

On the Development of GPU-Accelerated Hand Simulation for High-Fidelity Manipulation Data

Harim Ji
School of Mechanical Engineering
Seoul National University
Email: jiharim0911@snu.ac.kr

Hyelim Choi
School of Mechanical Engineering
Seoul National University
Email: helmchoi@snu.ac.kr

Dongjun Lee[†]
School of Mechanical Engineering
Seoul National University
Email: djlee@snu.ac.kr

Abstract—Physics simulation is one of the key enablers for collecting high-quality demonstration data for learning contact-rich dexterous manipulation skills. In this paper, we present a GPU-accelerated simulation framework for teleoperated dexterous manipulation data collection. The framework enables a human operator to interact with virtual manipulation scenes in real time through multiple hand embodiments, including robotic hands and anthropomorphic avatar hands, driven by various hand-tracking devices. To support contact-rich interaction with complex geometries, the framework combines GPU-accelerated signed-distance-field-based collision detection for complex object meshes with a GPU-accelerated dynamics solver that handles the nonlinear complementarity problem induced by Signorini–Coulomb contact constraints. This integration enables interactive teleoperation in contact-intensive scenes involving complex objects and high-dimensional hand-object dynamics, while providing dense simulation states and contact information that are difficult to obtain from physical data collection alone.

I. INTRODUCTION

High-fidelity data is becoming increasingly important for dexterous robotic manipulation as robotic systems move beyond simple pick-and-place tasks toward more complex tasks involving contact-rich behaviors such as tool use, in-hand manipulation, and tight-tolerance assembly. Learning such behaviors requires successful interaction episodes that capture rich contact dynamics, object motion, and hand-object coordination. However, acquiring these episodes in the real world is challenging: dexterous manipulation involves high-dimensional actuation, intermittent multi-contact interactions, and tightly coupled control and planning, which together create a chicken-and-egg problem. Reliable policies require high-quality data, but collecting such data at scale often requires policies that are already competent. One way to break this cycle is to leverage human intelligence through learning from demonstration, for example, by collecting teleoperated manipulation trajectories. While effective, this approach is limited by the cost and complexity of physical teleoperation systems, as well as by the limited throughput of real-world data collection. In addition, physical experiments often provide only partial observations of the underlying interaction state; contact forces, object poses, pressure distributions, and other quantities relevant to manipulation are difficult to measure accurately and densely. In this context, simulation offers a promising route toward scalable generation of dexterous manipulation data.



Fig. 1: Overview of ‘Leap hand assembly’ scenario consisting of four bolt–nut assembly tasks. The left half shows meshes used as colliders, while the right half shows meshes for rendering.

In this paper, we present a GPU-accelerated simulation framework for teleoperated dexterous manipulation data collection. The framework enables a human operator to interact with a virtual manipulation scene in real time through multiple hand embodiments, including robotic hands and anthropomorphic avatar hands, driven by various hand-tracking devices [22, 17, 8]. To support contact-rich manipulation with complex geometries, the framework combines TE-SDF [14], a GPU-accelerated signed-distance field (SDF) based collision detection between complex object meshes, with GPU-SubADMM [13], a GPU-accelerated dynamics solver that handles the nonlinear complementarity problem (NCP) derived from the exact Signorini–Coulomb contact constraints formulation. This combination enables interactive teleoperation in contact-intensive scenes involving many complex objects and high-dimensional hand-object dynamics, while providing dense simulation states and contact information. Using this framework, we demonstrate challenging dexterous manipulation scenes, such as the ‘Leap hand assembly,’ as shown in Fig. 1.

II. SIMULATION FRAMEWORK

In this section, we explain the core components and overall simulation framework.

A. Collision Detection for Complex Contact

Accurate collision detection is a crucial component of contact-rich simulation, as it provides the contact constraints or penalty forces that are subsequently processed by the dynamics solver. Inaccurate contact features, such as erroneous normals, penetration depths, or contact locations, can lead to non-physical behavior, including jittering, excessive penetration, or numerical instability, even when the dynamics solver itself is well-behaved. Direct triangle-mesh-based collision detection can represent complex geometry, but it is often difficult to use robustly as it relies only on surface intersection information. As a result, additional procedures are required to recover stable contact features, such as continuous collision handling methods and penetration-free dynamics solvers [7, 18], which are too expensive for interactive simulation.

Signed distance fields (SDFs) provide a more suitable representation for contact simulation because they encode volumetric proximity information. Through spatial queries, an SDF directly provides distance-to-surface information and surface normals, which can be used to construct contact constraints more robustly. However, conventional SDF representations involve important trade-offs. Dense voxel grids are simple and fast to query [23, 3, 28, 12], but their memory cost grows cubically with resolution, making it expensive to represent small geometric features and tight clearances. Sparse grid structures reduce memory usage but introduce hierarchical tree traversal during SDF queries [11, 16], which can be costly and difficult to parallelize efficiently on GPUs. Neural SDF representations offer another memory-efficient alternative [6, 4, 24, 29], but their repeated network evaluations introduce significant query overhead, making them less suitable for large-scale contact detection in interactive simulation.

To address this issue, in addition to standard separating-axis-based collision detection between convex polytopes, we adopt a new representation for SDFs, TE-SDF [14]. TE-SDF is a combination of the tetrahedralized spatial discretization of the occupied volume of the underlying geometry and localized distance functions encoded per tetrahedron. Tetrahedralized mesh (tetramesh) naturally provides an adaptive spatial decomposition; dense tetrahedra are located in geometrically complex regions, while coarse tetrahedra are located in simple regions [26]. This provides a memory-efficient representation of space compared to dense voxel grids. For each tetrahedron t , a localized distance function is represented by encoding a subset of the surface triangles E_t , which are the *possibly closest faces* satisfying the following:

$$d(x, \mathcal{F}) = d(x, E_t)$$

where $d(\cdot, \cdot) : \mathbb{R}^3 \times \mathcal{P}(\mathbb{R}^3) \rightarrow \mathbb{R}$ is a distance function from a point to a subset of $\mathbb{R}^3$, $\mathcal{F}$ is the surface of the geometry, and E_t is the subset of the surface triangles stored for tetrahedron

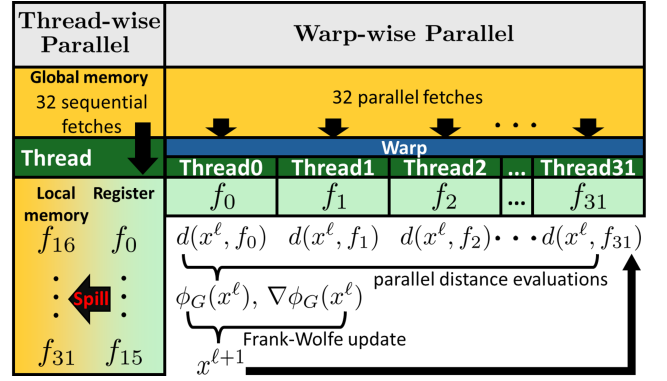


Fig. 2: Comparison between thread-wise and warp-wise parallel Face-TES collision detection scheme with $n(E_t) = 32$. Thread-wise parallelization spills half of the fetched faces from global to local memory, resulting in performance loss. Warp-wise parallelization has no such spill, as it contains all necessary data in registers. This scheme also enables parallelized face set fetching and SDF evaluation.

t . This reduces the SDF evaluation within a tetrahedron to a few point-to-triangle distance evaluations.

With this new representation, GPU-accelerated collision detection can be done as follows. First, every overlapping triangle-tetrahedron region is obtained with GPU-accelerated broadphase [15] and a parallelized intersection calculation between the triangle and the tetrahedron. Then the following optimization problem is solved for every pair using the Frank-Wolfe method, which is a similar approach done in [21]:

$$\min_{x \in P} \phi(x) \quad (1)$$

where P is the intersecting triangle-tetrahedron region. Here, a warp-level parallel approach [10] is used to solve these problems, as thread-level parallelism can cause register spilling into local memory [9] as shown in Fig 2.

B. GPU-Accelerated Contact Dynamics

After collision detection generates contact constraints, the resulting dynamics problem must be solved efficiently and robustly. Dexterous manipulation involves complicated dynamics involving contact and coupled motion between articulated hands and manipulated objects. These effects naturally lead to non-smooth contact dynamics governed by unilateral Signorini constraints and Coulomb friction. The corresponding time-stepping problem can be formulated as a nonlinear complementarity problem (NCP).

However, solving this problem at interactive rates remains challenging. Many practical simulators improve robustness and speed by modifying the original contact problem. For example, MuJoCo [27] and Drake [5] use convexified or relaxed contact constraints to formulate the problem as a cone complementary problem (CCP). These formulations are effective in many robotic simulation settings, but the relaxation introduces artificial compliance, contact softening, or gliding during slipping.

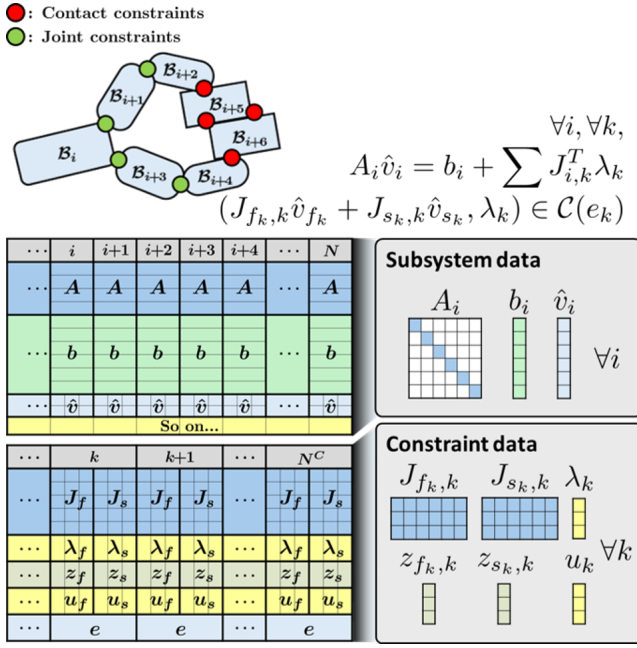


Fig. 3: GPU-accelerated SubADMM overview under a scenario with both contact and joint constraints. Each subsystem and contact constraint is designated a memory region with the same layout pattern. This structure is the same for joint constraints, which are not shown in the figure.

Another common approach is to use projection-based or position-level constraint solvers, such as those related to extended position-based dynamics (XPBD) [19]. These methods are computationally efficient and well suited to interactive applications, but their Gauss–Seidel-style sequential updates can exhibit limited scalability and convergence issues in tightly coupled contact systems. This limitation becomes especially important in scenes with large mass ratios, such as light fingertips interacting with heavier objects, where inaccurate impulse propagation can lead to visible artifacts even after many solver iterations [20].

To address these limitations, our framework uses a GPU-accelerated subsystem-based ADMM solver for non-smooth contact dynamics [13]. The solver decomposes the full system dynamics into body-level subsystem dynamics:

$$A_i \hat{v}_i = b_i + \sum_{k \in \mathcal{I}_i} J_{i,k}^T \lambda_k \quad i = 1, \dots, N \quad (2)$$

where i denotes the subsystem index, $\hat{v}_i$ denotes the representative velocity, $A_i \in \mathbb{R}^{6 \times 6}$, $b_i \in \mathbb{R}^6$ are the mass matrix and the dynamics vector including the momentum, Coriolis force and external forces, k denotes the constraint index, $\mathcal{I}_i$ denotes the set of constraint indices the subsystem is under, $J_{i,k} \in \mathbb{R}^{n_{c,k} \times 6}$, $\lambda_k \in \mathbb{R}^{n_{c,k}}$ are the time-scaled constraint Jacobian and the constraint impulse where $n_{c,k}$ denotes the constraint dimension. Constraints can be formulated as follows:

$$(J_{f_k,k} \hat{v}_{f_k} + J_{s_k,k} \hat{v}_{s_k}, \lambda_k) \in \mathcal{C}(e_k) \quad k = 1 \dots N_c \quad (3)$$

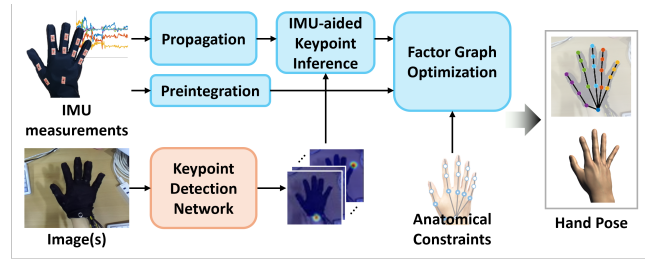


Fig. 4: Overall structure for markerless VIST. A keypoint detection network based on a lightweight vision transformer (ViT) processes visual data into keypoints. IMU measurement is fused via a factor graph optimization framework, refining the pose.

where f_k, s_k are the indices of each first and second subsystems which are involved in the k th constraint, and $\mathcal{C}(e_k) \in \mathbb{R}^{2n_k}$ is the set defined by the constraint error $e_k \in \mathbb{R}$ and the constraint type. The set $\mathcal{C}(e)$ for each constraint is defined for three constraint types: soft, hard, and contact constraints. Especially, we formulate the contact constraints using the Signorini-Coulomb condition (SCC) by expressing the set $\mathcal{C}(e)$ as follows:

$$\mathcal{C}(e) = \left\{ (x, \lambda) \left| \begin{array}{l} 0 \leq \lambda_n \perp x_n + \Phi(e) \geq 0, \\ 0 \leq \delta \perp \mu \lambda_n - \|\lambda_t\|_2 \geq 0, \\ \delta \lambda_t + \mu \lambda_n x_t = 0, \\ \delta \in \mathbb{R} \end{array} \right. \right\} \quad (4)$$

where $\Phi(e)$ is the error fixed by Baumgarte stabilization [1], the subscripts n and t denote the row corresponding to each contact normal and contact tangent, and $\mu \in \mathbb{R}$ denotes the friction coefficient. The resulting problem is reformulated as an augmented Lagrangian optimization problem and solved using ADMM. Each ADMM iteration decomposes the computation into body-level dynamics updates and constraint-level contact solves, both of which can be executed in parallel. GPU-SubADMM exploits this structure by representing bodies and contact constraints with fixed-size computational units, thereby balancing the workload across GPU threads and enabling a memory layout optimized for parallel execution.

C. Hand-Tracking System

Robust and accurate hand-tracking input is crucial for tele-operated dexterous manipulation data collection, since tracking errors directly affect the quality of hand-object interaction in simulation. Our framework supports a vision-based tracking system from Meta Quest [22], which provides a convenient and low-cost interface for real-time teleoperation. However, purely vision-based hand tracking can suffer from frequent occlusion during dexterous manipulation, especially when the hand is close to objects, tools, or other fingers. Such occlusions can cause erroneous pose estimates and may lead to task failure in fine manipulation scenarios.

To improve robustness under occlusion, we also integrate a visual-inertial skeletal tracking system (VIST) [17, 8]. VIST fuses visual observations with inertial measurements,

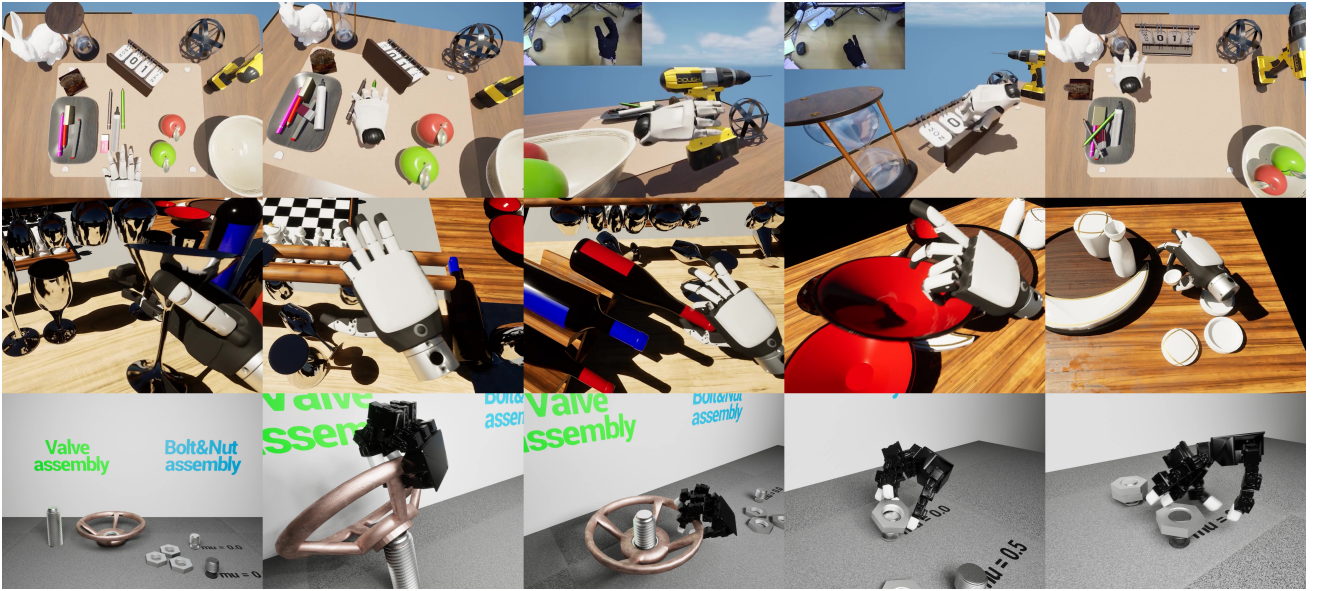
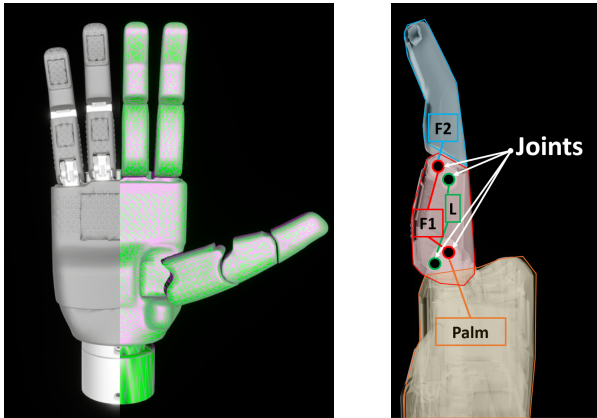


Fig. 5: Demonstration overviews. (Top) Tidy table. (Middle) Inspire hand dining. (Bottom) Leap hand assembly.



(a) Visual and collider mesh. (b) Synergy mechanism.

Fig. 6: Inspire hand asset for the ‘Inspire hand dining’ demo.

enabling more reliable hand-pose estimation even under severe occlusions. By supporting both pure vision-based tracking and visual-inertial tracking, the framework can accommodate different teleoperation setups while providing stable hand-motion input for contact-rich manipulation in simulation.

III. DEMONSTRATIONS

In this section, we evaluate the proposed framework through a set of teleoperation demonstrations as shown in Fig. 5.

A. Tidy table

In this scenario, an anthropomorphic avatar hand model is used, controlled by the markerless VIST hand-tracking system [8], while the object’s collider geometry is represented by convex polytopes. The user tidies up a messy table with various objects. We successfully simulate picking challenging

objects such as a thin pencil and nonprehensile manipulation such as standing a calendar.

B. Inspire hand dining

In this scenario, the Inspire robotic hand model [2] is used, controlled by the marker VIST hand-tracking system [17]. Instead of using convex polytopes, we use a TE-SDF for the underlying trimesh that faithfully represents the smooth surface of the fingers and the palm of the robot hand, as shown in Fig. 6a, as well as other objects with complex geometries (wine glasses, dishes, etc.). We include the synergy mechanism of the fingers for the hand using a 4-bar linkage system as shown in Fig. 6b. Collision detection between these complex geometries and dynamics solving with complex contact and joint constraints is executed at an interactive rate.

C. Leap hand bolting

In this scenario, the Leap hand model [25] is used, controlled by the Meta Quest hand-tracking system [22]. We successfully simulate challenging assembly tasks with complex geometries (valve, bolt, and nuts) within a single scene in real-time. Notice that the collision detection and dynamics solve are done accurately enough to handle challenging dynamics between the bolts and nuts.

IV. CONCLUSION

In this paper, we propose a new simulation framework for collecting dexterous manipulation data for challenging scenarios by combining an accurate and memory-efficient collision detection module, TE-SDF, and a GPU-accelerated dynamics solver, GPU-SubADMM, which solves the NCP dynamics governed by Signorini–Coulomb constraints. We also integrated with various hand-tracking systems from low-cost Meta Quest hand-tracking to robust and accurate VIST. As future work, we plan to extend the framework with additional

sensor-data modules, including multi-view image generation and tactile sensing, to support richer supervision for downstream learning. Also, enhanced mapping between the tracked hand and the robot hand, and integration with haptic feedback with calculated contact forces are also promising extensions.

ACKNOWLEDGMENT

The authors are with the Department of Mechanical Engineering, Institute of Advanced Machines and Design (IAMD), and Institute of Engineering Research (IOER), Seoul National University, Seoul, Republic of Korea. This work was supported by the Korea Planning & Evaluation Institute of Industrial Technology (KEIT) grant funded by the Ministry of Trade, Industry and Resources (MOTIR) (No. RS-2024-00441872, No. RS-2024-00419641, No. RS-2025-25455303).

REFERENCES

- [1] Joachim Baumgarte. Stabilization of constraints and integrals of motion in dynamical systems. *Computer methods in applied mechanics and engineering*, 1(1):1–16, 1972.
- [2] Beijing Inspire Robots Technology Co., Ltd. RH56BFX: The dexterous hands RH56BFX series. <https://en.inspire-robots.com/product/rh56bfx>. Accessed: 2026-06-21.
- [3] J Bender, C Duriez, F Jailliet, and G Zachmann. Continuous collision detection between points and signed distance fields. In *Workshop on virtual reality interaction and physical simulation*, volume 8, 2014. URL <https://viterbi-web.usc.edu/~jbarbic/ccd/XuBarbicVRIPHYS2014.pdf>.
- [4] Hugo Bertiche, Meysam Madadi, and Sergio Escalera. Neural implicit surfaces for efficient and accurate collisions in physically based simulations, 2021. URL <https://arxiv.org/abs/2110.01614>. arXiv:2110.01614.
- [5] Alejandro M. Castro, Frank N. Permenter, and Xuchen Han. An unconstrained convex formulation of compliant contact. *Trans. Rob.*, 39(2):1301–1320, April 2023. ISSN 1552-3098. doi: 10.1109/TRO.2022.3209077. URL <https://doi.org/10.1109/TRO.2022.3209077>.
- [6] Rohan Chabra, Jan E. Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 608–625, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58526-6.
- [7] Anka He Chen, Ziheng Liu, Yin Yang, and Cem Yuksel. Vertex block descent. 43(4), July 2024. ISSN 0730-0301. doi: 10.1145/3658179. URL <https://doi.org/10.1145/3658179>.
- [8] Hyelim Choi, Hyunreal Park, Harim Ji, Somang Lee, Youngseon Lee, Yongseok Lee, and Dongjun Lee. Robust hand tracking from visual-inertial fusion. In *Proceedings of the 2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026.
- [9] Nvidia Corporation. Cuda c++ programming guide. <https://docs.nvidia.com/cuda/cuda-programming-guide/02-basics/writing-cuda-kernels.html>, 2026. Accessed: 2026-01-10.
- [10] Nvidia Corporation. Cuda c++ programming guide. <https://docs.nvidia.com/cuda/cuda-programming-guide/03-advanced/advanced-kernel-programming.html#advanced-kernels-hardware-implementation-simt-architecture>, 2026. Accessed: 2026-01-10.
- [11] Sarah F. Frisken, Ronald N. Perry, Alyn P. Rockwood, and Thouis R. Jones. Adaptively sampled distance fields: a general representation of shape for computer graphics. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, page 249–254, USA, 2000. ACM Press/Addison-Wesley Publishing Co. ISBN 1581132085. doi: 10.1145/344779.344899. URL <https://doi.org/10.1145/344779.344899>.
- [12] Eran Guendelman, Robert Bridson, and Ronald Fedkiw. Non-convex rigid bodies with stacking. *ACM Trans. Graph.*, 22(3): 871–878, July 2003. ISSN 0730-0301. doi: 10.1145/882262.882358. URL <https://doi.org/10.1145/882262.882358>.
- [13] Harim Ji, Hyunsu Kim, Jeongmin Lee, Somang Lee, Seoki An, Jinuk Heo, Youngseon Lee, Yongseok Lee, and Dongjun Lee. Gpu-accelerated subsystem-based admm for large-scale interactive simulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13868–13874, 2025. doi: 10.1109/ICRA55743.2025.11128665. URL <https://ieeexplore.ieee.org/document/11128665>.
- [14] Harim Ji, Yongseok Lee, and Dongjun Lee. TE-SDF: Tetra-encoded signed distance field for memory-efficient and accurate collision detection. In *Proceedings of Robotics: Science and Systems (RSS)*, 2026.
- [15] Tero Karras. Maximizing parallelism in the construction of bvhs, octrees, and k-d trees. In *Proceedings of the Fourth ACM SIGGRAPH / Eurographics Conference on High-Performance Graphics*, EGGH-HPG'12, page 33–37, Goslar, DEU, 2012. Eurographics Association. ISBN 9783905674415. URL <https://diglib.org/bitstream/handle/10.2312/EGGH.HPG12.033-037/033-037.pdf?sequence=1>.
- [16] Dan Koschier, Crispin Deul, Magnus Brand, and Jan Bender. An hp-adaptive discretization algorithm for signed distance field generation. *IEEE Transactions on Visualization and Computer Graphics*, 23(10):2208–2221, 2017. doi: 10.1109/TVCG.2017.2730202. URL <https://ieeexplore.ieee.org/abstract/document/7987773>.
- [17] Yongseok Lee, Wonkyung Do, Hanbyeol Yoon, Jinuk Heo, WonHa Lee, and Dongjun Lee. Visual-inertial hand motion tracking with robustness against occlusion, interference, and contact. *Science Robotics*, 6(58):eabe1315, 2021.
- [18] Minchen Li, Zachary Ferguson, Teso Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M. Kaufman. Incremental potential contact: intersection- and inversion-free, large-deformation dynamics. *ACM Trans. Graph.*, 39(4), August 2020. ISSN 0730-0301. doi: 10.1145/3386569.3392425. URL <https://doi.org/10.1145/3386569.3392425>.
- [19] Miles Macklin, Matthias Müller, and Nuttapong Chentanez. Xpbd: position-based simulation of compliant constrained dynamics. MIG '16, page 49–54, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450345927. doi: 10.1145/2994258.2994272. URL <https://doi.org/10.1145/2994258.2994272>.
- [20] Miles Macklin, Kenny Erleben, Matthias Müller, Nuttapong Chentanez, Stefan Jeschke, and Viktor Makovychuk. Non-smooth newton methods for deformable multi-body dynamics. *ACM Transactions on Graphics (TOG)*, 38(5):1–20, 2019.
- [21] Miles Macklin, Kenny Erleben, Matthias Müller, Nuttapong Chentanez, Stefan Jeschke, and Zach Corse. Local optimization for robust signed distance field collision. *Proc. ACM Comput. Graph. Interact. Tech.*, 3(1), May 2020. doi: 10.1145/3384538. URL <https://doi.org/10.1145/3384538>.
- [22] Meta. Meta quest: Hand tracking in unreal engine. <https://developers.meta.com/horizon/documentation/unreal/unreal-hand-tracking-overview/>, 2026. URL <https://developers.meta.com/horizon/documentation/unreal/unreal-hand-tracking-overview/>. Accessed: 2026-06-21.
- [23] Yashraj Narang, Kier Storey, Ireteyio Akinola, Miles Macklin, Philipp Reist, Lukasz Wawrzyniak, Yunrong Guo, Adam Mora-

vanszky, Gavriel State, Michelle Lu, Ankur Handa, and Dieter Fox. Factory: Fast contact for robotic assembly. In *Robotics: Science and Systems*, 2022.

- [24] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019. doi: 10.1109/CVPR.2019.00025. URL <https://ieeexplore.ieee.org/document/8954065>.
- [25] Kenneth Shaw, Ananye Agarwal, and Deepak Pathak. LEAP Hand: Low-cost, efficient, and anthropomorphic hand for robot learning. In *Proceedings of Robotics: Science and Systems XIX*, Daegu, Republic of Korea, 2023. URL <https://rss2023.github.io/rss2023-website/program/papers/089/>. Paper ID 89.
- [26] Hang Si. Tetgen, a delaunay-based quality tetrahedral mesh generator. *ACM Trans. Math. Softw.*, 41(2), February 2015. ISSN 0098-3500. doi: 10.1145/2629697. URL <https://doi.org/10.1145/2629697>.
- [27] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109. URL <https://ieeexplore.ieee.org/document/6386109>.
- [28] Hongyi Xu, Yili Zhao, and Jernej Barbič. Implicit multibody penalty-based distributed contact. *IEEE Transactions on Visualization and Computer Graphics*, 20(9):1266–1279, 2014. doi: 10.1109/TVCG.2014.2312013. URL <https://ieeexplore.ieee.org/document/6767148>.
- [29] Ehsan Zobeidi and Nikolay Atanasov. A deep signed directional distance function for shape representation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2689–2695, 2024. doi: 10.1109/IROS58592.2024.10801713. URL <https://ieeexplore.ieee.org/abstract/document/10801713>.