NeRF-ysics: A Differentiable Pipeline for Enriching NeRF-Represented Objects with Dynamical Properties

Simon Le Cleac'h[1], Taylor Howell[1], Mac Schwager[2] and Zachary Manchester[3]

***Abstract*— We present NeRF-ysics a differentiable pipeline for enriching NeRF-represented objects with dynamical properties, enabling accurate real-to-simulation transfer and robotic interaction. These properties, including mass, inertia, and coefficients of frictions, are learned efficiently with gradient-based optimization by utilizing a differentiable physics simulator. This allows to build object models that are visually and *dynamically* accurate from still images and videos of the object in motion. This simulator can serve as the internal predictive model used by a robot to interact with a NeRF-represented environment. The differentiability is an essential property enabling efficient gradient-based optimizations. For instance, optimization of robot behavior interacting with NeRF via motion planning, trajectory optimization or policy optimization.**

I. INTRODUCTION

Constructing models of a robot's environment from camera inputs is a fundamental challenge in robotics and enabling robots to automatically build visually and dynamically accurate models of objects in their surroundings would be a major advancement for robotic manipulation.

These object models should be sufficiently dynamically accurate to enable synthesis of robot behaviors leveraging interactions with these objects. Differentability of such models would further enable the robot to quickly adapt its motion-plan or policy when interacting with its environment.

Recent advances in view synthesis [6] enable construction of visual models of a scene from a limited number of images with camera pose annotation. More recent works decompose the scene into distinct object instances each represented by an individual NeRF [4], [9]. In our work, each object is represented by a NeRF and we use the NeRF's occupancy field as a proxy for the geometry of the object.

The NeRF accurately models the visual properties of the object (geometry, color, reflectance). This might be sufficient for a range robotic applications where the objective is to avoid interactions with NeRF-represented scenes or objects [2]. However, additional dynamical properties are required when the goal is to interact with the environment e.g., robotic manipulation. Thus, we propose a differentiable pipeline that can be utilized to efficiently learn dynamical properties from a limited amount of data. A parallel can be drawn between this approach and how differentiable rendering tools enables efficient regression of visual properties [5]. These physics-conditioned NeRF-enhancements

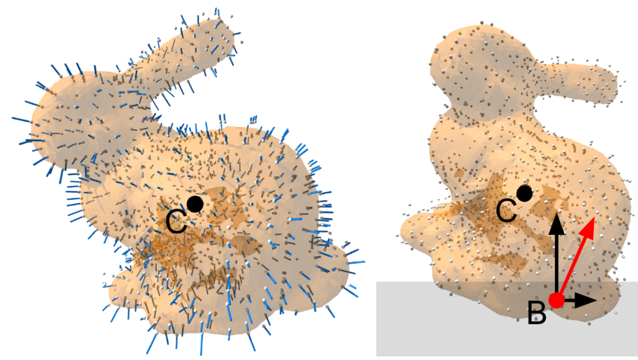

Fig. 1: Left, we visualize the result of the NeRF's occupancy field pre-processing step. First, we sample points (white dots) from the occupancy field between a minimal and maximal occupancy value. This biases the sampling of points towards the boundary of the object. Then for each sampled point, we compute the occupancy gradient (blue arrows) to obtain an approximate outward direction. This information is leveraged when simulating impact between two NeRF-represented objects. Finally, we obtain a guess over the center of mass (large black dot denoted C) and inertia matrix of the object by treating the occupancy field as a density field. Right, we illustrate how contact forces are applied to the object when it is in collision with a half-space (grey area). Forces are applied at the barycenter B of the sampled points violating the half-space constraints. They are composed of a normal component related to the object's stiffness and a tangential component related to the object's coefficient of friction. The amount of interpenetraton is amplified for clarity.

complete the pipeline building visually and *dynamically* accurate models of rigid objects from real-world camera inputs (Figure 2). Our key contributions are:

- An efficient and differentiable formulation of rigid-body contact operating on arbitrary occupancy fields
- A regression of dynamical properties from real-data demonstrating the improved accuracy of dynamics-augmented NeRF's.

II. NeRF-ysics

A. Occupancy Field Processing

We can simulate a NeRF-represented object by augmenting its representation with dynamical properties. These properties include the mass, center of mass, inertia matrix, and several scalar parameters modeling contact interactions: impact stiffness, coefficient of friction. In an offline phase, we process the occupancy field provided by the NeRF representation to extract information useful for contact simulation. The process is described in Figure 1.

*Toyota Research Institute provided funds to support this work.

[1]Mechanical Engineering Department, Stanford University {simonlc, thowell}@stanford.edu
[2]Aeronautics and Astronautics Department, Stanford University schwager@stanford.edu
[3]The Robotics Institute, Canergie Mellon University zacm@cmu.edu

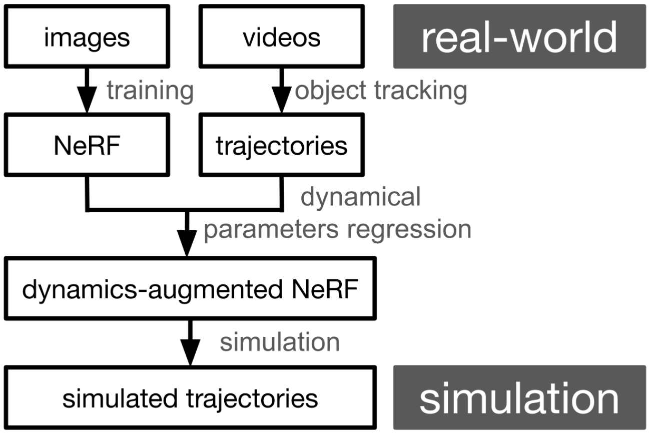

Fig. 2: Computational pipeline for processing real-world camera inputs i.e., still images and videos of a rigid object in motion. The pipeline produces a visually and dynamically accurate simulation model. This resulting model can be leveraged to synthesize robot behaviors: motion-planning, trajectory optimization, policy optimization for different application domains e.g., robotic manipulation.

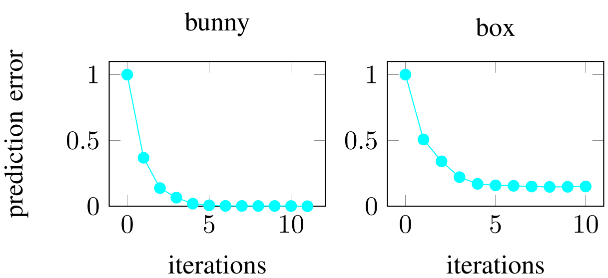

Fig. 3: Using Newton's method, we quickly regress the object's coefficient of friction from 5 tossing trajectories, each trajectory is 0.3-second long. For the bunny, we use synthetic data and we recover the exact friction parameter used to generate the trajectories. For the box, we use real data [8] and we identify a friction coefficient that decreases the trajectory prediction error tenfold.

B. Contact Interaction

Given the mass and inertia of the object, we can simulate its behavior in free space. Contact interactions with the environment or with a robot requires additional parameters e.g., the object stiffness and frictional properties. For brevity, we focus on contact between a NeRF-represented object and a half-space object (e.g., the floor). We can similarly simulate contact between NeRF and more complex shapes (e.g., sphere) or even other NeRF objects. Given the current position and orientation of the object, we identify the active points i.e., the ones intersecting with the half-space. We compute the barycenter B of these weighted active points (Figure 1). This is the point at which contact forces and torques will be applied on the NeRF-represented object.

C. Variational Integrator

We embed this contact model into a variational integrator [1]. With this formulation, we can efficiently differentiate the simulator using the implicit function theorem [3].

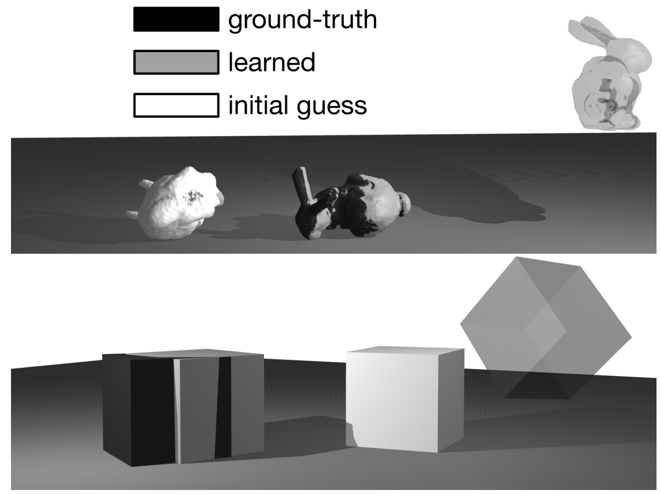

Fig. 4: We visualize how closely the NeRF-represented model augmented with learned coefficient of friction matches ground-truth trajectories. Each object is tossed from an initial position (right, faded gray), we display the final position of the ground-truth (black), the learned model (gray) and the initial model (white). For the bunny, the learned model exactly matches the ground-truth synthetic data. For the cube, the learned model performs significantly better than the initial model.

III. Learning Framework

We demonstrate how we can leverage differentiable simulation to efficiently estimate dynamical properties from object trajectories. The objective is to minimize the error between a ground-truth trajectory of length T and a trajectory generated by the simulator. We formulate an optimization problem to identify the object's coefficient of friction,

$$\begin{array}{ll} \underset{\theta}{\text{minimize}} & \sum_{t=0}^{T} ||f(x_t, \hat{u}_t; \theta) - \hat{x}_{t+1}|| \\ \text{subject to} & x_0 = \hat{x}_0, \end{array} \tag{1}$$

where x_t, u_t are the object's state and control input at time step t; f is the dynamics parameterized by θ the coefficient of friction, the symbol $\hat{\ }$ denotes ground-truth quantities.

IV. Results

We identify the coefficient of friction for two NeRF-represented objects: a box and a bunny (Figures 3, 4). Leveraging the simulator's differentiablity, we can apply the Gauss-Newton method [7] to Problem 1 and identify the friction coefficient in less than 10 seconds using a limited amount of data.

Acknowledgment

The authors would like to thank Jiajun Wu for technical discussions; Hong-Xing Yu and Michelle Guo for support and providing pre-trained object-centric neural scattering functions.

REFERENCES

[1] Taylor A. Howell, Simon Le Cleac'h, Zico Kolter, Mac Schwager, and Zachary Manchester. Dojo: A differentiable simulator for robotics. *arXiv preprint arXiv:2203.00806*, 2022.

[2] Michal Adamkiewicz, Timothy Chen, Adam Caccavale, Rachel Gardner, Preston Culbertson, Jeannette Bohg, and Mac Schwager. Vision-only robot navigation in a neural radiance world. *IEEE Robotics and Automation Letters*, 7(2):4606–4613, 2022.

[3] Ulisse Dini. *Lezioni di analisi infinitesimale*, volume 1. Fratelli Nistri, 1907.

[4] Michelle Guo, Alireza Fathi, Jiajun Wu, and Thomas Funkhouser. Object-centric neural scene rendering. *arXiv preprint arXiv:2012.08503*, 2020.

[5] Hiroharu Kato, Deniz Beker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Differentiable rendering: A survey. *arXiv preprint arXiv:2006.12057*, 2020.

[6] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020.

[7] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, second edition, 2006.

[8] Samuel Pfrommer, Mathew Halm, and Michael Posa. Contactnets: Learning discontinuous contact dynamics with smooth, implicit representations. *arXiv preprint arXiv:2009.11193*, 2020.

[9] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13779–13788, 2021.