Manipulating Articulated Objects from Demonstrations

Team 6

Ish Mehta, Jesse Dill
imehta34@gatech.edu, jdill33@gatech.edu

Abstract: One vital problem in robotics is to ensure the policy a robot has learned is accurate, especially when it comes to handling arbitrarily complex objects in real life. A useful approach would be to represent the learned robot policy in simulation, along with a reconstruction of the aforementioned object. Following this motivation, this work aims at learning a policy to manipulate articulated objects from demonstrations, and then simulating the policy without knowing the ground truth geometry of the articulated object. We do this by constructing a digital twin of the object from the demonstrations and simulating actions on the digital twin.

1 Introduction

We aim to construct a digital twin of an articulated object solely from demonstrations collected in the form of RGBD images. Specifically, we aim at learning the kinematic structure of the object. As an application of this, we aim to show that we can simulate a policy trained to manipulate articulated objects without needing the ground truth geometry of the objects.

For constructing the articulated object, we rely on advances in computer vision for novel view synthesis. Radiance field representations have demonstrated SOTA performance in capturing the visual fidelity of scenes. Gaussian Splatting [1] has shown that this representation can be trained using only geometry primitives, removing the need for a neural network to learn the scene. Many recent works have extended this to learning the geometry of dynamic scenes. Specifically, we build on the approach of , which tracks the transformation of a set of Gaussian particles over timesteps. We show that persistent tracking of Gaussian particles is sufficient to learn the object segments of an articulated object, as well as determining the joint location of the object. Through this, we can construct a digital twin of the articulated object and simulate the kinematics of the object.

For simulating a policy that is learned through demonstration, we train a behavior cloning policy using tele-operated human demonstrations. After training, we execute the policy swapping in the constructed digital twin in the place of the ground truth object.

Our main contribution is an approach to modeling articulated objects that requires no prior demonstrations, other than the object in the demonstration.

2 Related Works

2.1 Modeling Articulated Objects

Works like [2, 3] fall under a class of approaches that learns the geometry of articulated object from 2 partial point clouds (depth image), representing the start and end configurations of the object. These works are trained on a dataset of articulated objects, in which inference is to predict the articulated object's geometry and joint.

More similar to our approach, [4] models the articulated object using only the video sequence of the object being manipulating. While their approach only segments the object and predicts the joint in the 2D plane of the inputted image, our approach aims at building a digital twin of the object.

3 Method

3.1 Background

In our approach, we employ a 3D Gaussian Splatting (3DGS) representation [1]. 3DGS models the 3D geometry and visual fidelity of a scene through a set of images with known camera extrinsics. Key to this approach is treating the scene as a collection of 3D Gaussians. Each Gaussian takes up a portion of space in the representation, in which it can be interpreted as an ellipsoid where the covariance matrix defines the shape. A color and density value is associated with each Gaussian in the scene. To fit and populate the Gaussians to the underlying geometry of the scene through gradient based optimization. This is possible because the rendering process is fully differentiable. A key insight made by [5] is that 3DGS can be thought of as a particle based representation of the scene geometry. Then to extend 3DGS to dynamic scenes, this can be thought of learning the transformation/deformation of the particles over future timesteps. To learn the deformation/geometric changes of a moving object/figure, they first learn a 3DGS representation on the first time step. And for each successive time step, they learn the transformation of the Gaussian particles from the previous time step to the next.

3.2 Task Formulation

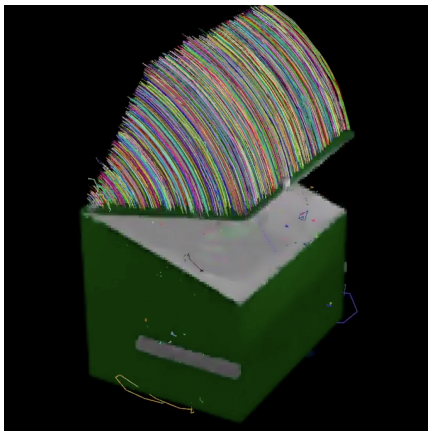
For constructing a digital twin of the articulated object, we divide human demonstrations into 2 groupings. One is purely for interacting with the object, ie. play data, in order to move the articulated object into different joint configurations. This set of interactions is used to construct the digital twin. The second set of demonstrations is intended to manipulate the articulated object to achieve a specific task, ie. closing a door.

For the interaction based demonstrations, we assume they have unlimited access to a range of camera views, along with segmentation masks of the object. While the human demonstrations may manipulate the object to reach different joint configurations, we only consider images of the demonstration that has no occlusion of the object.

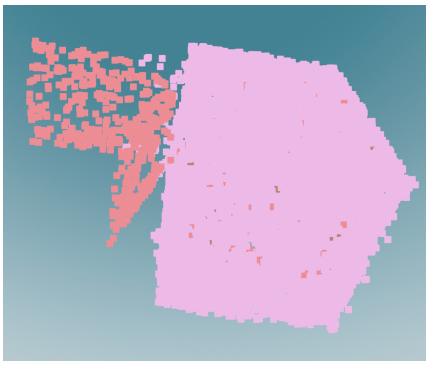
The set of collected images from these interactions are then fed into dynamic 3D Gaussian Splatting for training, until a persistent tracking of the Gaussian particles between all time steps is modeled.

3.3 Object Segmentation

Once the object has been modelled by the 3DGS representation, the next step would be to segment the object to separate the parts of the object that articulate with the ones that are stationary. For simplicity, lets assume that the object has only one articulating joint (multi-joint objects are explored in this paper). We can also assume each Gaussian on the surface of the to be a point corresponding to the center of the Gaussian, essentially simplifying the problem to segmentation of points clouds. Another important assumption is that the Gaussian's on the surface of the object remain at the same location with respect to the object surface through time. The idea behind segmentation is that the Guassians which maintain a similar distance amongst themselves can be considered as part of the same segment of the object. Let's consider a door of a microwave, all "points" on the door will maintain the same distance amongst themselves through time as the door is opened/closed. Similarly, all points on the body of the microwave follow the same pattern. However, as the door is opened/closed, the points on the door, will have varying distances to the points on the body of the object through time. This can be seen in 1 image (b) where the point clouds of the door open and closer are overlayed over each other. Using this as a heuristic, we can sum up the absolute distance between points and apply some form of clustering algorithm on this. A great way to visualise this heuristic can be seen in image (a) of 1.



(a) Articulation of trash can lid with respect to rest of the body.



(b) Point clouds of initial and final timesteps overlayed on each other.

Figure 1: Images to help visualise object segmentation.

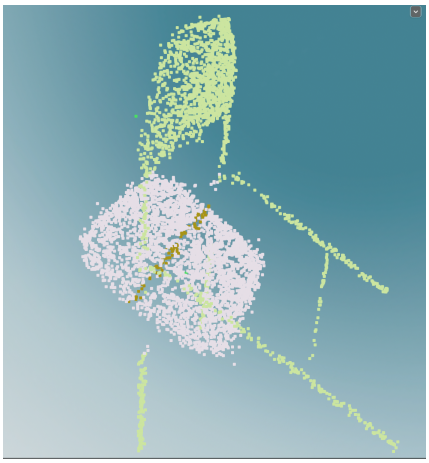


Figure 2: Points segmented as joint axis can be seen in brown for this folding chair.

In our approach, there were a few engineering tricks needed to make this work. Considering that we are finding the distance between every point to every other point, this is quadratic in computation cost in terms of the number of points to be used. Thus, this had to be thresholded arbitrarily to avoid memory overflow errors. In our attempt, we experimented with two different segmentation algorithms, K-Means and Hierarchical-DBSCAN (HDBSCAN).

3.4 Joint prediction

With the segmented object parts, we can now predict the location of the joint. The approach at a high level is that the points on axis of rotation will not move as the demonstrator articulates the joint. Essentially, following the equation $m \cdot v = v$ where m is the rotation matrix, and v are the set of points. This can be seen in the Figure .... To achieve this, we only need to use the segmented points that belong to the part of the object that articulates, and recover the rotation matrix. This can be done by comparing the points through time-steps and recovering the transformation matrix through a flavour of Iterative Closest Points (ICP). We implement vanilla ICP, to get a homogeneous transformation matrix $[\mathbf{R}|T]$, from which the rotation matrix $\mathbf{R}$ can be extracted. In order to find the set of points v, we start at the initial time step, and apply the transformation matrix to the segmented points to predict the articulation. The predicted points are compared to the final times step, the points that satisfy the equation mentioned above i.e. the points that have remain stationary

through the transformation are said to be part of the joint. To implement this, a few engineering trick are used again. To compare the articulated predicted point cloud with the segmentation from the actual time step, a simple euclidean distance is used between the points. Since we only applied the transformation matrix to generate our predicted, it is unlikely to find points that satisfy our equation. Thus, we use two different ways to identify the joint. First, use a top-k approach and take the top-k values that have the minimum distance between the two point clouds. The second approach which performs similarly is to use a threshold to select for instance points which moved $t\%$ compared to the max distance, where t is a hyperparameter to be tuned. Figure 2 shows an example of the points classified to be on the joint axis as brown on the seat of the folding chair.

Once the point within the segmented point cloud that belong to the joint are identified, we can now interpolate the rotation axis of the joint by a straightforward Linear Regression approach. However, since our segmentation is noisy, we implement RANSAC [6] to extract the equation of the line that depicts the rotation axis.

3.5 Behavior Cloning

Since our goal is to train a policy to interact with a digital twin of the object, a robot policy is first trained on a sample object using human demonstrations using some flavour of B Behavior Cloning (BC). Using the information gathered, we can now create a digital twin of the object, which can be exported to the robot simulation environment. More about this in Section 4.

4 Experiments/Results

4.1 Policy Setup

Let the policy objective be to manipulate articulated object consisting of a single rotational joint. A successful manipulation is when the object's joint is moved from it's starting position, to a joint position that rotated by 50 percent or more towards it's maximum range of rotation. The object be a covered box, with a single hinge towards the top. Therefore, opening the box hinge by 50 of its total rotation is considered a success. We use,, as our choice of simulation, and we we teleoperate a () arm to produce roughly 300 successful task demonstrations. Using Robomimic [7], we train a behavior cloning model, identical to the BC-RNN model discussed in [7] with identical hyperparameters as they used for their Lift task. See Figure 4 for a picture of the environment setup.

4.2 Digital Twin Setup

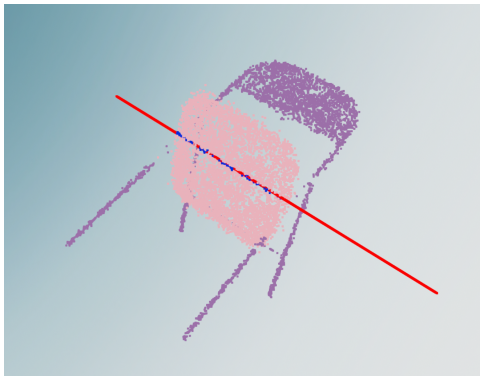
For our final digital twin setup, we place 60 different static camera views around the articulated object, uniformly sampled along a hemisphere. The articulated object is then set in 25 different joint configurations, which leaves for a total of 1500 RGBD images of the articulated object. This data is then fed into the dynamic 3DGS, and our digital twin construction process begins.

For most of our evaluation on the articulated objects, we focused on single joint examples, but also evaluating our clustering approach on multi-joint objects such as cabinets and a pair of glasses.

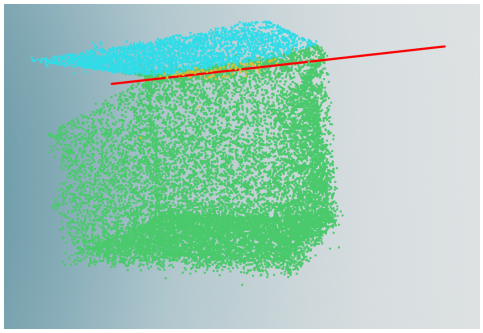
4.3 Current Progress on Policy

Some hiccups were encountered when setting up a custom environment in Robosuite [8]. The first issue is that the intended dataset of articulated objects we have been experimenting with SAPIEN [9], is not easily ported into Robosuite's Mujoco simulation setup. Because of time constraints we opted for creating an environment using a prebuilt articulated object asset provided by Robosuite.

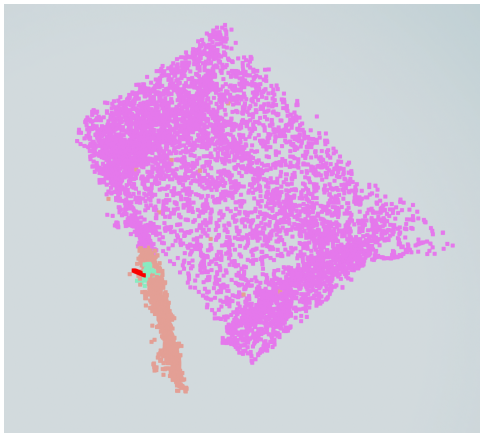
After training the BC model using the hyperparamaters as used in Robomimic, even evaluating policy on the ground truth articulated object geometry, the policy achieved a success rate of 0. We found that our policy often had the robot go towards the hinged object, but stop right before picking



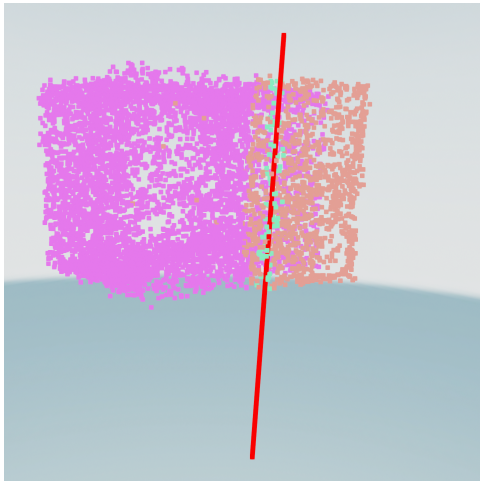
(a) Rotation axis for folding chair shown as red line.



(b) Rotation axis for lid of a trash can shown as red line.



(c) View 1 of inaccurate rotation axis for microwave door.



(d) View 2 of inaccurate rotation axis for microwave door.

(e) Failed attempt of K-means clustering 3 segments.

(f) HDBSCAN dropping particles while clustering.

Figure 3: Current progress on creating a digital twin of an articulated object.

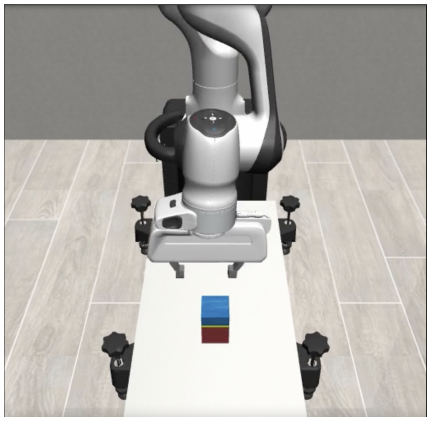


Figure 4: Picture of the task environment setup with the articulated object in Robosuite [8]

the object up, and stayed there. If given more time, we would like to do a hyperparameter sweep until we find a set of parameters that achieves a non-zero success rate.

4.4 Current Progress on Digital Twin

We had success with objects that have a defined shape, thus making segmentation easier, and only one joint to rotate about. These successes can be seen in Figures 3 images (a) rotation of axis for a folding chair and (b) rotation axis for the lid of a trash can. While these appear well, we found that our clustering approach is quite brittle depending on several factors encountered for a given object. For instance, the lack of particles near the joint can cause poor segmentation, as seen in the examples for the microwave in sub figures (c) and (d). Objects with more occlusion around the center of articulation/joint tends to result in poorer augmentations. Furthermore, we find that our clustering approach doesn't easily extend to multi-join objects. Subfigrue (e) shows the failed attempt of K-means to cluster 3 segments (one for each door of cabinet, and one for the body) but is only able to find two. Meanwhile, subfigure (f) shows that HDBSCAN [10] completely fails to cluster the second door of the cabinet. However, for an object like a folding chair, the articulated joint is clearly visible, with zero occlusion, and so our method successfully learns the object segmentation and also the center of rotation of the folding chair.

We planned on creating a URDF asset of the folding chair object and importing it into the Robosuite simulation. However, because of time constraints and lack of familiarity with Mujoco, we were unable to achieve this in the time allotted to us.

5 Conclusion

5.1 Policy

Initially, we thought that the task setup with the hinged object was very similar to the tasks investigated in Robomimic, and so we assumed using their hyperparameters would result in decent results. This was not the case in practice, and we did not have enough time to do a proper hyper parameter sweep on the model we employed to achieve even partially decent results.

5.2 Digital Twin

Based on our current results, it seems that a Gaussian Splatting based approach to modeling articulated objects is promising, but our current method is performing quite poorly. We believe our approach is inherently limiting. We believe a more principled approach would be to modify the

training process of the Gaussian Splatting in order to determine the object segmentations and the joint location as a byproduct of the gradient optimization process.

References

[1] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. 2023.

[2] Z. Jiang, C.-C. Hsu, and Y. Zhu. Ditto: Building digital twins of articulated objects from interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5616–5626, 2022.

[3] N. Heppert, M. Z. Irshad, S. Zakharov, K. Liu, R. A. Ambrus, J. Bohg, A. Valada, and T. Kollar. Carto: Category and joint agnostic reconstruction of articulated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21201–21210, 2023.

[4] N. Heppert, T. Migimatsu, B. Yi, C. Chen, and J. Bohg. Category-independent articulated object tracking with factor graphs. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3800–3807. IEEE, 2022.

[5] J. Luiten, G. Kopanas, B. Leibe, and D. Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*, 2023.

[6] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 (6):381–395, 1981.

[7] M. Beliaev, A. Shih, S. Ermon, D. Sadigh, and R. Pedarsani. Imitation learning by estimating expertise of demonstrators. In *International Conference on Machine Learning*, pages 1732–1748. PMLR, 2022.

[8] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, S. Nasiriany, and Y. Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.

[9] X. Zhang, R. Chen, A. Li, F. Xiang, Y. Qin, J. Gu, Z. Ling, M. Liu, P. Zeng, S. Han, Z. Huang, T. Mu, J. Xu, and H. Su. Close the optical sensing domain gap by physics-grounded active stereo sensor simulation. *IEEE Transactions on Robotics*, pages 1–19, 2023. doi:10.1109/TRO.2023.3235591.

[10] L. McInnes, J. Healy, S. Astels, et al. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205, 2017.