Model simplification for deformable object manipulation

Rahul Kashyap Swayampakula, Zixuan Huang, Dmitry Berenson
{rahulswa, zixuanh, dmitryb}@umich.edu

***Abstract*—This is a detailed summary of directed study on Model simplification for deformable object manipulation. The work mainly focuses on developing a simplified models for representation of deformable objects to achieve better planning performance for manipulation. During the course of this study we implemented few existing baselines algorithms and investigated possible directions to extend them to improve planning efficiency.**

I. Introduction

Reduced order modelling (ROMs), is a technique to simplify a high-fidelity mathematical model by reducing its computational complexity while preserving the dominant behavior of the complex model. The ROMs are aimed to offer real-time computational performance for high fidelity models. This concept of ROMs are used in many applications such as electronics, fluid dynamics, structural dynamics, design optimization, etc.

In case of deformable object manipulation, the object modeling plays an important role. Deformable objects are usually modeled as mass spring systems, with each mass point (which we refer to as a particle in this work) also representing a possible pick point for the robot. The high-fidelity models of deformable objects often contain thousands of such particles, making planning extremely computationally expensive. This gives us the primary motivation for looking into generation of ROM for deformable object manipulation.

There are some recent works rolled out in simplifying the particle based representation of the deformable objects using different approaches. Through this directed study, we looked into several existeing methods of model simplification procedures, their implementation, limitations and the potential extensions which can in increasing the planning performance while manipulation

II. Baseline

For this project, the baseline implementation is considered as replicating the paper Goal conditioned model simplification for deformable object manipulation [1]. This paper uses the goal configuration to perform the model simplification and generates the keypoint from the simplified model to represent the current state. The key point is defined as the points whose linear interpolation is sufficient to fit a complete model. To generate keypoints, the method uses the error between the linear model obtained from keypoints and the original (goal model in this case) as the metric to determine the desired key points to simplify the model. We iterate through the number of keypoints and calculate the error in the fit. The loop will stop when the error in the fit is less than the threshold (hyperparameter) as shown below (Algorithm 1).

Algorithm 1: Model simplification process

Input: M_O, ξ_G
Output: M_S, $\hat{V}_O$
$N_S = 2$;
do
  $M_S, \xi_G^S \leftarrow \texttt{Simplify}(M_O, \xi_G^O, N_S)$;
  $error \leftarrow \texttt{Error}(\xi_G^S, \xi_G^O)$;
  $N_S \leftarrow N_S + 1$;
while *error* > *threshold*;
$\hat{V}_O \leftarrow \texttt{Extract}(M_S, \xi_G^S, M_O, \xi_G^O)$

Fig. 1: Goal-conditioned model simplification

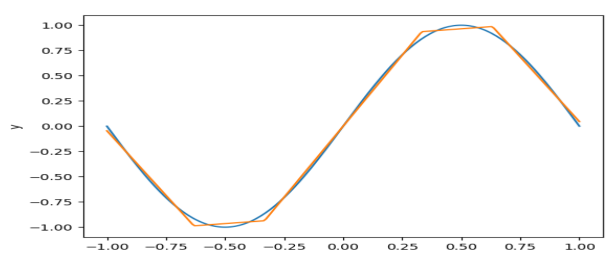

Fig. 2: Linear fitting for a Non-linear curve

In the process of replication of the baseline paper the key feature extraction is the main part. For 1D deformable object the key point extraction is done using the existing python libraries. For 2D deformable objects, it is done using Quadric edge collapse decimation (QECD) from Pymeshlab [2]. Once these key points are extracted, the planning is performed using CEM based planner to reach the goal state.

A. Evaluation

To evaluate the performace planning using the simplified model, we have used Softgym [3] based simlation to check the completion of the task. During this process, CEM based planner is used in softgym simulator for planning task in the reduced action space. The evaluation is done based on whether the planner is able to converge to the goal state with a cost of less than 0.001. This performance has been verified across

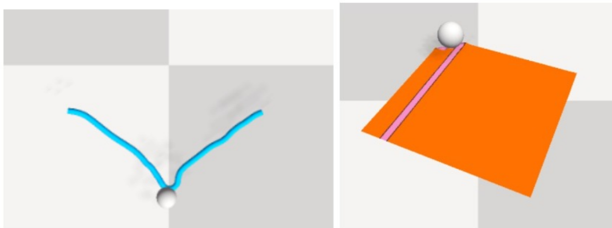

Fig. 3: Manipulation on simplified models

various datasets across 1D and 2D deformables. For 1D deformables, the generated configurations are: Straight Rope, V-shape rope, C-shape rope, M-shape rope. For 2D deformables, generated configurations are diagonal fold, straightening cloth, side fold. The experiments are designed to reach one configuration from another among the above generated set of configurations.

B. Analysis

After implementing the baseline, we have seen that this method is performing well in few cases. The keypoints results in a reduced action space to perform the planning for manipulation task. This reduced action space will take less computation time compared to the high fidelity model where the action space is significantly large. The performance is faster and better in case of the experiments of Straightening, making a V-shape rope etc.

However, the method fails on significantly large set of cases. Like converting V-shape to C-shape, folding a cloth at random location, etc. Also the baseline method have a reduced action space which generate a discrete set of points, so one important aspect one need to consider is the consistency of picking action. It has been observed in this case that the picking action is arbitrarily selected between two actions. This makes the jump between the points of action and makes the manipulation unrealistic and inconsistent. This has led to make the manipulation fail for some fairly simple cases as mentioned above. This inconsistency in pick point also make the algorithm impossible to reach to complex configurations. Because this inconsistency can lead to a much more complex configuration where the calculated keypoints may not be sufficient for manipulation action.

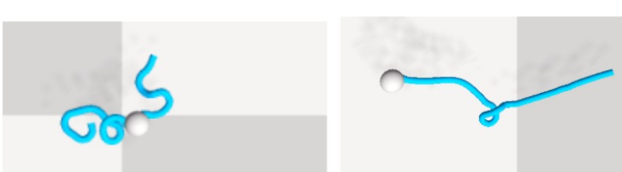

Fig. 4: Failure cases of Manipulation using baseline

C. Extension

As an intermediate goal of the project, we have worked on improving the performance of the baseline without making major changes to algorithm. The major problems we faced are inconsistency in pick points and not able to cover multiple configurations. So, to tackle the coverage of multiple configuration we have generated the keypoints using multiple configuration rather than only goal configuration. This is because the goal-conditioned simplificaiton is only capturing the nonlinearities in the final configuration but not the change of non-linearities between the initial and final configuration. This method didn't improve the performance because it is observed in many cases additional keypoints are not useful for the planning tasks. It didn't help in reaching complex configurations.

The major failure is because of the inconsistency in the picking action. To tackle this we have performed a smooth change of picking action. We have reduced the loss over a n number of actions keeping the pickpoint fixed. Whichever pickpoint has least cost over the set of action that has been considered as best pickpoint and the action is performed over that point. After n actions, the pickpoint will be chosen again based on the above method and keeps repeating. Eventhough the method is able to restrict the fast manipulation switches happening between the action steps, it doesn't improve the behaviour of the baseline. So, we started looking at better heuristics which can led to approximate the deformable objects representation for planning.

III. Topology aware simplification

As the linear simplification method is not able to perform the simplification for all the desired configurations. For example, if we consider a case of straightening a rope, with its initial state as knotted halfway. The above algorithm ignores the details of the knot and tries to straighten the rope because the simplification is purely done on the desired configuration. And this can result in a knot which is not desirable. So, to get a better simplified model can we use alternate approaches for simplification. So we aim to look into Topology based approach where we can use topology features to track and perform the simplification of model.

The idea is inspired by *sequential persistent homology* (seqPH) [4] where the authors have proposed a topology feature based tracking approach for deformable objects. The key idea is to generate the topology features for the deformable object in its observable state and simplify the model based on these features. The main motivation behind using the topology features is that they can be even used to track the 3D deformations. Before going to simplification procedure we wanted to check what are the extracted features using topology and its robustness.

The idea of extracting topological features is based on Persistent homology. It is a very widely used concept in applied topology. More details of Persistent Homology can be found in seqPH paper. After performing the PH operation this will help us identifying the connections, loops and voids in a data. So we want to identify these features and perform simplification procedure further. For this the code was written using RIPSER library in Julia [5] [6]. The point cloud data generated from meshes like garments, T-Shirts, cloths, etc are used to observe the topological features.

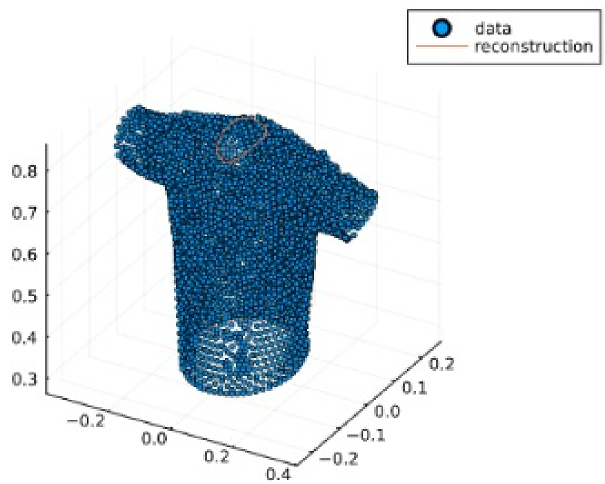

Fig. 5: Extracted topological features using Persistent homology

A. Analysis

The PH based features are able to identify the loop, voids in the given 3D mesh. By identifying these features we can perform planning using these points and complete a manipulation task. But these features are tough to track and not robust against variations. Also the PH based topology features doesn't generate any feature in case of 1D or 2D deformable object configurations. So this makes the topology based heuristic not useful for model simplification.

IV. Other methods

We have also looked into other methods of mesh simplification or remeshing schemes based on Meshgraphnets [7] and Meshlab/blender. [8] [9]

A. Meshlab & Blender

The Meshlab and blender are common mesh visualization tool. They have many inbuilt remeshing strategies and mesh simplification algorithms. So we have tried them to generate a simplified mesh while preserving the properties of the mesh. The algorithms are mainly a variations of QECD and more robust across various types of meshes. The resulting simplified mesh has good simplification but failed in the dynamics simulation. The dynamics simulation is really important because that helps in predicting the rollouts for the planning. So these simplifications are eliminated in our use case.

B. Meshgraphnets

The Meshgraphnets is one of the poweful adaptive remeshing strategy in the computer graphics. The paper is based on Anisotropic Adaptive remshing algorithm. Where the key remeshing algorithm takes place using sizing fields. The sizing field is a vector defined at each point on the mesh, which helps in defining the local property of the mesh. Using a sizing field based operator the edges in the mesh are generated. The Meshgraphnets paper is able to show the dynamics simulation performed on this simplified mesh. Using the trained wights we were able to generate the result for one of the shown rollouts of simulation. This simplification procedure looks promising but we need sometime to investigate into details of it.

V. Conclusion

The model simplification for deformable object manipulation is a non-trivial task. Although there are many methods of generating ROMs for various meshes the problem of deformable object maipulation is generating simplified model while changing the configuration unlike representing simplified dynamics. The baseline and topology based simplification procedures are purely based on geometry. On the other hand, Meshgraphnets generates adaptive simplification based on the given action. In our case we need a simplified mesh where we can perform actions and predict rollouts. There should be a function that should be able to corelate between dynamics and geometry. So, as a future direction it would be interesting to develop a formulation to connect geometry and dynamics. One way to do this is to learn a policy based on the sizing fields to generate the more sparse mesh for planning tasks.

Results are available at : Link

References

[1] S. Wang, R. Papallas, M. Leonetti, and M. Dogar, "Goal-conditioned action space reduction for deformable object manipulation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[2] A. Muntoni and P. Cignoni, "PyMeshLab," Jan. 2021.

[3] X. Lin, Y. Wang, J. Olkin, and D. Held, "Softgym: Benchmarking deep reinforcement learning for deformable object manipulation," in *Conference on Robot Learning*, 2020.

[4] R. Antonova, A. Varava, P. Shi, J. F. Carvalho, and D. Kragic, "Sequential topological representations for predictive models of deformable objects," in *Proceedings of Machine Learning Research*, 2020.

[5] M. Čufar, "Ripserer.jl: flexible and efficient persistent homology computation in julia," *Journal of Open Source Software*, vol. 5, no. 54, p. 2614, 2020. [Online]. Available: https://doi.org/10.21105/joss.02614

[6] M. Čufar and Žiga Virk, "Fast computation of persistent homology representatives with involuted persistent homology," 2021. [Online]. Available: https://arxiv.org/abs/2105.03629

[7] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, and P. W. Battaglia, "Learning mesh-based simulation with graph networks," in *International Conference on Learning Representations*, 2021.

[8] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: http://www.blender.org

[9] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, "MeshLab: an Open-Source Mesh Processing Tool," in *Eurographics Italian Chapter Conference*, V. Scarano, R. D. Chiara, and U. Erra, Eds. The Eurographics Association, 2008.