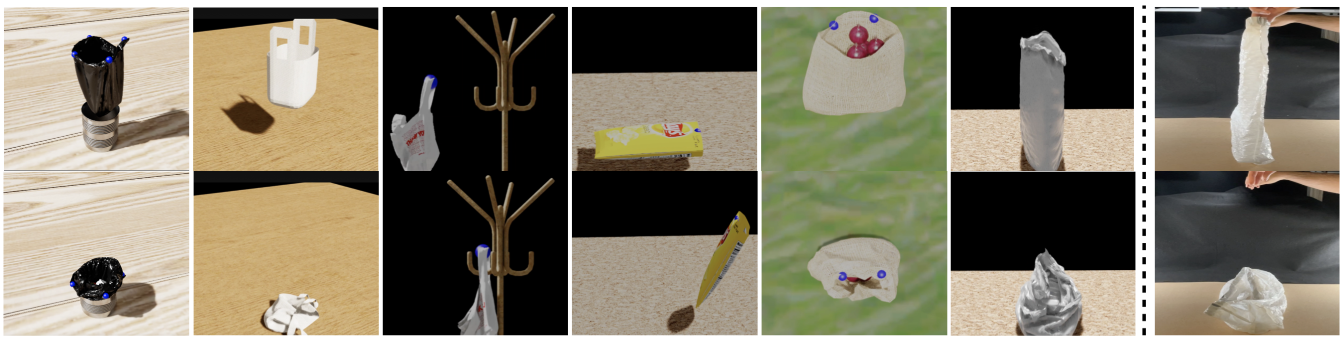
BagSim: Realistic Physics-Based Simulation for Deformable Bags

Lawrence Yunliang Chen
University of California, Berkeley
USA
yunliang.chen@berkeley.edu

Karthik Dharmarajan
University of California, Berkeley
USA
kdharmarajan@berkeley.edu

Andy Huang
University of California, Berkeley
USA
handy1221@berkeley.edu

Jeffrey Tan
University of California, Berkeley
USA
tanjeffreyz02@berkeley.edu



Trash Bag Fitting | Cloth Bag Dropping | Plastic Bag Hanging | Chip Bag Sliding & Lifting | Sack Bag Putting Down | Trash Bag Falling Sim vs Real

Figure 1: Examples of BagSim showing different types of bags going through different motions, including fitting a trash bag into a trash can, dropping a cloth bag onto the floor, hanging a plastic bag onto a hanger, sliding and lifting a chip bag on a table, and putting a sack bag full of onions onto the grass. The right two columns show dropping a trash bag in BagSim compared to real, suggesting realistic visual and dynamics simulation.

ABSTRACT

Simulating the behavior of deformable objects is a challenging task in computer graphics but has wide applications including animation, virtual reality, and gaming. In this project, we focus on simulating deformable bags. Simulating deformable bags is particularly difficult due to their complex shapes, varieties of materials, and complex behaviors for self-collision and collision with other objects. Existing simulators use simplifications such as pretending the bags to be a piece of folded fabric, and fail to accurately simulate critical properties of bags such as handles and different bag shapes. In this project, we present BagSim, an environment built on top of NVIDIA Omniverse for realistically simulating the movement of deformable bags. We design and create meshes and textures for various types of bags (with and without handles or bottoms), and simulate the bag movement (elasticity, dynamics, deformation) for bags of different materials (stiff vs soft) when dragged around by control points. Our simulator allows interaction between the deformable bag and rigid objects. We perform a systematic study of particle-based dynamics (PBD) and finite-element method (FEM) and their parameters and tune the simulation parameters to match real bags. Furthermore, we implement key point tracking for markers in real to quantify the errors between sim and real. Using our custom-created bag meshes, textures, and tuned parameters, BagSim supports many bag types (we use PBD for soft bags and FEM for stiffer bags) and simulates the dynamic movement of deformable bags more accurately than existing simulators (eg. SoftGym, DeformableRavens, DEDO).

CCS CONCEPTS

• **Computing methodologies → Physical simulation**; *Motion processing*; *Collision detection*; *Mesh geometry models*; Ray tracing.

KEYWORDS

Deformable object simulation, Plastic bag, Physical simulation

1 INTRODUCTION

Accurate simulation of deformable objects such as cables, cloth, fluid, and biological tissues is a crucial component in many applications such as animation, computer-aided design, films, and games. The demand for realistic motions becomes even more apparent for virtual reality applications and robotic simulators as the quality of simulation makes a huge difference in the user experience and for training machine learning models. However, it is also well-known that modeling and simulation of such deformable objects are challenging, and they have been an active research area in mechanical engineering and computer graphics for several decades.

There are many different ways to model the deformation of objects [8]. On the one extreme of the spectrum are non-physical approaches, where a designer manually adjusts control points or shape parameters to edit or design the shape. However, this is not scalable. On the other end of the spectrum are approaches that are based on continuum mechanics, which considers the material properties, external forces, and environmental constraints that affect the deformation of objects. However, tuning the parameters, also known as system identification, is also very time-consuming and challenging. Even with identified parameters, there are still aleatoric uncertainties in the real world that cannot be fully modeled. Additionally, simulating and rendering realistic visuals can be slow and inaccurate.

In recent years, there has been huge progress in the physics-based simulation of cables and fabrics, which has been found useful by the robotics community to learn control policies for deformable objects [9, 11, 23]. For example, OpenAI Gym [3] and Mujoco [17] both support rope and cloth simulations, as well as sponge-like soft 3D objects. Lin et al. [12] build a higher-fidelity simulator for ropes, cloth, and water. Huang et al. [10] simulate soft 3D objects such as fruits, internal organs, and flexible containers.

In this project, we focus on deformable bag simulation. There has been a growing interest in simulating the behavior of bags, which are ubiquitous in daily life, including in home, commercial, and industrial settings. Building a high-fidelity simulator can be useful in robotics as it can significantly reduce the real-world data required for training models. For example, [4, 5] study the task of opening a plastic bag and inserting objects, and they train their model entirely in real, which requires collecting a large amount of real-world data. This can potentially be accelerated with the help of a bag simulator. Similarly, [22] leveraged simulation to learn a robot policy for hanging a bat onto a rack. In contrast to cloth, however, there exist very few high-fidelity simulators for bags. DeformableRavens [15] is one of the first few works that present physics-based bag simulation. However, their bag model is hugely simplified. [19] presents a graph-based model of a handbag and its simulation. Antonova et al. [1] build up on that and allow more complex movements. Still, the bag model is limited and the parameters are not systematically tuned to match with the real world. We aim to address these shortcomings.

Simulating deformable objects like bags is challenging due to the variety of behaviors depending on the material used. Existing simulators fail to accurately simulate critical properties of bags such as handles and different bag shapes, which limits their usefulness for several computer graphics applications. In this work, we present BagSim, a physics-based simulator environment built on top of NVIDIA Omniverse for simulating deformable bags.

We design and create meshes and textures for various types of bags (with and without handles or bottoms), and simulate the bag movement (elasticity, dynamics, deformation) for bags of different materials (stiff vs soft) when dragged around by control points. Our simulator allows interaction between the deformable bag and rigid objects. We perform a systematic study of particle-based dynamics (PBD) and finite-element method (FEM) and their parameters and tune the simulation parameters to match real bags. Furthermore, we implement key point tracking for markers in real to quantify the errors between sim and real. Using our custom-created bag meshes, textures, and tuned parameters, BagSim supports many bag types (we use PBD for soft bags and FEM for stiffer bags) and simulates the dynamic movement of deformable bags more accurately than existing simulators.

2 RELATED WORK

2.1 Deformable Object Simulation

There are many ways for simulating deformable objects, including non-physical models, mass-spring models, particle models, continuum models, and finite element methods [8]. Although most models incorporate some physical principles to compute the shapes or motions of deformable objects, in many applications, particularly in design, employ purely geometric techniques[2]. This highly relies on the skill of the designer and is not scalable.

Mass-spring systems are a physics-based model that has been used widely and effectively for modeling deformable objects due to their simplicity [8]. In mass-spring systems, an object is modeled as a collection of point masses connected by springs. Using Newton's laws, a simple second-order differential equation can be derived, and the position of the masses can be simulated using numerical ODE solvers, such as implicit Euler's method and Verlet integration. Although they are simple to implement, mass-spring systems have some drawbacks. First, it is not very accurate as the discrete model is a significant approximation of the true physics that occurs in a continuous body. Second, it has poor stability in stiff systems, and extra damping needs to be added.

Continuum models treat deformable objects as a continuum: solid bodies with mass and energies distributed throughout [8]. The finite element method (FEM) is a widely-used approximation for a continuous function that satisfies some equilibrium expressions [14]. In FEM, the continuum, or object, is divided into elements joined at discrete node points. A function that solves the equilibrium equation is found for each element. While more accurate than mass-spring systems, FEM is significantly computationally heavier.

Many current state-of-the-art physics-bases simulator engines use particle-based systems, such as SoftGym [12], Isaac Gym [13], and NVIDIA Omniverse. Particle-based systems can be derived from continuum mechanics [6] and achieve a good balance between fidelity and speed. In this work, we leverage the PhysX engine of NVIDIA Omniverse and both its PBD and FEM models for simulating deformable bags.

2.2 Modeling and Simulation of Cables, Fabrics, and Fluids

Researchers have developed several analytic physics models to describe the dynamics of moving cables. For example, Gatti-Bonoa et al. [7] present a 2D dynamic model of fly fishing. They model the fly line as a long elastica and the fly rod as a flexible Euler-Bernoulli beam, and propose a system of differential equations to predict the movement of the fly line in space and time. In contrast to continuum models, Wang et al. [18] propose using a finite-element model to represent the fly line by a series of rigid cylinders that are connected by massless hinges.

Prior work related to the simulation of deformable objects such as fabrics, cables, and fluids also focused on creating an artificial environment in which robots could be trained to manipulate such objects. [20] develops an efficient high-fidelity simulator for simulating fluids as they interact with rigid objects. [15] presents an

Figure 2: Meshes and textures we created from scratch for various types of bags using Blender.

open-source simulation and benchmark called *DeformableRavens* based on the PyBullet engine for robotic manipulation of deformable 1D, 2D, and 3D objects. It provides support for such deformable objects using soft-body physics based on mass-springs and self-collisions among vertices. While this approach to simulation is well-tested and simple, it is far from accurate. In [11], the authors study various simulator models for accurately simulating the movement of a cable, including a PyBullet model consisting of capsule rigid bodies connected by 6 DOF spring constraints, a soft-body rod connected to a capsule rigid body at the endpoint, and a segmented model also consisting of a string of capsule rigid bodies but is connected using ball joints. SoftGym [12] is one of the best simulators for simulating cloths, also using particle-based models. However, it does not support more complex fabric meshes as the fidelity rapidly decreases with complex self-collision modeling.

2.3 Simulation of Deformable Bags

DeformableRavens [15] is one of the first few works that offer physics-based simulation of bags. However, their bag model is a hugely-simplified mesh by wrapping a single piece of cloth so it does not have handles. Additionally, as its physics is based on mass-springs systems, the simulated material is far from accurate as they scrunch up, flatten out, form creases, and ultimately retain some of the changes in their structure due to external forces. Mass-spring systems like the ones used in DeformableRavens can not accurately reflect these properties of bags, which may cause robots trained using the simulation to behave in unexpected ways in real-world scenarios. Moreover, it can easily go unstable when the parameters of the material are beyond some ranges.

Weng et al. [19] presents a graph-based model of a handbag and its simulation. They design the bag templates with handles to include multi-hole structures using Blender, and build the simulation environment in Unity based on Obi Cloth extension. Antonova et al. [1] build up on that by using similar mesh models but use the Isaac Gym [13] environment to allow more complex movements. Still, the bag model is limited and the parameters are not systematically tuned to match with the real world. In this work, we leverage both the particle-based model and finite element method offered by the Omniverse PhysX engine to create more realistic simulations both visually and dynamically by choosing the suitable model for different bags with tuned parameters.

3 TECHNICAL APPROACH

To bridge the sim2real visual and dynamics gaps found in prior work, we create new visually realistic bag models, systematically study parameters of deformable physics models, add dynamic control points, and develop a method for comparing simulated bags to real ones.

Figure 3: Comparison of Position-Based Dynamics (PBD) and Finite Element Analysis (FEM). Left: PBD, where the material is modeled as a system of particles. Right: FEM, where the material domain is discretized as a mesh consisting of a finite number of geometry elements (tetrahedra).

3.1 Bag Models

We choose to model a set of diverse bags to showcase visual and dynamic differences both between the different bags and from other state-of-the-art simulators. To showcase a cloth material for a bag, we create a Trader Joe's tote bag from scratch in Blender. Similarly, we create a black plastic trash bag and a plastic bag with circular handles from Blender. To add a more rigid bag option, we model a Lay's potato chip bag. Finally, we design a sack bag to handle food. Results are shown in Figure 2.

To create a more realistic bag shape, we apply a series of elastic deforms, pinches, and cloth inflations starting from a normal sphere. Generating the textures for the bags is more complicated, however. Texture mapping in Blender is simple enough to get started with but has a high learning curve when assigning more complicated textures. For example, applying a uniform texture to an entire mesh is straightforward enough after some tedious manipulation of the mesh in UV coordinates. However, when applying multiple textures to a mesh and overlaying them on top of each other (such as a logo over a cloth), it becomes a much more difficult task involving more complicated mesh manipulations in several UV coordinate maps and may result in unexpected mesh bugs when imported into Omniverse.

3.2 Bag Physics Simulation

Due to the wide variety of properties that bags in the real world can take, we propose a method that utilizes a combination of two popular methods of simulating deformable bodies, from previous

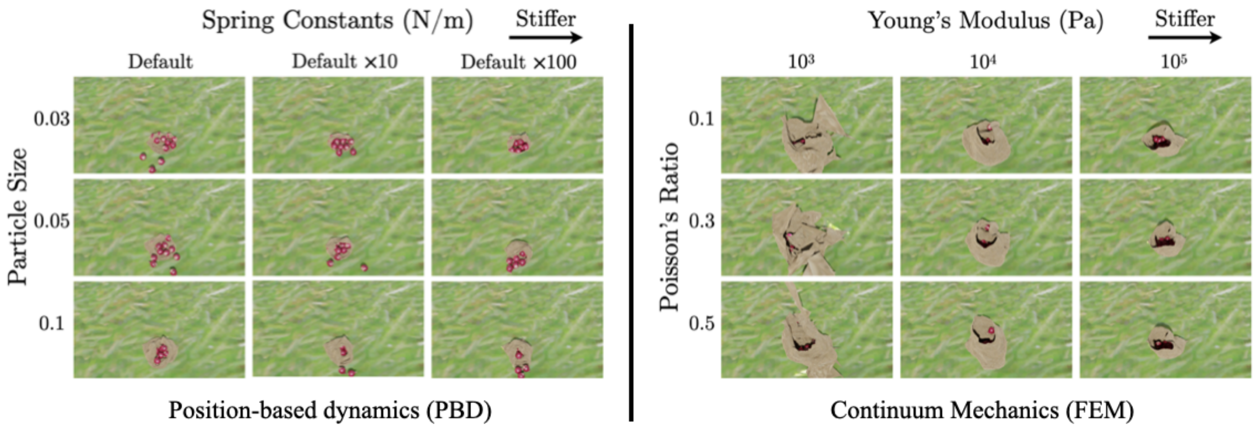


Figure 4: Meshes and textures we created from scratch for various types of bags using Blender.

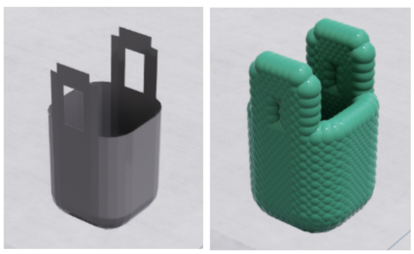


Figure 5: Mesh (left) and particle views (right) of a custom bag mesh with handles.

literature, Position-Based Dynamics (PBD), and FEM (Finite Element Method). A visual comparison of the underlying structures in both approaches is shown in Figure 3.

3.2.1 Position-Based Dynamics (PBD). In position-based dynamics, the materials is modeled as a discrete system with particles. The simulation works as an implicit integration with internal forces derived from holonomic constraints, which might include temporary and unilateral constraints [21]. Compared to mass-spring systems, PBD is more accurate and stable. One limitation of PBD, however, is that it is not always easy to find a meaningful physical interpretation of some of the parameters. This implies more tuning efforts. In NVIDIA Omniverse, the main parameters include the structural, bending, and shearing constants, and the particle size.

3.2.2 Finite Element Method (FEM). Continuum mechanics offers a more physically accurate description of modeling material deformation. A constitutive model defines the relationship between stress and measurement of material deformation [21]. For isotropic Hookean materials, the coefficient matrix is fully determined by two scalars: (i) Young's modulus which describes material stiffness, and (ii) Poisson's ratio, a ratio of lateral and longitudinal strain. Simulating such a system requires solving a partial differential equation, which is only feasible for simple and small-scale problems. For general cases, FEM is commonly used. The central idea is to discretize the material as a mesh consisting of a finite number of

Figure 6: Illustration of keypoint tracking in real and sim. We attach markers to bags and record videos of the bag movement in both sim and real. For real observations, we track the markers using OpenCV, and compare them with the positions of the markers in sim to evaluate the simulation error.

geometry elements, e.g., tetrahedra. FEM can be more accurate for certain materials than PBD, but is more computationally expensive.

We build on top of NVIDIA Omniverse, as it has several capabilities that allow it to take in arbitrary meshes and convert them into deformable objects (FEM based dynamics) or particle based cloths (PBD dynamics), as shown in Figure 1. Using the default parameters when converting an arbitrary mesh to a cloth or deformable body leads to very odd results. Upon some parameter tuning, however, the cloth can be made to look like a proper bag. We perform a systematic analysis of parameters for both Position-Based Dynamics and Finite Element Method in Section 4 to qualitatively determine how the bags behave.

We utilize Omniverse's ability to handle collisions between rigid objects and deformable objects or particle cloths. Figure 1 shows different bags in collision with various objects in different scene, including the ground plane and a group of red onions.

3.3 Dynamic Control Points

To enable more dynamic motion of bags, we implement a new NVIDIA Omniverse extension, which creates a GUI for setting up motions for bags. A user can create control points, which are blue dots that can be attached to the bag. The user can then specify waypoints through the interface for the control point to move through. Motions through waypoints are executed with linear interpolation, as the goal of the dynamic motions is to be able to check bag dynamics with the real world.

3.4 Real-and-Sim Marker Tracking

To be able to compare the dynamics of bags in the real world to dynamics in the simulation, we propose a marker tracking method, as shown in Figure 6. We add square markers to the bags in similar locations in both simulation and real, and execute a similar trajectory. We implement marker tracking for real bags using OpenCV color thresholding, followed by contour detection in the masked image. The distances between the corresponding markers can then be measured to provide the difference in the bag motion from simulation and real using the following formula:

$$\text{Error} = \sum_i ||\mathbf{x}_i^{\text{real}} - \mathbf{x}_i^{\text{sim}}||^2, \tag{1}$$

where i is the index of the markers in both sim and real. The lower the error, the higher fidelity of the simulation is. Zero-order optimization such as differential evolution [16] can then be potentially used to find parameters that minimize the error.

4 EVALUATION

We evaluate our technical contributions by creating various dynamic scenes as shown in 1. For these bags, the parameters are tuned based upon human observation. The blue control points manipulate the bags in various ways, such as lifting a trash bag out of a garbage can, putting a plastic bag on a hook, or placing down a sack bag. We also systematically tune PBM and FEM parameters for one scene and compare certain scenes with real world examples.

4.1 PBD and FEM Parameter Tuning

To learn more about the effect of certain parameters on the bag's dynamics for both PBD and FEM, we change their values and record the bag's motion in the same scene, as shown in Figure 4. The scene is dropping the sack bag while it is filled by 14 red onions. This scene in particular, allows for visualization of the creases of the bag as it falls onto the ground, as well as the effect of the red onions on the bag due to collisions.

For the PBD simulation, we increase the order of magnitude of the structural/bending/shear constants by 2 at most. We change the particle size from the values of 0.03 to 0.05 to 0.1. As the spring constants increase, the bag becomes stiffer. We also notice that increasing the particle size makes the bag stiffer. With smaller particle sizes, such as 0.03, it is possible for the red onions to fit through gaps in the particle mesh, allowing the onions to escape from the bag entirely. This is another unrealistic undesirable property, prompting to use larger particles sizes.

For the FEM simulation, we increase Young's Modulus by 1-2 orders of magnitude, and change Poisson's ratio from 0.1 to 0.3 to 0.5. We notice that changing Poisson's ratio does not produce a significant difference in the bag dynamics for this particular scene. Interestingly, extremely low Young's Modulus values can result in instable or over stretchy material, leading to spiky looking bags.

In general, the PBD simulation was able to better capture wrinkles and more fine deformation of the bag, whereas the FEM simulation held a more rigid shape. This suggests that depending on the type of bag, whether extremely stiff or more soft, a different method can be used such as FEM or PBD.

4.2 Sim-Real Comparison

We model three scenes in simulation that were executed with real bags. One scene is lifting up a Lay's chip bag, as shown in Figure 1. Since the Lay's chip bag was determined to be stiff, we modeled it with FEM and tuned the parameters manually. The overall behavior of the chip bag is similar in both simulation and real, but the simulated bag tends to have too much of an elastic effect, such that when the control point is stopped, the bag continues to push a non-trivial amount upward.

The second scene compared against the real world is dropping a gray trash bag. Since the gray trash bag has a smoother, finer material, we model its dynamics with PBD and manually tune its parameters. While the mesh of the gray trash bag does not perfectly represent the initial position of the bag in the real world, when both bags land, the creases and positions of deformation are roughly comparable.

As described in Section 3.4, we add markers to both real world and simulated bags. When attempting to perform a proper comparison, this method did not work quite as nicely, as some of the tags got sometimes occluded, and the cameras positions are not exact. Due to these issues, better bag tracking methods can be considered in future work.

5 CONCLUSION AND FUTURE WORK

In this project, we present BagSim, an environment built on top of NVIDIA Omniverse for realistically simulating the movement of deformable bags. We design and create meshes and textures for various types of bags, including tote bags made of fabric material, trash and shopping bag with plastic material, as well as chip bags and sacks. We systematically compare PBD and FEM models and study the effect of the parameters, including the spring constants and the particle sizes for PBD and Young's modulus and Poisson ratio for FEM. We learned that PBD is more visually realistic for soft and stretchy bags, while FEM is more realistic for stiff bags (such as the chip bag).

Furthermore, we build extensions on Omniverse to add control points and define their trajectories. This allows us to simulate diverse behaviors, such as fitting a trash bag into a trash can, dragging and lifting a bag containing rigid objects, and hanging a bag onto a hanger. Additionally, we implement key point tracking for markers in real and propose a way to quantify the errors between sim and real. Applying what we learned, we select suitable models for different bags and systematically tune their parameters, making our bags more realistic and behaviorally diverse than existing environments.

Some future directions to further improve BagSim include supporting non-homogeneous material and translucent bags. For some bags, different parts are made of different materials, and our current model does not take that into account. Additionally, future work can bring in robot models into the environment and model the interaction between the robot grippers and the bag, which would be useful for robotics research.

6 CONTRIBUTIONS

6.1 Lawrence Yunliang Chen (CS284A)

Lawrence identifies the sim2real gap for deformable object simulation, proposes the problem idea, and formalizes our project goal. He conducts literature review on the existing simulators, including their models/engines, and bag meshes. He works with Karthik to explore the capabilities of NVIDIA Omniverse, studies the comparison between PBD and FEM models and their parameters, and proposes scenes to create. He makes the presentation slides, edits all videos, and writes a significant portion of the project report.

6.2 Karthik Dharmarajan (CS184)

Karthik researches various simulators for performing the bag simulation, including creating an initial attempt in NVIDIA Omniverse. He programs the custom Omniverse extension for controlling control points on bags, and sets up and runs most of the scenes in Omniverse. He runs various evaluation experiments such as the PBD and FEM tuning study. He helps write significant portions of the project report.

6.3 Andy Huang (CS184)

Andy researches the capability of the Unity engine in the early stage of the project, and is the major designer of our bag meshes. He designs the meshes and the textures and iterates on them based on other members' feedback of the bag behavior in the simulator. He also helps draft the project proposal and milestone report and makes helpful edits to the presentation.

6.4 Jeffrey Tan (CS184)

Jeffrey Tan researches various simulators including Obi Cloth and Unity in the early stage of the project. He then helps Karthik study Omniverse, tunes the parameters, and creates various videos. Further, he implements the marker tracking for real bags, allowing us to compare between sim and real. He also makes helpful edits to the project proposal, milestone report, and presentation.

ACKNOWLEDGMENTS

We thank Professors Ren Ng and James O'Brien, and the CS 184 course teaching staff for their support and helpful feedback for this project.

REFERENCES

[1] Rika Antonova, Peiyang Shi, Hang Yin, Zehang Weng, and Danica Kragic Jensfelt. 2021. Dynamic environments with deformable objects. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

[2] Jernej Barbič, Marco da Silva, and Jovan Popović. 2009. Deformable object animation using reduced optimal control. In *ACM SIGGRAPH 2009 papers*. 1–9.

[3] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. Openai gym. *arXiv preprint arXiv:1606.01540* (2016).

[4] Lawrence Yunliang Chen, Baiyu Shi, Roy Lin, Daniel Seita, Ayah Ahmad, Richard Cheng, Thomas Kollar, David Held, and Ken Goldberg. 2023. Bagging by Learning to Singulate Layers Using Interactive Perception. *arXiv preprint arXiv:2303.16898* (2023).

[5] Lawrence Yunliang Chen, Baiyu Shi, Daniel Seita, Richard Cheng, Thomas Kollar, David Held, and Ken Goldberg. 2022. AutoBag: Learning to Open Plastic Bags and Insert Objects. *arXiv preprint arXiv:2210.17217* (2022).

[6] Olaf Etzmuss, Joachim Gross, and Wolfgang Strasser. 2003. Deriving a particle system from continuum mechanics for the animation of deformable objects. *IEEE Transactions on Visualization and Computer Graphics* 9, 4 (2003), 538–550.

[7] Caroline Gatti-Bono and NC Perkins. 2004. Numerical model for the dynamics of a coupled fly line/fly rod system and experimental validation. *Journal of sound and vibration* 272, 3-5 (2004), 773–791.

[8] Sarah FF Gibson and Brian Mirtich. 1997. *A survey of deformable modeling in computer graphics*. Technical Report. Technical report, Mitsubishi Electric Research Laboratories.

[9] Huy Ha and Shuran Song. 2022. Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding. In *Conference on Robot Learning*. PMLR, 24–33.

[10] Isabella Huang, Yashraj Narang, Clemens Eppner, Balakumar Sundaralingam, Miles Macklin, Tucker Hermans, and Dieter Fox. 2021. Defgraspsim: Simulation-based grasping of 3d deformable objects. *arXiv preprint arXiv:2107.05778* (2021).

[11] Vincent Lim, Huang Huang, Lawrence Yunliang Chen, Jonathan Wang, Jeffrey Ichnowski, Daniel Seita, Michael Laskey, and Ken Goldberg. 2022. Real2sim2real: Self-supervised learning of physical single-step dynamic actions for planar robot casting. In *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 8282–8289.

[12] Xingyu Lin, Yufei Wang, Jake Olkin, and David Held. 2021. Softgym: Benchmarking deep reinforcement learning for deformable object manipulation. In *Conference on Robot Learning*. PMLR, 432–448.

[13] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. 2021. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470* (2021).

[14] Andrew Nealen, Matthias Müller, Richard Keiser, Eddy Boxerman, and Mark Carlson. 2006. Physically based deformable models in computer graphics. In *Computer graphics forum*, Vol. 25. Wiley Online Library, 809–836.

[15] Daniel Seita, Pete Florence, Jonathan Tompson, Erwin Coumans, Vikas Sindhwani, Ken Goldberg, and Andy Zeng. 2021. Learning to rearrange deformable cables, fabrics, and bags with goal-conditioned transporter networks. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 4568–4575.

[16] Rainer Storn and Kenneth Price. 1997. Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization* 11, 4 (1997), 341.

[17] Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 5026–5033.

[18] Gang Wang and Norman Wereley. 2011. Analysis of fly fishing rod casting dynamics. *Shock and Vibration* 18, 6 (2011), 839–855.

[19] Zehang Weng, Fabian Paus, Anastasiia Varava, Hang Yin, Tamim Asfour, and Danica Kragic. 2021. Graph-based task-specific prediction models for interactions between deformable and rigid objects. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 5741–5748.

[20] Zhou Xian, Bo Zhu, Zhenjia Xu, Hsiao-Yu Tung, Antonio Torralba, Katerina Fragkiadaki, and Chuang Gan. 2023. FluidLab: A Differentiable Environment for Benchmarking Complex Fluid Manipulation. In *International Conference on Learning Representations*.

[21] Hang Yin, Anastasia Varava, and Danica Kragic. 2021. Modeling, learning, perception, and control methods for deformable object manipulation. *Science Robotics* 6, 54 (2021), eabd8803.

[22] Yifan You, Lin Shao, Toki Migimatsu, and Jeannette Bohg. 2021. Omnihang: Learning to hang arbitrary objects using contact point correspondences and neural collision estimation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 5921–5927.

[23] Mélodie Hani Daniel Zakaria, Miguel Aranda, Laurent Lequièvre, Sébastien Lengagne, Juan Antonio Corrales Ramón, and Youcef Mezouar. 2022. Robotic Control of the Deformation of Soft Linear Objects Using Deep Reinforcement Learning. In *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*. IEEE, 1516–1522.