IsaacIPC: Coupling High-Fidelity Simulation and Realistic Rendering for Contact-Rich Robotic Systems

Qixin Liang
Anker Humanoid Lab
Shenzhen, China
The University of Hong Kong
Hong Kong SAR, China

Zhongqing Han
Anker Humanoid Lab
Shenzhen, China

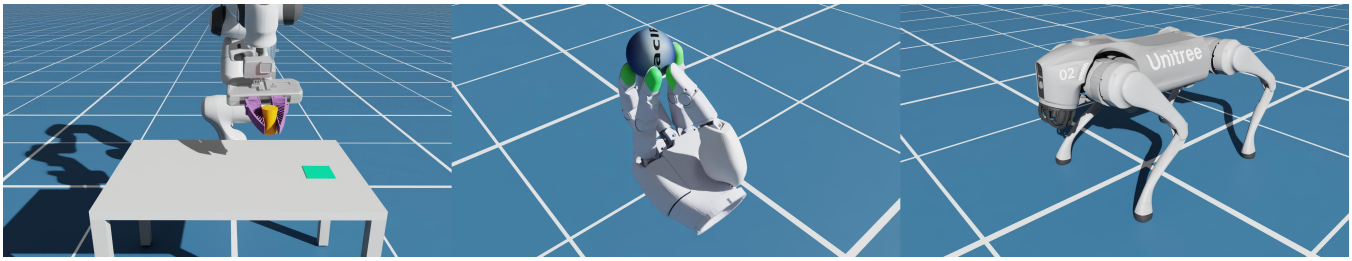

Figure 1: IsaacIPC qualitative demonstrations across rigid–deformable robotic systems: a soft Universal Manipulation Interface (UMI) gripper, a dexterous hand, and a quadruped robot.

Abstract

We present IsaacIPC, a robotic simulation framework that couples GPU accelerated incremental potential contact (IPC) with Isaac Sim / Isaac Lab. IsaacIPC maps simulated deformation between simulation and visual meshes, enabling real-time realistic rendering with applications to data collection and policy evaluation. For tactile sensing, we introduce the geometric mortar contact potential (GMCP), which defines a barrier potential over contact samples on tactile surfaces to better resolve contact-pressure distributions. We evaluate GMCP on contact benchmarks and demonstrate IsaacIPC on rigid–deformable robotic simulations including a quadruped robot, a dexterous hand, and a universal manipulation interface (UMI) gripper.

CCS Concepts

• **Computing methodologies** → **Physical simulation**; **Robotics**.

Keywords

Contact Simulation, Robotic Simulation, Tactile, Embodied Intelligence, Data Engine, IsaacSim/Lab

1 Introduction

High-fidelity robotic simulation frameworks are increasingly important for embodied intelligence: they enable scalable policy training, systematic evaluation, and contact-rich interactions that are expensive to collect in the real world [Genesis Authors 2024; Mittal et al. 2025]. Contact simulation is central to these systems because manipulation, locomotion, and tactile sensing all rely heavily on physical interaction with the environment. Incremental Potential Contact (IPC) [Li et al. 2020] is attractive in this setting because its barrier formulation, combined with continuous collision detection, robustly maintains intersection- and inversion-free simulation under large deformations and challenging contact.

Recent work has improved IPC along complementary directions. For efficiency, IPC has been accelerated by projective dynamics [Lan et al. 2022b], preconditioned nonlinear conjugate gradients [Shen et al. 2024], and GPU Gauss–Newton solvers [Huang et al. 2024, 2025a]. For accuracy, Convergent IPC [Li et al. 2023] and GCP [Huang et al. 2025b] study continuous contact potentials and their discretization. These developments have made IPC-based contact simulation increasingly useful in robotics: IPC-GraspSim [Kim et al. 2022] applies IPC to grasp simulation, while TacIPC [Du et al. 2024a], Taccel [Li et al. 2025], TacEx [Nguyen et al. 2024], UniVTAC [Chen et al. 2026], and Tac2Real [Yan et al. 2026] use IPC or its variants for tactile simulation, data generation, and policy learning.

Tactile sensing places a strong requirement on contact pressure accuracy, as many tactile sensors rely on contact-induced deformation to generate signals. TacEx [Nguyen et al. 2024] integrates GIPC [Huang et al. 2024] with Isaac Sim / Isaac Lab [Mittal et al. 2025] for GelSight visuotactile sensor [Yuan et al. 2015], but GIPC-modeled objects are not mapped to high-fidelity visual assets with textures and material appearance for NVIDIA Omniverse [NVIDIA 2026]. UniVTAC [Chen et al. 2026] builds on TacEx for visuo-tactile data generation and benchmarking, and inherits this limitation. This leaves a practical gap for real-to-sim workflows that require robust deformable contact, scalable rollout, and high-fidelity rendering from detailed visual assets.

In this work, we make two key contributions.

- First, we introduce IsaacIPC, a simulation framework that bridges IsaacSim/Lab [Mittal et al. 2025] with libuipc [spiriMirror 2026] through a dual-mesh mapper, combining high-fidelity contact simulation with realistic rendering for data collection and robot-learning policy evaluation.

- Second, we propose the Geometric Mortar Contact Potential (GMCP), a mortar-based contact formulation that samples contact on tactile surfaces and incorporates the barrier potential to improve contact-pressure simulation for tactile sensing.

Together, these contributions offer a practical foundation for improving contact-rich simulation in robotic interactions.

2 Related Work

2.1 Barrier-Based Contact Simulation

Incremental Potential Contact (IPC) [Li et al. 2020] formulates contact as a smoothed barrier potential with continuous collision detection (CCD), guaranteeing intersection- and inversion-free trajectories under implicit time integration. This framework has been extended along multiple axes: to rigid bodies with curved-trajectory CCD [Ferguson et al. 2021], to stiff affine bodies with reduced degrees of freedom [Lan et al. 2022a], to unified rigid-deformable multibody systems with articulation constraints [Chen et al. 2022], to high-order finite element meshes [Ferguson et al. 2023a], and to adaptive remeshing within the timestep [Ferguson et al. 2023b].

A major thrust of recent work accelerates IPC to practical speeds. Lan et al. [Lan et al. 2022b] integrate the barrier into projective dynamics on the GPU. Shen et al. [Shen et al. 2024] eliminate Hessian assembly via a preconditioned nonlinear conjugate gradient (PNCG) method, and Zhang et al. [Zhang et al. 2026] further improve this with multilevel preconditioning. GIPC [Huang et al. 2024] derives analytic eigensystems for GPU-friendly Gauss–Newton solves, and StiffGIPC [Huang et al. 2025a] adds multilevel preconditioning and affine-deformable coupling for stiff materials. Barrier-Augmented Lagrangian [Guo et al. 2024] combines barriers with augmented Lagrangian for improved conditioning, while AL-IPC [Zheng et al. 2026] replaces barriers entirely with a second-order augmented Lagrangian, avoiding TOI locking.

On the accuracy side, Convergent IPC [Li et al. 2023] reformulates IPC in the continuous setting and gives a convergent discretization, with convergence demonstrated under joint refinement of the mesh, time step, and contact activation distance. The Geometric Contact Potential (GCP) [Huang et al. 2025b] derives a continuum barrier from geometric interaction sets and uses an adaptive locality parameter to avoid rest-state spurious forces while relaxing the coupling between barrier extent and mesh resolution. Classical mortar contact methods improve interface consistency by enforcing contact constraints in a weak, integrated form over overlapping slave-master regions, and are widely used for accurate contact pressure transfer on nonmatching meshes [Farah et al. 2018; Puso and Laursen 2004; Wohlmuth 2000]. They are typically formulated with Lagrange multiplier or augmented Lagrangian enforcement rather than feasible-iterate barrier formulations [Huang et al. 2025b]; our work instead incorporates mortar integration into a barrier potential to improve contact pressure accuracy while retaining nonpenetration.

2.2 IPC in Robotic Simulation

IPC has recently been adopted in robotics applications involving robust contact handling and deformable objects. IPC-GraspSim [Kim et al. 2022] introduced IPC to parallel-jaw grasp simulation with compliant fingertips, showing that explicitly modeling soft jaw deformation and frictional contact can improve grasp outcome prediction. ZeMa [Du et al. 2024b] further extended IPC-based simulation to unified rigid-deformable robotic manipulation, combining (finite element method) FEM and affine rigid bodies for applications including grasp generation, grasp repair, and reinforcement learning. Embedded IPC [Du et al. 2024c] improves the runtime of this line of work by using a reduced subspace for elasticity while retaining contact constraints on an embedded high-resolution surface, targeting interactive robot manipulation scenarios.

IPC-based methods have also been explored for visuo-tactile sensing, where elastomer deformation and frictional contact directly determine the tactile signal. TacIPC [Du et al. 2024a] uses FEM with IPC to simulate optical tactile sensor gels with intersection- and inversion-free deformation, enabling tactile image and marker displacement prediction. TacEx [Nguyen et al. 2024] integrates GIPC [Huang et al. 2024] with Isaac Sim / Isaac Lab [Mittal et al. 2025], combining FEM with visuotactile rendering modules for GelSight sensors [Yuan et al. 2015]. Taccel [Li et al. 2025] scales this direction further by combining IPC and affine body dynamics with NVIDIA Warp [Macklin 2022] to support many parallel tactile-robotics environments. UniVTAC [Chen et al. 2026] builds on TacEx [Nguyen et al. 2024] from the data and evaluation side, providing a unified visuo-tactile simulation platform, representation-learning pipeline, and benchmark for tactile-driven manipulation. Tac2Real [Yan et al. 2026] targets online reinforcement learning and zero-shot sim-to-real transfer, using PNCG-IPC [Shen et al. 2024] as the contact solver in a visuotactile simulation framework. These works demonstrate the value of IPC-quality contact for robotics, while also motivating simulation frameworks that combine robust deformable contact, scalable GPU execution, and direct integration with modern robot-learning environments.

3 IsaacIPC

In this section, we present IsaacIPC (Fig. 2), a simulation framework that couples the high-fidelity real-time rendering and massively parallel environment infrastructure of Isaac Sim / Isaac Lab [Mittal et al. 2025] with the GPU-accelerated contact simulation of libuipc [Huang et al. 2024, 2025a; spiriMirror 2026], incorporating our GMCP contact model (§4.1) and a Dual-Mesh Mapper (§3.2).

3.1 System Architecture

IsaacIPC adopts a three-layer architecture (Fig. 2). The *IsaacSim/Lab* layer provides robot articulation and photorealistic rendering via the Universal Scene Description (USD) / Fabric stage, with an optional one-way coupling to PhysX [NVIDIA Corporation 2025]. The *IsaacIPC* layer mediates between IsaacSim/Lab and libuipc: the *ExternalSim Bridge* coordinates each physics step; the *Articulation Constraint* reads joint-driven body poses and applies them as constraints inside libuipc; the *Dual-Mesh Mapper* (§3.2) interpolates the visual mesh $\mathbb{M}_{\text{vis}}$ onto the simulation mesh $\mathbb{M}_{\text{sim}}$ each frame; and the *Fabric Bridge* writes the interpolated $\mathbb{M}_{\text{vis}}$ vertex positions back to the USD / Fabric stage. The *libuipc* layer runs all physics on the GPU: the *Simulations* block handles rigid and deformable body dynamics with GPU contact, of which GMCP (§4.1) is one contact model; the *Subscene Tabular* layout partitions the simulation

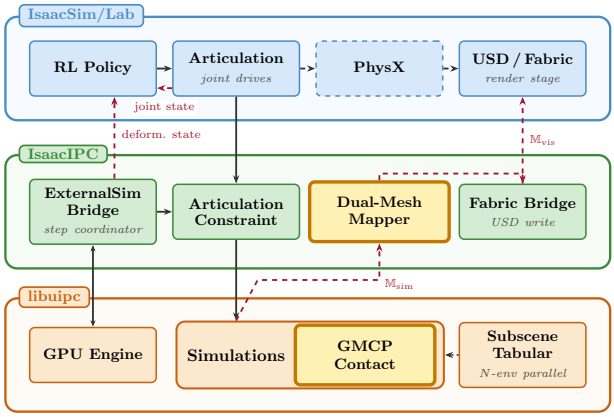

Figure 2: System architecture of IsaacIPC.

into multiple fully isolated subscenes on a single GPU, enabling massively parallel rollouts.

3.2 Dual-Mesh Mapper

Reconstructed or procedurally generated simulation assets [Tencent Hunyuan3D Team 2025] typically provide only surface meshes, whereas deformable simulation requires volumetric discretization (e.g., tetrahedralization via TetGen [Si 2015]), which may alter mesh topology and break texture-coordinate parameterization and material bindings essential for photorealistic rendering. Inspired by barycentric embedding in skeleton-driven animation [Capell et al. 2002] and Embedded IPC [Du et al. 2024c], we therefore maintain two meshes: a *simulation mesh* $\mathbb{M}_{\text{sim}}$ for contact and deformation, and a *visual mesh* $\mathbb{M}_{\text{vis}}$ for rendering, preserving the original texture coordinates, materials bindings, and topology.

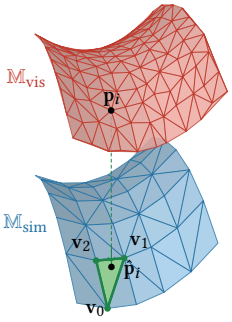

Figure 3: Dual-mesh mapper.

Precomputation. At initialization, each visual vertex $\mathbf{p}_i \in \mathcal{V}$ ($i = 1, \ldots, N$) is embedded onto the surface of the simulation mesh $\mathbb{M}_{\text{sim}}$ via closest-point projection. For each $\mathbf{p}_i$, we find the nearest simulation triangle $T_i \in \mathbb{M}_{\text{sim}}$ (subscript i labels the assignment to vertex $\mathbf{p}_i$, not a global enumeration of triangles) with vertices $\{\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2\}$, and compute its barycentric coordinates (u_i, v_i, w_i) such that

$$\hat{\mathbf{p}}_i = u_i\,\mathbf{v}_0 + v_i\,\mathbf{v}_1 + w_i\,\mathbf{v}_2, \qquad u_i + v_i + w_i = 1, \tag{1}$$

where $\hat{\mathbf{p}}_i$ is the projection of $\mathbf{p}_i$ onto T_i. Let $\mathbf{e}_1 = \mathbf{v}_1 - \mathbf{v}_0$, $\mathbf{e}_2 = \mathbf{v}_2 - \mathbf{v}_0$, and $\mathbf{n}_i = (\mathbf{e}_1 \times \mathbf{e}_2)/\|\mathbf{e}_1 \times \mathbf{e}_2\|$ be the unit normal of T_i. Setting $\mathbf{d} = \mathbf{p}_i - \mathbf{v}_0$ and $\delta_i = \mathbf{d} \cdot \mathbf{n}_i$ (signed offset of $\mathbf{p}_i$ from the plane of T_i), we have

$$\mathbf{d} = v_i\,\mathbf{e}_1 + w_i\,\mathbf{e}_2 + \delta_i\,\mathbf{n}_i. \tag{2}$$

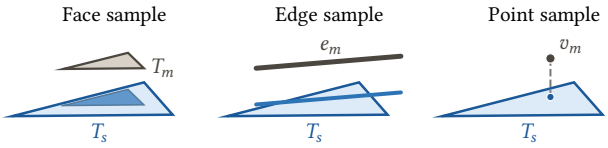

Figure 4: Contact sampling on a slave triangle, which is extracted from the tactile surface.

Taking the dot product of Eq. (2) with $\mathbf{e}_1$ and $\mathbf{e}_2$ in turn, the $\delta_i \mathbf{n}_i$ term vanishes since $\mathbf{n}_i \cdot \mathbf{e}_1 = \mathbf{n}_i \cdot \mathbf{e}_2 = 0$, yielding

$$\begin{pmatrix} \mathbf{e}_1 \cdot \mathbf{e}_1 & \mathbf{e}_1 \cdot \mathbf{e}_2 \\ \mathbf{e}_1 \cdot \mathbf{e}_2 & \mathbf{e}_2 \cdot \mathbf{e}_2 \end{pmatrix} \begin{pmatrix} v_i \\ w_i \end{pmatrix} = \begin{pmatrix} \mathbf{d} \cdot \mathbf{e}_1 \\ \mathbf{d} \cdot \mathbf{e}_2 \end{pmatrix}, \tag{3}$$

with $u_i = 1 - v_i - w_i$. The mapping $\{(T_i,\, u_i,\, v_i,\, w_i,\, \delta_i)\}_{i=1}^{N}$ is computed once and stored.

Per-frame update. After each simulation step, the simulation vertices are updated to deformed positions. The visual vertex positions are recovered by interpolating the deformed simulation vertices and restoring the normal offset:

$$\mathbf{p}'_i = u_i\,\mathbf{v}'_0 + v_i\,\mathbf{v}'_1 + w_i\,\mathbf{v}'_2 + \delta_i\,\mathbf{n}'_i, \qquad i = 1, \ldots, N, \tag{4}$$

where $\mathbf{v}'_0, \mathbf{v}'_1, \mathbf{v}'_2$ are the deformed positions of the three vertices of T_i, and $\mathbf{n}'_i$ is the unit normal of the deformed triangle T_i. The mesh topology, texture coordinates, and material bindings of $\mathbb{M}_{\text{vis}}$ remain unchanged throughout; only the vertex positions $\{\mathbf{p}'_i\}$ are updated each frame.

The mapping is rigid-invariant and introduces a piecewise-linear approximation error of $O(h^2)$ for general deformations, where h is the simulation triangle edge length; rendering fidelity thus improves quadratically with mesh refinement.

4 Geometric Mortar Contact Potential

We consider contacts in which pressure is evaluated on a tactile surface. We use this surface as the slave geometry. The opposing geometry is the master side, whose local contact feature may be a surface patch, a line segment, or a point.

4.1 Contact Sampling

The slave triangulated mesh $\mathcal{S}$ provides the local parameterization and normal field $\mathbf{n}_s$. Contact samples are constructed by projecting and clipping master features against local slave triangles. We refer to samples on clipped face overlaps as *face samples*, samples on clipped edge segments as *edge samples*, and projected master vertices as *point samples* (see Fig. 4). Each sample k is either a quadrature point on a clipped face overlap or edge segment, or a projected master vertex. It stores a position $\mathbf{x}_{s,k}$ on $\mathcal{S}$, an associated master position $\mathbf{x}_{m,k}$, a sample weight w_k, and a signed gap g_k measured along the slave normal.

Face samples. For a slave triangle T_s and a master triangle T_m, we project both triangles onto the tangent plane of T_s and clip their overlap using Sutherland-Hodgman clipping [Sutherland and Hodgman 1974]. The clipped polygon is fan-triangulated and integrated with 2D Gauss quadrature. At sample k, barycentric coordinates $\beta^s_{a,k}$ and $\beta^m_{a,k}$ interpolate the slave and master positions,

$$\mathbf{x}_{s,k} = \textstyle\sum_a \beta^s_{a,k}\mathbf{x}_{s,a}, \qquad \mathbf{x}_{m,k} = \textstyle\sum_a \beta^m_{a,k}\mathbf{x}_{m,a} \qquad a = 0, 1, 2. \tag{5}$$

The signed gap is

$$g_k = \mathbf{n}_s \cdot (\mathbf{x}_{m,k} - \mathbf{x}_{s,k}). \tag{6}$$

Edge samples. For a master edge $e_m = (\mathbf{x}^e_{m,0}, \mathbf{x}^e_{m,1})$, we project the edge onto the tangent plane of a slave triangle and clip the projected segment by the triangle. Using the edge parameter $\eta \in [0, 1]$,

$$\mathbf{x}_{m,k} = (1 - \eta_k)\mathbf{x}^e_{m,0} + \eta_k \mathbf{x}^e_{m,1}, \tag{7}$$

each 1D Gauss point gives slave barycentric coordinates $\beta^s_{a,k}$ on T_s and

$$\mathbf{x}_{s,k} = \textstyle\sum_a \beta^s_{a,k} \mathbf{x}_{s,a}, \qquad a = 0, 1, 2. \tag{8}$$

The signed gap is

$$g_k = \mathbf{n}_s \cdot (\mathbf{x}_{m,k} - \mathbf{x}_{s,k}). \tag{9}$$

Point samples. For a master vertex, we project this vertex onto a slave triangle and generate a point sample. The sample uses the projected slave position $\mathbf{x}_{s,k}$ and the master vertex position $\mathbf{x}_{m,k}$ to compute the gap g_k as in Eq. (6) and Eq. (9).

4.2 GMCP Barrier Potential

Let $\mathcal{K}_q$ denote the contact samples of type q on the slave surface. The discrete barrier potential is

$$\Psi = \sum_{q \in \{2,1,0\}} \sum_{k \in \mathcal{K}_q} \kappa_q \, w_k \, \gamma_k \, B(g_k, \varepsilon_k), \tag{10}$$

where $q = 2, 1, 0$ denotes face, edge, and point samples, w_k is the sample weight, with units corresponding to area, length, or a point contribution, κ_q is the barrier stiffness for sample type q, γ_k is a master-feature weight defined in Sec. 4.3, and ε_k is the barrier support radius defined in Sec. 4.4. The barrier function B follows IPC [Li et al. 2020]; other function, such as GCP [Huang et al. 2025b], can also be adopted.

For the current contact samples, we use the gradient and Hessian approximation

$$\nabla \Psi = \sum_q \sum_{k \in \mathcal{K}_q} \kappa_q w_k \gamma_k \frac{\partial B}{\partial g_k} \nabla g_k, \tag{11}$$

$$\nabla^2 \Psi = \sum_q \sum_{k \in \mathcal{K}_q} \kappa_q w_k \gamma_k \frac{\partial^2 B}{\partial g_k^2} \nabla g_k (\nabla g_k)^T, \tag{12}$$

with Hessian contributions projected to the positive semi-definite cone.

4.3 Master-Feature Weights γ

The weight γ controls transitions between face, edge, and point samples on the master side. We use the C^1 Hermite step to smoothly scale sample contributions near master-feature boundaries.

$$H(x, \delta) = \begin{cases} 0 & x \le 0, \\ 3t^2 - 2t^3, \quad t = x/\delta & 0 < x < \delta, \\ 1 & x \ge \delta, \end{cases} \tag{13}$$

For face samples, we use

$$f_T = H\Big(\min_a \beta^m_a, \delta_T\Big), \tag{14}$$

where β^m_a are barycentric coordinates on the master triangle and δ_T is the transition width near master-triangle edges. For edge samples, we use

$$f_e = H(\eta, \delta_e) H(1 - \eta, \delta_e). \tag{15}$$

Here η is the master edge parameter and δ_e is the transition width near master-edge endpoints. The master-feature weight is

$$\gamma = \begin{cases} f_T & \text{face sample}, \\ f_e & \text{edge sample}, \\ 1 & \text{point sample}. \end{cases} \tag{16}$$

Thus face samples decay near master-triangle edges, edge samples decay near master-edge endpoints, and point samples keep full weight.

4.4 Adaptive Barrier Support Radius

Inspired by GCP [Huang et al. 2025b], we use an adaptive support radius ε_k for each contact sample k to avoid spurious forces. We assign the support radius as:

$$\varepsilon_k = \min\big(0.9\, g_{\text{ref},k}, \varepsilon_{\max}\big), \tag{17}$$

where $g_{\text{ref},k}$ is the gap to the associated master feature, and $\varepsilon_{\max}$ is a prescribed support-radius bound. This choice gives $g_{\text{ref},k} > \varepsilon_k$ which avoids spurious forces.

4.5 Linearized Gap Step Size

For the current contact samples, we compute a step-size bound from the linearized gap. For a displacement $\Delta\mathbf{x}$ that decreases sample k's gap, we use

$$\alpha^*_k = 0.9 \, \frac{g_k}{-\nabla g_k \cdot \Delta \mathbf{x}}, \qquad \nabla g_k \cdot \Delta \mathbf{x} < 0. \tag{18}$$

The global step size is

$$\alpha = \min_k \alpha^*_k, \tag{19}$$

where the minimum is taken over all contact samples k with $\nabla g_k \cdot \Delta\mathbf{x} < 0$, with g_k and ∇g_k evaluated at the penetration-free positions.

5 Results

In this section, we evaluate GMCP and IsaacIPC through quantitative and qualitative tests. All tests were run on a workstation with an AMD EPYC 9T24 CPU with 32 cores and an NVIDIA RTX5880-Ada-48Q GPU.

5.1 Quantitative Tests for GMCP

In this subsection, E is Young's modulus, ν is Poisson's ratio, u_x, u_y, and u_z are displacement components along the x, y, and z axes, and σ_{ij}, $i, j \in \{x, y, z\}$ is a Cauchy stress component. All quantities are reported in SI units. GMCP is implemented in libuipc, which branches from `main` at commit `1dd8e36`.

Contact patch test. The contact patch test checks whether a contact method transmits a uniform stress state across a non-matching interface [Wriggers and Laursen 2006]. We use two linear-elastic blocks of size $1 \times 1 \times 0.5$, separated by an initial gap of 0.002.

The bottom block has 56 vertices and 125 tetrahedra, while the top block has 32 vertices and 70 tetrahedra. Both use $E = 1000$ and $\nu = 0$. The bottom surface of the bottom block is constrained with $u_x = u_y = u_z = 0$, the top block is constrained with $u_x = u_y = 0$, and a uniform pressure of 10 is applied to the top surface of the top block along $-z$. All runs are quasi-static, frictionless, and use barrier support radius 0.001.

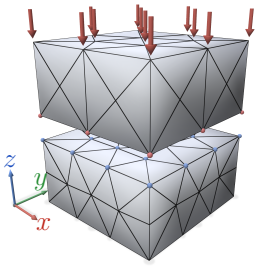

Figure 5: Contact patch test.

The analytical solution is $\sigma_{zz} = -10$ uniformly in both blocks, with all other stress components equal to zero. We report the maximum relative σ_{zz} error and the maximum spurious stress, denoted σ_{spur}, defined as the largest absolute non-target Cauchy stress component among σ_{xx}, σ_{yy}, σ_{xy}, σ_{yz}, and σ_{xz}. As shown in Table 1, GMCP transmits the target uniform stress with small relative error for all tested barrier stiffness κ values.

Table 1: GMCP contact patch test results.

κ	σ_{zz} max rel. err.	σ_{spur}
10^4	8.00×10^{-4}	6.00×10^{-3}
10^6	4.66×10^{-5}	2.48×10^{-4}
10^8	4.66×10^{-5}	2.48×10^{-4}

Hertzian Contact. The Hertzian contact test evaluates how accurately the method recovers the normal pressure distribution in smooth curved contact at finite resolution [Laursen 2003]. We use a quarter-symmetry model in which a deformable one-eighth hemisphere of radius $R = 0.05$ indents a deformable elastic block represented by a quarter-cylinder sector of outer radius 0.12 and height 0.06. The block has 10,488 vertices and 55,921 tetrahedra, and the hemisphere has 11,838 vertices and 61,843 tetrahedra. The minimum initial gap between the block and the hemisphere is 5×10^{-5}. Both bodies use the same linear-elastic material, with $E = 2.1 \times 10^{11}$ and $\nu = 0.3$. The block bottom is fixed with $u_z = 0$. On both bodies, the two symmetry planes $x = 0$ and $y = 0$ impose $u_x = 0$ and $u_y = 0$, respectively. On the hemisphere, a uniform pressure $Q = 10^7$ is applied to the top surface along $-z$ and ramped over 10 load steps. The simulation is quasi-static and frictionless, with GMCP barrier support radius 10^{-5}.

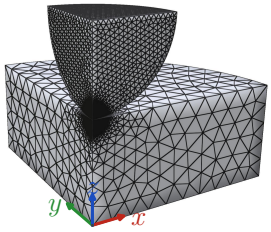

Figure 6: Hertzian contact.

The analytical contact pressure is given by the Hertz solution [Johnson 1987],

$$p(r) = \frac{3QR^2}{2\alpha^2}\sqrt{1-\frac{r^2}{\alpha^2}}, \qquad \alpha = \left(\frac{3q\pi R^3}{4E^*}\right)^{1/3},$$

where r is the radial distance from the contact center and $E^* = E/[2(1-\nu^2)]$ is the effective Young's modulus. Fig. 7 compares this analytical profile with the numerical GMCP result, which shows close agreement. The small oscillations of a few numerical points are expected to decrease with mesh refinement or higher-order elements.

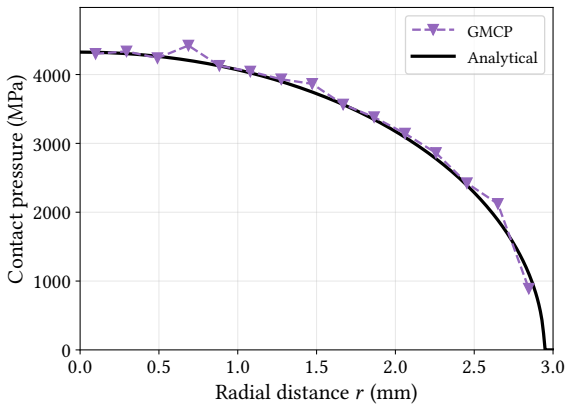

Figure 7: Hertz contact of GMCP compared with the analytical solution.

5.2 Qualitative Tests for IsaacIPC

Unless otherwise specified in this subsection, rigid bodies are simulated using affine body dynamics [Lan et al. 2022a], while deformable bodies are simulated using linear tetrahedral finite elements. This subsection focuses on qualitative demonstrations of the capabilities of the IsaacIPC framework. The contact model can be GMCP or another suitable model, and detailed simulation settings are omitted.

Legged locomotion. The Unitree Go2 quadruped robot [Unitree Robotics 2026], shown in the right panel of Fig. 1, is modeled as a full-body articulated system with rigid links and joint constraints in libuipc. All components are represented as rigid bodies except for the four black foot pads, which are modeled using FEM. For simulation efficiency, the simulation mesh is coarser than the rendering mesh; nevertheless, high-quality visual rendering is preserved through the dual-mesh mapper (Fig. 3).

To ensure that the qualitative behavior of the system is governed by transparent and interpretable kinematic commands, we use a manually designed open-loop trot controller as the motion input. The controller generates an alternating diagonal-leg trot gait and plans D-shaped foot trajectories in the sagittal plane. Fig. 9 visualizes the distribution of contact forces between the feet and the ground. The same setup also supports multi-environment parallel execution with domain randomization over foot elastic modulus, friction coefficient, and controller parameters, enabled by the subscene-based IsaacIPC architecture (see Fig. 8). Single-environment reset further enables selective reset, providing a practical option for rigid-deformable coupled reinforcement learning environments.

Dexterous in-hand manipulation. The Sharpa Wave dexterous hand [Sharpa Robotics 2026], shown in the middle panel of Fig. 1, is demonstrated on an in-hand ball-rotation task. As in the legged

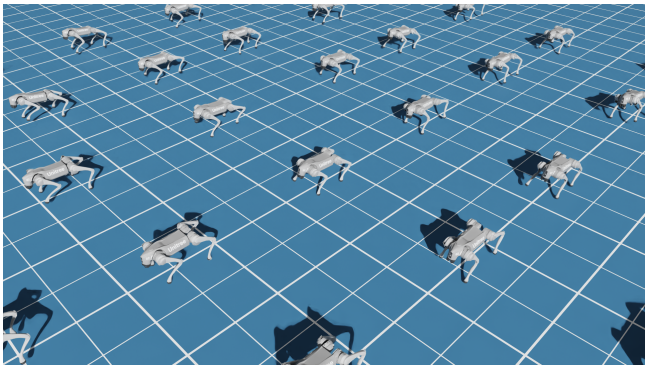

Figure 8: Parallel multi-environment simulation of the legged locomotion test.

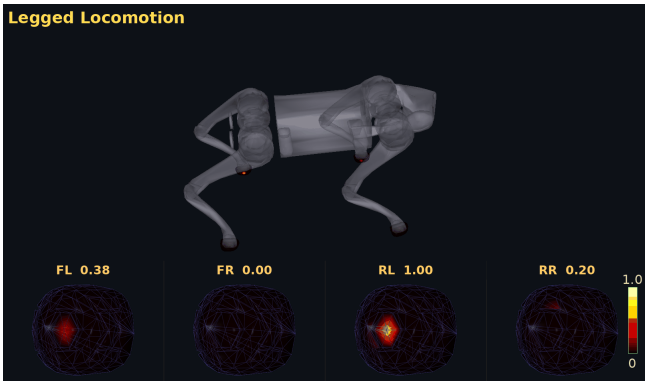

Figure 9: Contact force distribution on the four foot pads of the legged robot.

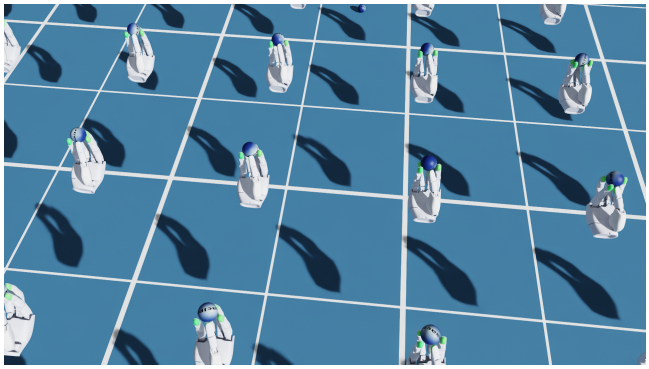

Figure 10: Parallel multi-environment simulation of dexterous hand manipulation test.

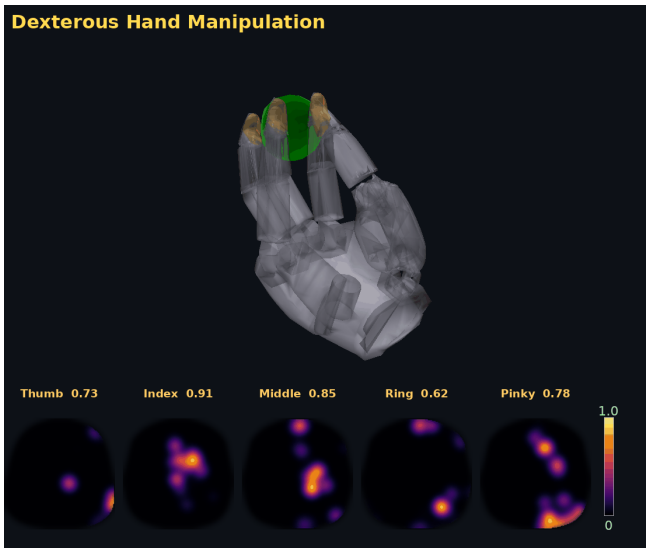

Figure 11: Contact force distribution on the five fingertips of the dexterous hand.

locomotion example, the rigid links and joint constraints are handled by libuipc, while the five green elastomer fingertips and the manipulated ball are modeled with FEM. The target trajectories for the hand's 22 actuated degrees of freedom are obtained from rollouts of a trained IsaacLab in-hand manipulation policy.

Fig. 11 visualizes the fingertip contact force distribution, and Fig. 10 shows the corresponding domain randomized multi environment simulation. This example illustrates how IsaacIPC can support high-quality data collection for dexterous in-hand manipulation.

UMI gripper manipulation. The Universal Manipulation Interface (UMI) [Chi et al. 2024] facilitates large-scale manipulation data collection for robot policy training. We model the UMI gripper in IsaacIPC. In this test, the Franka robotic arm inverse kinematics is solved by IsaacSim, and the UMI gripper is driven one-way by the arm. Fig. 12 shows a fisheye view of rigid-cube pick-and-place manipulation, with the scene background sourced from LeHome [Li et al. 2026]. To expose richer contact details on the inner gripper surfaces, where piezoresistive tactile sensors are commonly mounted, we additionally show a softer-cylinder grasping example in Fig. 13. This setup can also be used to evaluate robot learning policies.

6 Conclusion

We presented IsaacIPC, a robotic simulation framework that connects libuipc with IsaacSim/Lab. Through a dual-mesh mapper, IsaacIPC computes contact and deformation on simulation meshes while preserving the visual meshes, material bindings, USD / Fabric representation, and parallel environment infrastructure used by Omniverse based rendering and robot simulation. This design is not tied to a single contact model and can be extended to other physics backends.

We also introduced geometric mortar contact potential (GMCP), a mortar-based contact formulation for tactile simulation. GMCP constructs a barrier potential from contact samples on the tactile surface, providing a direct discretization of contact pressure on the sensor side. The contact patch and Hertzian contact tests validate its behavior on normal contact-pressure transfer, and the robotic examples demonstrate how IsaacIPC can be used in rigid–deformable contact-rich systems.

Figure 12: Fisheye view of UMI pick-and-place manipulation with a rigid cube.

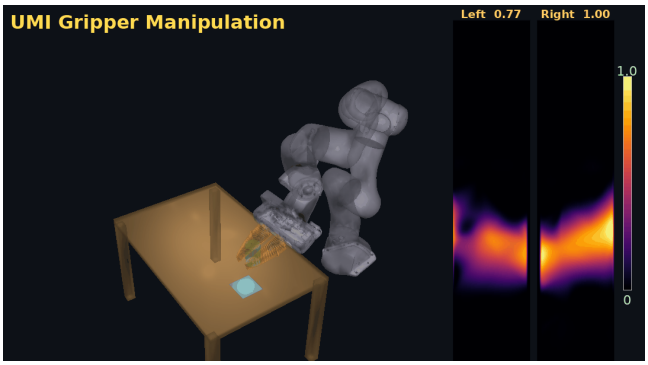

Figure 13: Contact force distribution on the inner flat surfaces of the UMI gripper.

Several limitations remain. High-fidelity robotic simulation always have to balance accuracy, robustness, and efficiency, especially when scaling rigid–deformable contact simulation to real-time or near-real-time reinforcement-learning settings. The current GMCP formulation mainly focuses on normal contact pressure; tangential traction, friction, stick–slip transitions, and shear deformation are important for tactile sensing and should be considered. Mortar formulations also depend on reliable projection, clipping, and quadrature on the contact interface, and can become more involved near sharp features, rapidly changing contact topology, or highly nonmatching meshes. Exploring alternative pressure-transfer schemes, adaptive quadrature, or higher-order contact representations may further improve accuracy, robustness, and efficiency.

Future work will also broaden IsaacIPC beyond the current rigid–deformable examples. Promising directions include coupling GPU-accelerated IPC with additional physical effects such as fluids, fracture, and other multiphysics phenomena; integrating asset generation and scene reconstruction pipelines; and using large-scale simulation to generate tactile and force-feedback data for pretraining and post-training embodied foundation models.

References

Steve Capell, Seth Green, Brian Curless, Tom Duchamp, and Zoran Popović. 2002. Interactive Skeleton-Driven Dynamic Deformations. *ACM Transactions on Graphics* 21, 3 (2002), 586–593. doi:10.1145/566654.566622

Baijun Chen, Weijie Wan, Tianxing Chen, Xianda Guo, Congsheng Xu, Yuanyang Qi, Haojie Zhang, Longyan Wu, Tianling Xu, Zixuan Li, Yizhe Wu, Rui Li, Xiaokang Yang, Ping Luo, Wei Sui, and Yao Mu. 2026. UniVTAC: A Unified Simulation Platform for Visuo-Tactile Manipulation Data Generation, Learning, and Benchmarking. arXiv:2602.10093 [cs.RO]

Yunuo Chen, Minchen Li, Lei Lan, Hao Su, Yin Yang, and Chenfanfu Jiang. 2022. A Unified Newton Barrier Method for Multibody Dynamics. *ACM Transactions on Graphics* 41, 4, Article 66 (July 2022), 14 pages. doi:10.1145/3528223.3530076

Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. 2024. Universal Manipulation Interface: In-the-Wild Robot Teaching Without In-the-Wild Robots. *arXiv preprint arXiv:2402.10329* (2024).

Wenxin Du, Wenqiang Xu, Jieji Ren, Zhenjun Yu, and Cewu Lu. 2024a. TacIPC: Intersection- and Inversion-free FEM-based Elastomer Simulation for Optical Tactile Sensors. *IEEE Robotics and Automation Letters* (2024).

Wenxin Du, Siqiong Yao, Xinlei Wang, Yuhang Xu, Wenqiang Xu, and Cewu Lu. 2024b. Intersection-free Robot Manipulation with Soft-Rigid Coupled Incremental Potential Contact. *IEEE Robotics and Automation Letters* (2024).

Wenxin Du, Chang Yu, Siyu Ma, Ying Jiang, Zeshun Zong, Yin Yang, Joe Masterjohn, Alejandro Castro, Xuchen Han, and Chenfanfu Jiang. 2024c. Embedded IPC: Fast and Intersection-free Simulation in Reduced Subspace for Robot Manipulation. arXiv:2409.16385 [cs.RO]

P. Farah, W. A. Wall, and A. Popp. 2018. A mortar finite element approach for point, line, and surface contact. *Internat. J. Numer. Methods Engrg.* 114, 3 (2018), 255–291. doi:10.1002/nme.5743

Zachary Ferguson, Pranav Jain, Denis Zorin, Teseo Schneider, and Daniele Panozzo. 2023a. High-Order Incremental Potential Contact for Elastodynamic Simulation on Curved Meshes. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Proceedings (SIGGRAPH '23)*. Los Angeles, CA, USA. doi:10.1145/3588432.3591488

Zachary Ferguson, Minchen Li, Teseo Schneider, Francisca Gil-Ureta, Timothy Langlois, Chenfanfu Jiang, Denis Zorin, Danny M. Kaufman, and Daniele Panozzo. 2021. Intersection-free Rigid Body Dynamics. *ACM Transactions on Graphics* 40, 4, Article 183 (Aug. 2021), 16 pages. doi:10.1145/3450626.3459802

Zachary Ferguson, Teseo Schneider, Danny M. Kaufman, and Daniele Panozzo. 2023b. In-Timestep Remeshing for Contacting Elastodynamics. *ACM Transactions on Graphics* 42, 4, Article 145 (Aug. 2023), 15 pages. doi:10.1145/3592428

Genesis Authors. 2024. Genesis: A Generative and Universal Physics Engine for Robotics and Beyond. https://github.com/Genesis-Embodied-AI/Genesis.

Dewen Guo, Minchen Li, Yin Yang, Sheng Li, and Guoping Wang. 2024. Barrier-Augmented Lagrangian for GPU-based Elastodynamic Contact. *ACM Transactions on Graphics* 43, 6, Article 225 (Nov. 2024), 17 pages. doi:10.1145/3687988

Kemeng Huang, Floyd M. Chitalu, Huancheng Lin, and Taku Komura. 2024. GIPC: Fast and Stable Gauss-Newton Optimization of IPC Barrier Energy. *ACM Transactions on Graphics* 43, 2 (Jan. 2024), 18 pages. doi:10.1145/3643028

Kemeng Huang, Xinyu Lu, Huancheng Lin, Taku Komura, and Minchen Li. 2025a. StiffGIPC: Advancing GPU IPC for Stiff Affine-Deformable Simulation. *ACM Transactions on Graphics* (2025).

Zizhou Huang, Maxwell Paik, Zachary Ferguson, Daniele Panozzo, and Denis Zorin. 2025b. Geometric Contact Potential. *ACM Transactions on Graphics* 44, 4 (Aug. 2025), 24 pages. doi:10.1145/3731142

Kenneth Langstreth Johnson. 1987. *Contact mechanics*. Cambridge university press.

Chung Min Kim, Michael Danielczuk, Isabella Huang, and Ken Goldberg. 2022. IPC-GraspSim: Reducing the Sim2Real Gap for Parallel-Jaw Grasping with the Incremental Potential Contact Model. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. 6180–6187.

Lei Lan, Danny M. Kaufman, Minchen Li, Chenfanfu Jiang, and Yin Yang. 2022a. Affine Body Dynamics: Fast, Stable and Intersection-free Simulation of Stiff Materials. *ACM Transactions on Graphics* 41, 4, Article 67 (July 2022), 14 pages. doi:10.1145/3528223.3530064

Lei Lan, Guanqun Ma, Yin Yang, Changxi Zheng, Minchen Li, and Chenfanfu Jiang. 2022b. Penetration-free Projective Dynamics on the GPU. *ACM Transactions on Graphics* 41, 4, Article 69 (July 2022), 16 pages. doi:10.1145/3528223.3530069

Tod A Laursen. 2003. *Computational contact and impact mechanics: fundamentals of modeling interfacial phenomena in nonlinear finite element analysis*. Springer Science & Business Media.

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 Transactions on Graphics* 39, 4, Article 49 (July 2020), 20 pages. doi:10.1145/3386569.3392425

Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M. Kaufman. 2023. Convergent Incremental Potential Contact. arXiv:2307.15908 [math.NA]

Yuyang Li, Wenxin Du, Chang Yu, Puhao Li, Zihang Zhao, Tengyu Liu, Chenfanfu Jiang, Yixin Zhu, and Siyuan Huang. 2025. Taccel: Scaling Up Vision-based Tactile Robotics via High-performance GPU Simulation. arXiv:2504.12908 [cs.RO]

Zeyi Li, Yushi Yang, Shawn Xie, Kyle Xu, Tianxing Chen, Yuran Wang, Zhenhao Shen, Yan Shen, Yue Chen, Wenjun Li, Yukun Zheng, Chaorui Zhang, Siyi Lin, Fei Teng, Hongjun Yang, Ming Chen, Steve Xie, and Ruihai Wu. 2026. LeHome: A Simulation Environment for Deformable Object Manipulation in Household Scenarios. arXiv:2604.22363 [cs.RO] https://arxiv.org/abs/2604.22363

Miles Macklin. 2022. Warp: A High-performance Python Framework for GPU Simulation and Graphics. https://github.com/NVIDIA/warp. NVIDIA GPU Technology Conference (GTC).

Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, Antonio Serrano-Munoz, Xinjie Yao, René Zurbrügg, Nikita Rudin, et al. 2025. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831* (2025).

Duc Huy Nguyen, Tim Schneider, Guillaume Duret, Alap Kshirsagar, Boris Belousov, and Jan Peters. 2024. TacEx: GelSight Tactile Simulation in Isaac Sim – Combining Soft-Body and Visuotactile Simulators. In *8th Conference on Robot Learning (CoRL 2024)*. Munich, Germany.

NVIDIA. 2026. NVIDIA Omniverse. https://www.nvidia.com/en-us/omniverse/. Accessed: 2026-05-22.

NVIDIA Corporation. 2025. PhysX SDK. https://developer.nvidia.com/physx-sdk.

Michael A. Puso and Tod A. Laursen. 2004. A Mortar Segment-to-Segment Contact Method for Large Deformation Solid Mechanics. *Computer Methods in Applied Mechanics and Engineering* 193, 6–8 (2004), 601–629. doi:10.1016/j.cma.2003.10.010

Sharpa Robotics. 2026. Sharpa Robotics GitHub Organization. https://github.com/sharpa-robotics. Accessed: 2026-05-20.

Xing Shen, Runyuan Cai, Mengxiao Bi, and Tangjie Lv. 2024. Preconditioned Nonlinear Conjugate Gradient Method for Real-time Interior-point Hyperelasticity. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Papers (SIGGRAPH '24)*. Denver, CO, USA, 11 pages. doi:10.1145/3641519.3657490

Hang Si. 2015. TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator. *ACM Trans. Math. Software* 41, 2 (2015), 11:1–11:36. doi:10.1145/2629697

spiriMirror. 2026. libuipc. https://github.com/spiriMirror/libuipc. Accessed: 2026-05-21.

Ivan E. Sutherland and Gary W. Hodgman. 1974. Reentrant polygon clipping. *Commun. ACM* 17, 1 (Jan. 1974), 32–42. doi:10.1145/360767.360802

Tencent Hunyuan3D Team. 2025. Hunyuan3D 2.5: Towards High-Fidelity 3D Assets Generation with Ultimate Details. arXiv:2506.16504 [cs.CV] https://arxiv.org/abs/2506.16504

Unitree Robotics. 2026. Unitree Robotics GitHub Organization. https://github.com/unitreerobotics. Accessed: 2026-05-20.

Barbara I. Wohlmuth. 2000. A Mortar Finite Element Method Using Dual Spaces for the Lagrange Multiplier. *SIAM J. Numer. Anal.* 38, 3 (2000), 989–1012. doi:10.1137/S0036142999350929

Peter Wriggers and Tod A Laursen. 2006. *Computational Contact Mechanics*. Vol. 2. Springer.

Ningyu Yan, Shuai Wang, Xing Shen, Hui Wang, Hanqing Wang, Yang Xiang, and Jiangmiao Pang. 2026. Tac2Real: Reliable and GPU Visuotactile Simulation for Online Reinforcement Learning and Zero-Shot Real-World Deployment. arXiv:2603.28475 [cs.RO]

Wenzhen Yuan, Rui Li, Mandayam A. Srinivasan, and Edward H. Adelson. 2015. Measurement of Shear and Slip with a GelSight Tactile Sensor. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 304–311.

Yu Zhang, Xing Shen, Kemeng Huang, Wei Chen, Yin Yang, Taku Komura, Tiantian Liu, and Xingang Pan. 2026. An Efficient Multilevel Preconditioned Nonlinear Conjugate Gradient Method for Incremental Potential Contact. arXiv:2604.19892 [cs.GR]

Juntian Zheng, Zhaofeng Luo, and Minchen Li. 2026. Robust and Efficient Penetration-Free Elastodynamics without Barriers. *ACM Transactions on Graphics* (2026). doi:10.1145/3811035