TacEx: GelSight Tactile Simulation in Isaac Sim – Combining Soft-Body and Visuotactile Simulators

Duc Huy Nguyen[1,3], Tim Schneider[1,2], Guillaume Duret[1,2],
Alap Kshirsagar[1], Boris Belousov[3], Jan Peters[1,3,4]

***Abstract*— Training robot policies in simulation is becoming increasingly popular; nevertheless, a precise, reliable, and easy-to-use tactile simulator for contact-rich manipulation tasks is still missing. To close this gap, we develop TacEx – a modular tactile simulation framework. We embed a state-of-the-art soft-body simulator for contacts named GIPC and vision-based tactile simulators Taxim and FOTS into Isaac Sim to achieve robust and plausible simulation of the visuotactile sensor GelSight Mini. We implement several Isaac Lab environments for Reinforcement Learning (RL) leveraging our TacEx simulation, including object pushing, lifting, and pole balancing. We validate that the simulation is stable and that the high-dimensional observations, such as the gel deformation and the RGB images from the GelSight camera, can be used for training. The code, videos, and additional results will be released online `https://sites.google.com/view/tacex`.**

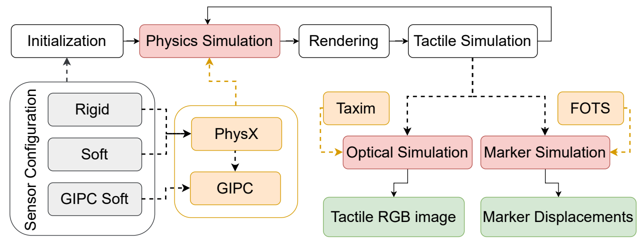

Fig. 1. **Overview of the TacEx Tactile Simulation Pipeline**. First, the simulation is initialized according to a given Sensor Configuration. Then the physics are simulated using PhysX and GIPC, followed by the scene rendering. Finally, the tactile sensor is simulated using the optical simulation (Taxim) and marker simulation (FOTS), yielding a tactile RGB image and a marker displacements field. After this, the physics are simulated again and the process repeats.

I. INTRODUCTION

Tactile sensing plays an important role for human perception of touch [1] and for advanced manipulation tasks in robotics [2]–[4]. Contact properties such as contact geometry, object stiffness, and surface texture can be estimated using tactile sensors [5]. Furthremore, slip detection [6], hardness estimation [7], and grasping of soft objects [8] are facilitated by the sense of touch. However, finger coordination based on tactile feedback is a complex control problem with high-dimensional observation space, therefore several Deep RL approaches have been explored [9], [10]. A crucial bottleneck for applying RL to tactile-rich manipulation tasks is the lack of stable and reliable contact simulation that includes soft-body interaction and tactile sensing. Although a number of simulators have appeared recently that aim to remedy this issue [10]–[14], each simulator uses a different physics engine, simulates a different tactile sensor, different robot, and runs in a different robotics simulator altogether – making comparison and interoperability challenging. To address these issues, we develop TacEx – a novel tactile simulation framework embedded in NVIDIA's Isaac Sim [15] and Isaac Lab [16], [17] that is modular, extensible, and based on the latest advancements in tactile simulation. We additionally integrate GIPC [18] for GPU-accelerated and inversion-free simulation of soft-body contacts. By leveraging Isaac Sim, we gain access to powerful features, such as photorealistic rendering, ROS support, and GPU-accelerated physics simulation, and by integrating TacEx into Isaac Lab – an extensible RL framework built on top of Isaac Sim – we enable support for teleoperation, GPU-parallelized training, and various RL libraries.

[1]TU Darmstadt [2]École centrale de Lyon [3]DFKI [4]Hessian.AI
This research was supported by the Hessian Ministry of Higher Education, Science, and the Arths through the "The Adaptive Mind", "Third Wave of AI", and "DFKI" grants, l'Agence Nationale de Recherche (ANR) and the German Federal Ministry of Education and Research (BMBF) through the project Aristotle (ANR-21-FAI1-0009-01), and the Lichtenberg II at TU Darmstadt funded by the BMBF and the State of Hesse.

II. TacEx Simulation Framework

In this section, we present TacEx – our modular framework for tactile simulation (c.f., Fig. 1). We compare three different approaches for simulating the physical behavior of sensors and objects: i) PhysX to simulate the gelpad as a rigid body with compliant contact; ii) PhysX FEM-based soft body simulation for the gelpad; iii) GIPC [18] to simulate the gelpad as a soft body. Since PhysX is the built-in physics engine of Isaac Sim, baselines i) and ii) are straightforward to implement by directly setting asset properties; for iii), we modified the GIPC code and created Python bindings.

We integrate the GIPC simulation with Isaac Sim in the following manner. Isaac Sim is used for scene setup, robot simulation, and rendering. The gelpad is attached to the sensor case and moves kinematically in response to the robot's motion, which is handled by PhysX. The non-attached gelpad vertices are handled by GIPC: as the robot moves in Isaac Sim, the sensor case moves and the attachment points are recomputed, followed by a call to the GIPC solver that computes new positions for the remaining gelpad vertices and other GIPC-modeled objects. This enables the gelpad and objects to move, deform, and interact dynamically in Isaac Sim. For optical simulation, we use Taxim [11]; specifically, a GPU-accelerated implementation [19]. First, we generate height maps with cameras in Isaac Sim and smooth them with pyramid Gaussian kernels. Then we use

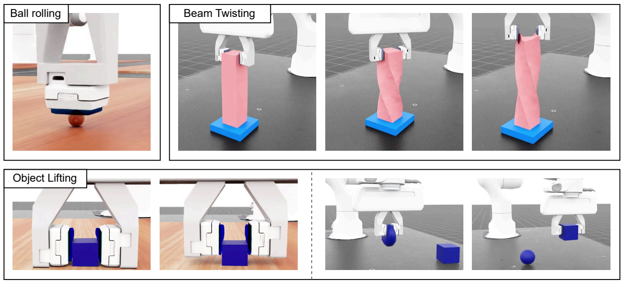

Fig. 2. **Simulation Showcases**. We do a ball rolling experiment for testing the simulation performance, and we further evaluate the capabalities of our GIPC simulation by twisting and stretching a soft body beam and test how well the gelpads can be used for lifting objects (see website for videos).

a polynomial lookup table to map the surface normals of the height maps to RGB values. As a final step, we attach shadows to the images. We further use the generated height maps to simulate the marker motion with FOTS [20]. For this, we compute the contact centers based on the height maps and extract the z rotation of the objects relative to the gelpads from Isaac Sim.

III. Demonstrations and Evaluation

We showcase the behavior, capabilities, and limitations of our framework in a series of experiments (visualized in Fig. 3). First, a ball rolling experiment demonstrates a contact-rich manipulation task with a single GelSight sensor. Second, object lifting with two soft gelpads showcases robust grasping capabilities. Third, the limits of GIPC simulation are tested in a challenging beam twisting environment. Subsequently, we implement three RL tasks: object pushing, object lifting, and pole balancing – to demonstrate how TacEx can be used within Isaac Lab for RL training.

We evaluate the simulation time of the optical simulation Taxim/FOTS in Table I. Results for PhysX are presented in Table II. The runtimes for soft-body GIPC simulation are shown in Table III. We measure the average simulation time per frame in ms for the physics simulation with GIPC during the ball rolling experiment without tactile simulation.

TABLE I

Tactile Simulation Speed. Average simulation time per frame in ms for optical (resolution of 480×640) and for marker simulation (10×10 markers) during contact in the ball rolling experiment with rigid body gelpads.

num_envs	height map gen	optical sim	marker sim
1	1.3718	5.9015	4.4863
2	0.8508	3.8886	2.8838
4	0.5988	3.0424	2.1184
8	0.4323	2.5773	1.7587
16	2.8827	5.7314	5.0450
18	3.5149	5.931	5.2343

IV. Conclusion and Future Work

We presented TacEx – a novel framework for simulating GelSight tactile sensors. The framework enables the usage of GelSight Mini sensors for RL. It is built on top of Isaac Sim and Isaac Lab, which gives the user access to a wealth of features non-existent in current tactile simulators. We designed the framework to be modular, extendable, and easy to use by integrating multiple different simulation approaches. The gelpad can either be simulated as a rigid body with compliant contact, or as a soft body. For the soft body simulation, one can use PhysX or our integration of GIPC. To simulate the sensor output, we create height maps with cameras in Isaac Sim and use the approach from Taxim [11] for the optical and the one from FOTS [20] for the marker simulation. We demonstrated framework features and simulation behavior with multiple examples. We also provide three environments for RL with tactile sensing.

Our tactile simulation, specifically the physics simulation approaches, lacks experiments that investigate whether they can be used for Sim2Real or not. Therefore, we aim to do more quantitative experiments for comparing different tactile simulation approaches, as well as Sim2Real experiments in future works. We also plan to extend our framework to include more RL environments and more tactile simulation approaches with the goal of creating a benchmarking platform for tactile simulators and algorithms for tactile sensing.

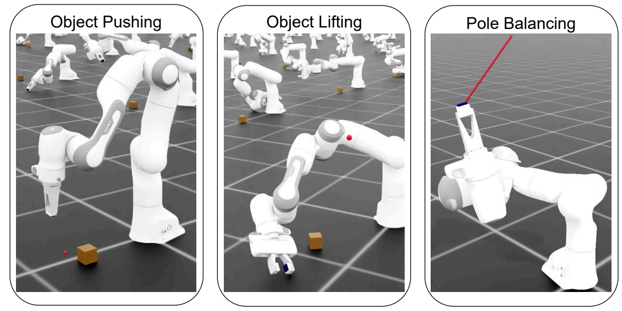

Fig. 3. **RL environments.** We implemented three environments in Isaac Lab and trained policies to validate that our framework can be used for Reinforcement Learning. In each environment, we used the marker displacements from our tactile simulation. For training the RL policies, we used PPO [21] as implemented in [22]. We have validated that the training pipeline works, and we are currently working towards obtaining successful policies that leverage tactile feedback.

TABLE II

PhysX Simulation Speed. Average simulation time per frame in ms for the physics simulation with PhysX during the ball rolling experiment. The soft body gelpad has a mesh resolution of 10 and uses 16 solver iterations.

num_envs	1	16	32	64	128
rigid	3.6930	0.2426	0.1286	0.0673	0.0361
soft	4.7069	0.4496	0.2718	0.1798	0.1267

TABLE III

GIPC Simulation Speed. Compared to the ball rolling experiments with PhysX, the ball here is a soft body. We use different mesh resolutions for the ball to measure the performance w.r.t the amount of vertices and tetrahedra.

num_vert	num_tetra	GIPC
1029	3717	24.95 ms
7900	40370	110.47 ms
12509	66563	221.61 ms

REFERENCES

[1] R. S. Dahiya, G. Metta, M. Valle, and G. Sandini, "Tactile sensing—from humans to humanoids," *IEEE transactions on robotics*, vol. 26, no. 1, pp. 1–20, 2009.

[2] Z. Yu, W. Xu, S. Yao, J. Ren, T. Tang, Y. Li, G. Gu, and C. Lu, "Precise Robotic Needle-Threading with Tactile Perception and Reinforcement Learning."

[3] A. Wilson, H. Jiang, W. Lian, and W. Yuan, "Cable Routing and Assembly using Tactile-driven Motion Primitives."

[4] G. Wang, X. Liu, Z. Liu, P. Huang, and Y. Yang, "Visual-Tactile Perception Based Control Strategy for Complex Robot Peg-in-Hole Process via Topological and Geometric Reasoning," vol. 9, no. 10, pp. 8410–8417.

[5] Q. Li, O. Kroemer, Z. Su, F. F. Veiga, M. Kaboli, and H. J. Ritter, "A review of tactile information: Perception and action through touch," vol. 36, no. 6, pp. 1619–1634, conference Name: IEEE Transactions on Robotics. [Online]. Available: https://ieeexplore.ieee.org/document/9136877/?arnumber=9136877

[6] W. Yuan, R. Li, M. A. Srinivasan, and E. H. Adelson, "Measurement of shear and slip with a GelSight tactile sensor," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 304–311.

[7] W. Yuan, C. Zhu, A. Owens, M. A. Srinivasan, and E. H. Adelson, "Shape-independent hardness estimation using deep learning and a GelSight tactile sensor," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 951–958. [Online]. Available: http://arxiv.org/abs/1704.03955

[8] Y. Han, K. Yu, R. Batra, N. Boyd, C. Mehta, T. Zhao, Y. She, S. Hutchinson, and Y. Zhao, "Learning generalizable vision-tactile robotic grasping strategy for deformable objects via transformer," pp. 1–13, conference Name: IEEE/ASME Transactions on Mechatronics. [Online]. Available: https://ieeexplore.ieee.org/document/10552075/?arnumber=10552075

[9] A. Church, J. Lloyd, R. Hadsell, and N. F. Lepora, "Deep Reinforcement Learning for Tactile Robotics: Learning to Type on a Braille Keyboard," vol. 5, no. 4, pp. 6145–6152.

[10] W. Chen, J. Xu, F. Xiang, X. Yuan, H. Su, and R. Chen, "General-Purpose Sim2Real Protocol for Learning Contact-Rich Manipulation With Marker-Based Visuotactile Sensors," vol. 40, pp. 1509–1526.

[11] Z. Si and W. Yuan, "Taxim: An Example-based Simulation Model for GelSight Tactile Sensors."

[12] Y. Lin, J. Lloyd, A. Church, and N. F. Lepora, "Tactile Gym 2.0: Sim-to-real Deep Reinforcement Learning for Comparing Low-cost High-Resolution Robot Touch."

[13] Z. Si, G. Zhang, Q. Ben, B. Romero, Z. Xian, C. Liu, and C. Gan, "DIFFTACTILE: A Physics-based Differentiable Tactile Simulator for Contact-rich Robotic Manipulation."

[14] J. Xu, S. Kim, T. Chen, A. R. Garcia, P. Agrawal, W. Matusik, and S. Sueda, "Efficient tactile simulation with differentiability for robotic manipulation," in *6th Annual Conference on Robot Learning*.

[15] NVIDIA, "Nvidia isaac sim." [Online]. Available: https://developer.nvidia.com/isaac/sim

[16] ——, "Isaac lab." [Online]. Available: https://isaac-sim.github.io/IsaacLab/

[17] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023.

[18] K. Huang, F. M. Chitalu, H. Lin, and T. Komura, "Gipc: Fast and stable gauss-newton optimization of ipc barrier energy," *ACM Trans. Graph.*, vol. 43, no. 2, 3 2024. [Online]. Available: https://doi.org/10.1145/3643028

[19] T. Schneider, "Taxim-gpu." [Online]. Available: https://git.ias.informatik.tu-darmstadt.de/tactile-sensing/taxim-gpu

[20] Y. Zhao, K. Qian, B. Duan, and S. Luo, "FOTS: A Fast Optical Tactile Simulator for Sim2Real Learning of Tactile-motor Robot Manipulation Skills."

[21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms." [Online]. Available: https://arxiv.org/abs/1707.06347

[22] L. Schneider, J. Frey, T. Miki, and M. Hutter, "Learning risk-aware quadrupedal locomotion using distributional reinforcement learning," 2023.