Learning Articulated Rigid Body Dynamics Simulations From Video

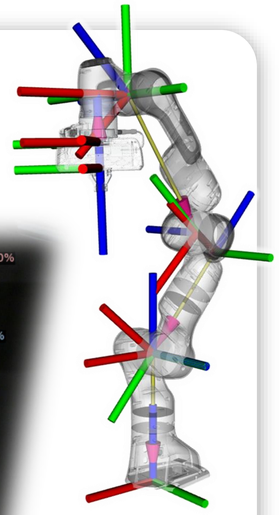
Eric Heiden[1], Ziang Liu[2], Vibhav Vineet[3], Erwin Coumans[4], Gaurav S. Sukhatme[1]

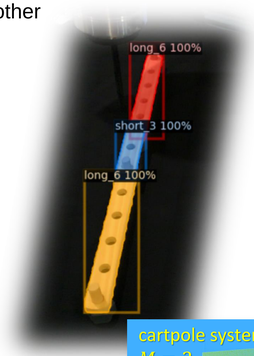
1 USC University of Southern California
2 Stanford University
3 Microsoft Research
4 NVIDIA

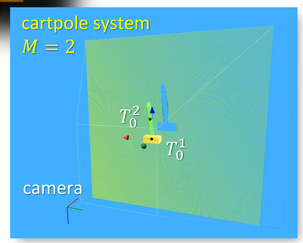
https://eric-heiden.github.io/video2sim

Motivation

- Automate simulation setup that requires the definition of the kinematic structure ("URDF") and simulation parameters
- Build simulator from pixel-based observations (depth or RGB video) instead of relying on motion capture or other instrumentation







Approach

1. Object recognition

- Identify known objects in the video via Detectron2 instance segmentation network

2. Pose tracking

- Set up scene + camera in inverse renderer **nvdiffrast**

$$f_{\text{rast}}\colon \mathrm{SE}(3)^M \times \mathrm{SE}(3) \to \mathbb{R}^{H\times W\times C}$$

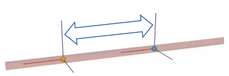
- Find world transforms of rigids bodies in the image via pixel-based loss:

$$\underset{[T_0^1, T_0^2, \ldots, T_0^M]}{\text{minimize}} \left\| f_{\text{rast}}([T_0^1, T_0^2, \ldots, T_0^M], T_0^{\text{cam}}) - \mathbf{x}_{\text{real}} \right\|^2$$

3. Articulation inference

- Consider relative motion $T_i^j[t]$ between bodies i and j at time t

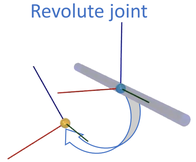
Prismatic joint



$$\Delta r = T_i^j[t+1].r - T_i^j[t].r$$
$$\Delta p = T_i^j[t+1].p - T_i^j[t].p$$

Joint axis: $\frac{\Delta p}{\|\Delta p\|}$

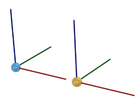
Revolute joint



Joint axis: $\frac{\Delta r}{\|\Delta r\|}$

Pivot point: $T_i^j[t].p + \frac{\Delta r + \Delta p}{\|\Delta r\|^2}$

Static joint

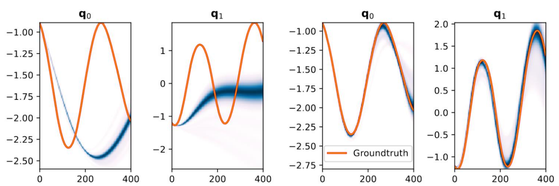


Static transform $T_i^j[t]$

- For each joint type, find joint parameters for all time steps via RANSAC
- Memorize joint and model error for best candidate in *cost matrix* C
- Find articulations as minimum spanning trees in C with root bodies I_{root}

4. System identification

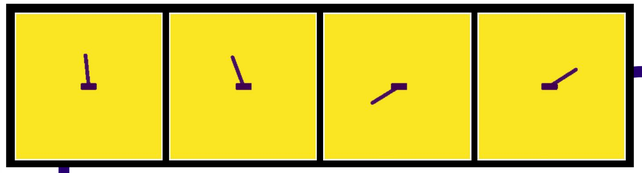
- Infer posterior distribution over simulation parameters via particle-based inference algorithm Constrained Stein Variational Gradient Descent (CSVGD)
- Combine differentiable physics engine and rasterizer to infer dynamics parameters from video



(a) SVGD (b) CSVGD

Video2Sim Pipeline

Depth or RGB color image sequence

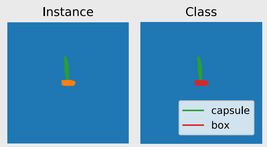


First frame

Object recognition

Detectron2

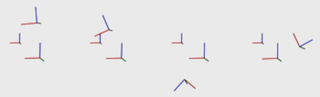
Instance segmentation of initial frame



Object instances, segmentation maps

Pose tracking

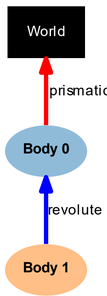
Find poses of objects by minimizing pixel deviation through inverse rendering



3D world poses of objects

Articulation inference

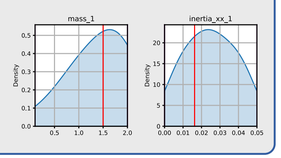
Determine joint types, kinematic parameters, joint position through RANSAC



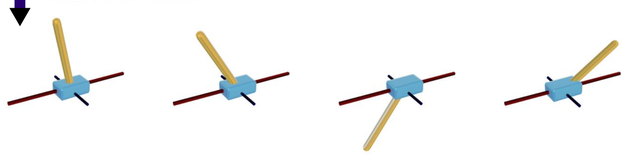
Articulation model

System identification

Bayesian inference of simulation parameters via differentiable simulation + inverse rendering



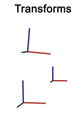
Realistic simulation

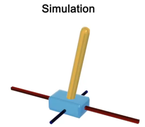


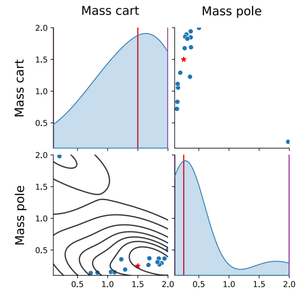
Experiments

Cartpole

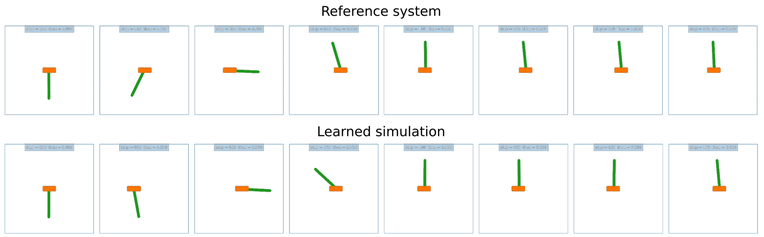
Find simulation from depth video of a cartpole simulated in PyBullet





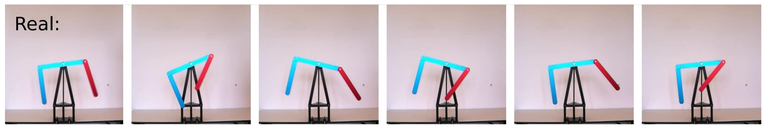


Model-predictive control via Model Predictive Path Integral (MPPI)

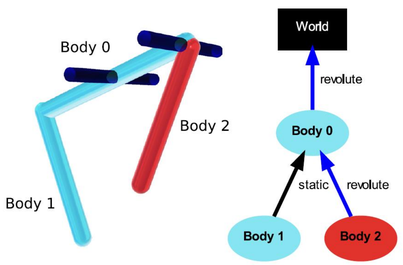


Nikolaus Rott's coupled pendulum

Find simulation from RGB video of a real mechanism



Inferred articulation:



References

Heiden, Millard, Coumans, Sheng, Sukhatme. **NeuralSim: Augmenting Differentiable Simulators with Neural Networks**. *ICRA 2021.*

Murthy, Macklin, Golemo, Voleti, Petrini, Weiss, Considine, Parent-Levesque, Xie, Erleben, Paull, Shkurti, Nowrouzezahrai, Fidler. **gradSim: Differentiable simulation for system identification and visuomotor control.** *ICLR,* 2021.

Heiden, Denniston, Millard, Ramos, Sukhatme. **Probabilistic Inference of Simulation Parameters via Parallel Differentiable Simulation.** *ICRA, 2022.*

Tiny Differentiable Simulator