the contact solver, necessitating only a minimal number of PGS iterations in each ADMM iteration to reach the penetration-free state. Moreover, the parallelization and prefactorization contribute to the scalability of our method, rendering its applicability for interactive moderate-scale simulations. The integration of the contact stabilization, anisotropic friction, and Rayleigh damping further enhances the realism of our solver. Thorough experiments validate the efficacy of our method, highlighting its high performance and ability to produce visually plausible and accurate effects in simulated scenarios.

While our current work is practical, there is room for improvement. Despite handling complex scenarios, our program underutilizes the multi-core CPU due to bottlenecks in the sequential PGS. As mentioned by [34], a fast graph partition of the system could maximize the parallelism in PGS. Alternatively, a fast GPU-based solver is another promising future direction. Additionally, investing in a fast and robust continuous collision detection scheme could also enhance the method's performance and reliability in complex interactions.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s00371-024-03438-8>.

Acknowledgements This work was supported in part by the National Key R&D Program of China (no. 2023YFC3604505), Natural Science Foundation of China (no. U20A20195, 62272017, 62172437), the Postdoctoral Fellowship Program of CPSF under grant number GZC20233375, Beijing Natural Science Foundation (L232065), CAS Interdisciplinary Project (JCTD- 2020-11).

Author Contributions Siyan Zhu: Writing—original draft, Visualization, Validation, Software, Methodology. Cheng Fang: Writing—original draft, Visualization, Validation, Software, Methodology. Peng Yu: Writing—review & editing, Validation, Resources&Material, Funding acquisition. Xiao Zhai: Writing—original draft, review & editing, Methodology, Math derivation. Aimin Hao: Project administration, Resources&Material, Funding acquisition. Junjun Pan: Writing - review & editing, Project administration, Funding acquisition.

Data availability No datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors declare no Conflict of interest.

References

- Agudo, A.: Unsupervised 3d reconstruction and grouping of rigid and non-rigid categories. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**, 519–532 (2020)
- Andrews, S., Erleben, K., Ferguson, Z.: Contact and friction simulation for computer graphics. In: *ACM SIGGRAPH 2022 Courses*, pp. 1–172 (2022)
- Baraff, D., Witkin, A.: Dynamic simulation of non-penetrating flexible bodies. *ACM SIGGRAPH Comput. Graph.* **26**(2), 303–308 (1992)
- Baraff, D., Witkin, A.: Large steps in cloth simulation. In: *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '98*, pp. 43–54. Association for Computing Machinery, New York (1998)
- Bertails-Descoubes, F., Cadoux, F., Daviet, G., Acary, V.: A nonsmooth newton solver for capturing exact coulomb friction in fiber assemblies. *ACM Trans. Graph. (TOG)* **30**(1), 1–14 (2011)
- Bouaziz, S., Martin, S., Liu, T., Kavan, L., Pauly, M.: Projective dynamics: fusing constraint projections for fast simulation. *ACM Trans. Graph.* **33**(4) (2014)
- Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends @ Mach. Learn.* **3**(1), 1–122 (2011)
- Chan, S.H., Wang, X., Elgendy, O.A.: Plug-and-play ADMM for image restoration: fixed-point convergence and applications. *IEEE Trans. Comput. Imaging* **3**(1), 84–98 (2016)
- Chen, Y.L., Ly, M., Wojtan, C.: Unified treatment of contact, friction and shock-propagation in rigid body animation. In: *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 1–2 (2023)
- Daviet, G.: Simple and scalable frictional contacts for thin nodal objects. *ACM Trans. Graph. (TOG)* **39**(4), 61–1 (2020)
- Daviet, G.: Interactive hair simulation on the GPU using ADMM. In: *ACM SIGGRAPH 2023 Conference Proceedings*, pp. 1–11 (2023)
- Deul, C., Charrier, P., Bender, J.: Position-based rigid-body dynamics. *Comput. Animat. Virtual Worlds* **27**(2), 103–112 (2016)
- Erleben, K., Macklin, M., Andrews, S., Kry, P.G.: The matchstick model for anisotropic friction cones. In: *Computer Graphics Forum*, vol. 39, pp. 450–461. Wiley Online Library, Hoboken (2020)
- Fang, Y., Li, M., Gao, M., Jiang, C.: Silly rubber: an implicit material point method for simulating non-equilibrated viscoelastic and elastoplastic solids. *ACM Trans. Graph. (TOG)* **38**(4), 1–13 (2019)
- Frâncu, M., Moldoveanu, F.: Position based simulation of solids with accurate contact handling. *Comput. Graph.* **69**, 12–23 (2017)
- Gast, T.F., Schroeder, C., Stomakhin, A., Jiang, C., Teran, J.M.: Optimization integrator for large time steps. *IEEE Trans. Vis. Comput. Graph.* **21**(10), 1103–1115 (2015)
- Grigorev, A., Black, M.J., Hilliges, O.: Hood: hierarchical graphs for generalized modelling of clothing dynamics. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16965–16974 (2023)
- Han, H., Sun, M., Zhang, S., Liu, D., Liu, T.: Gpu cloth simulation pipeline in lightchaser animation studio. In: *SIGGRAPH Asia 2021 Technical Communications*, pp. 1–4 (2021)
- Lan, L., Ma, G., Yang, Y., Zheng, C., Li, M., Jiang, C.: Penetration-free projective dynamics on the GPU. *ACM Trans. Graph. (TOG)* **41**(4), 1–16 (2022)
- Lee, J., Lee, M., Lee, D.: Modular and parallelizable multibody physics simulation via subsystem-based ADMM. *arXiv preprint arXiv:2302.14344* (2023)
- Li, J., Daviet, G., Narain, R., Bertails-Descoubes, F., Overby, M., Brown, G.E., Boissieux, L.: An implicit frictional contact solver for adaptive cloth simulation. *ACM Trans. Graph. (TOG)* **37**(4), 1–15 (2018)
- Li, J., Liu, T., Kavan, L.: Laplacian damping for projective dynamics. In: *Proceedings of the 14th Workshop on Virtual Reality Interactions and Physical Simulations*, pp. 29–36 (2018)
- Li, M., Ferguson, Z., Schneider, T., Langlois, T.R., Zorin, D., Panozzo, D., Jiang, C., Kaufman, D.M.: Incremental potential contact: intersection-and inversion-free, large-deformation dynamics. *ACM Trans. Graph.* **39**(4), 49 (2020)
- Li, M., Kaufman, D.M., Jiang, C.: Codimensional incremental potential contact. *ACM Trans. Graph.* **40**(4) (2021)
- Ly, M., Jouve, J., Boissieux, L., Bertails-Descoubes, F.: Projective dynamics with dry frictional contact. *ACM Trans. Graph. (TOG)* **39**(4), 57 (2020)
- Macklin, M., Erleben, K., Müller, M., Chentanez, N., Jeschke, S., Kim, T.Y.: Primal/dual descent methods for dynamics. In: *Computer Graphics Forum*, vol. 39, pp. 89–100. Wiley Online Library, Hoboken (2020)
- Macklin, M., Erleben, K., Müller, M., Chentanez, N., Jeschke, S., Makovychuk, V.: Non-smooth newton methods for deformable multi-body dynamics. *ACM Trans. Graph. (TOG)* **38**(5), 1–20 (2019)
- Macklin, M., Müller, M., Chentanez, N.: XPBD: position-based simulation of compliant constrained dynamics. In: *Proceedings of the 9th International Conference on Motion in Games*, pp. 49–54 (2016)
- Macklin, M., Müller, M., Chentanez, N., Kim, T.Y.: Unified particle physics for real-time applications. *ACM Trans. Graph. (TOG)* **33**(4), 1–12 (2014)
- Müller, M., Heidelberger, B., Hennix, M., Ratcliff, J.: Position based dynamics. *J. Vis. Commun. Image Represent.* **18**(2), 109–118 (2007)
- Müller, M., Macklin, M., Chentanez, N., Jeschke, S., Kim, T.Y.: Detailed rigid body simulation with extended position based dynamics. In: *Computer Graphics Forum*, vol. 39, pp. 101–112. Wiley Online Library, Hoboken (2020)
- Narain, R., Overby, M., Brown, G.E.: Admm \supseteq projective dynamics: fast simulation of general constitutive models. In: *Symposium on Computer Animation*, vol. 1, p. 2016 (2016)
- Otaduy, M.A., Tamstorf, R., Steinemann, D., Gross, M.: Implicit contact handling for deformable objects. In: *Computer Graphics Forum*, vol. 28, pp. 559–568. Wiley Online Library, Hoboken (2009)
- Peiret, A., Andrews, S., Kövecses, J., Kry, P.G., Teichmann, M.: Schur complement-based substructuring of stiff multibody systems with contact. *ACM Trans. Graph. (TOG)* **38**(5), 1–17 (2019)
- Peng, Y., Deng, B., Zhang, J., Geng, F., Qin, W., Liu, L.: Anderson acceleration for geometry optimization and physics simulation. *ACM Trans. Graph. (TOG)* **37**(4), 1–14 (2018)
- Servin, M., Lacoursiere, C., Melin, N.: Interactive simulation of elastic deformable materials. In: *Proceedings of Sigrad Conference*, pp. 22–32. Citeseer (2006)
- Shi, A., Kim, T.: A unified analysis of penalty-based collision energies. *Proc. ACM Comput. Graph. Interact. Tech.* **6**(3), 1–19 (2023)

38. Smith, B., Goes, F.D., Kim, T.: Stable Neo-Hookean flesh simulation. *ACM Trans. Graph. (TOG)* **37**(2), 1–15 (2018)
39. Tasora, A., Mangoni, D., Benatti, S., Garziera, R.: Solving variational inequalities and cone complementarity problems in nonsmooth dynamics using the alternating direction method of multipliers. *Int. J. Numer. Methods Eng.* **122**(16), 4093–4113 (2021)
40. Terzopoulos, D., Platt, J., Barr, A., Fleischer, K.: Elastically deformable models. In: *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 205–214 (1987)
41. Tournier, M., Nesme, M., Gilles, B., Faure, F.: Stable constrained dynamics. *ACM Trans. Graph. (TOG)* **34**(4), 1–10 (2015)
42. Verschoor, M., Jalba, A.C.: Efficient and accurate collision response for elastically deformable models. *ACM Trans. Graph. (TOG)* **38**(2), 1–20 (2019)
43. Wang, H., Yang, Y.: Descent methods for elastic body simulation on the GPU. *ACM Trans. Graph. (TOG)* **35**(6), 1–10 (2016)
44. Zhang, J., Peng, Y., Ouyang, W., Deng, B.: Accelerating ADMM for efficient simulation and optimization. *ACM Trans. Graph. (TOG)* **38**(6), 1–21 (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.



Siyan Zhu is currently pursuing a Master's degree at the School of Computer Science and Engineering, Beihang University. His primary research interests revolve around physics-based animation, with a focus on fast collision handling and AI-accelerated cloth simulation.



Cheng Fang is currently pursuing a Master's degree at the School of Computer Science and Engineering, Beihang University. Her primary research interests lie in computer graphics, physical simulation, and virtual surgery.



Peng Yu is currently a Postdoctoral Researcher at Beihang University, where he also completed his Ph.D. in 2022 from the School of Computer Science and Engineering. His research is centered on the innovative application of physical simulation techniques within the realm of virtual reality (VR) for surgical simulation. His work encompasses the simulation of both rigid and soft body dynamics, the modeling of intricate interactions between these entities, haptic rendering, and the quantitative assessment of VR simulators using physiological data. Dr. Yu's contributions to the field have been recognized through numerous publications in prestigious academic journals and at international conferences, including IEEE VR, Computer Methods and Programs in Biomedicine, BMC Medical Education, and CAVW. His research achievements enabled him to obtain the Postdoctoral Fellowship Program of CPSF.



Xiao Zhai is a simulation researcher at Weta FX. His work focuses on the computational physics technologies used in VFX productions, including the Academy Award-winning *Avatar: The Way of Water*. He holds a Ph.D. in Computer Applied Technologies from Beihang University, Beijing, China.



Aimin Hao received the BS, MS, and PhD degrees in computer science from Beihang University. He is a professor with Computer Science School and the associate director with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University. His research interests are in virtual reality, computer simulation, computer graphics, geometric modeling, image processing, and computer vision.



Junjun Pan is currently a professor in School of Computer Science, Beihang University. He received both BS and MS degrees in School of Computer Science, Northwestern Polytechnical University, China. In 2006, he studied in National Centre for Computer Animation (NCCA), Bournemouth University, UK, as PhD candidate with full scholarship. In 2010, he received the PhD degree and worked in NCCA as Postdoctoral Research Fellow. From 2012 to 2013, he worked as a Research

Associate in Center for Modeling, Simulation and Imaging in Medicine, Rensselaer Polytechnic Institute, USA. In November 2013, he was appointed as Associate Professor in School of Computer Science, Beihang University, China. His research interests include virtual surgery and computer animation.