ROB 590 Project
Visuo-Tactile Dynamics and State-Estimation for a Deformable Tactile Sensor

Sarvesh Mayilvahanan
smayil@umich.edu

I. Introduction

This project attempts to use a neural implicit representation for deformable object state-estimation, specifically the Soft Bubble Gripper [1]. This was accomplished through recent advances in this task, VIRDO [2] and VIRDO++ [3].

Modelling the dynamics and having functional state-estimation for a deformable tactile sensor is important for general understanding of the sensor and making better grasping decisions as well as a number of downstream tasks. However, creating a model to accomplish this is challenging due to the complexity of the membrane dynamics as well as a lack of data. This study demonstrates that the structure of VIRDO++ is able to capture this model through a neural implicit representation using visual (pointcloud) and tactile (wrench) data.

II. Implicit Representation of Membrane Dynamics

In this work, we explore two different approaches to representing membrane geometry and dynamics: VIRDO and VIRDO++. VIRDO directly encodes the membrane's contact formation observed from the bubble sensor's depth map through a PointNet architecture to represent deformed membrane geometry. VIRDO provides latent states of the membrane useful for downstream tasks. VIRDO++ extends VIRDO to represent the dynamics of deformable objects for downstream planning and control tasks. To achieve this, it adopts an Action Module that consumes the current latent state of the membrane and the action of end-effector.

Figure 1 shows the overall model architecture of VIRDO++, with four main modules: Force Module, Action Module, Deformation Module, and the Object Module.

A. Deformable Object Dynamics

VIRDO++ uses a latent object code α, force code z_t, and contact embedding c_t to represent the deformable object state, each encoding different information. The object code encodes relevant information about the specific object, which is then used to recover a signed-distance field. The force code encodes boundary conditions relevant to the object's current state. The contact embedding encodes relevant contact information, specific to an object's state.

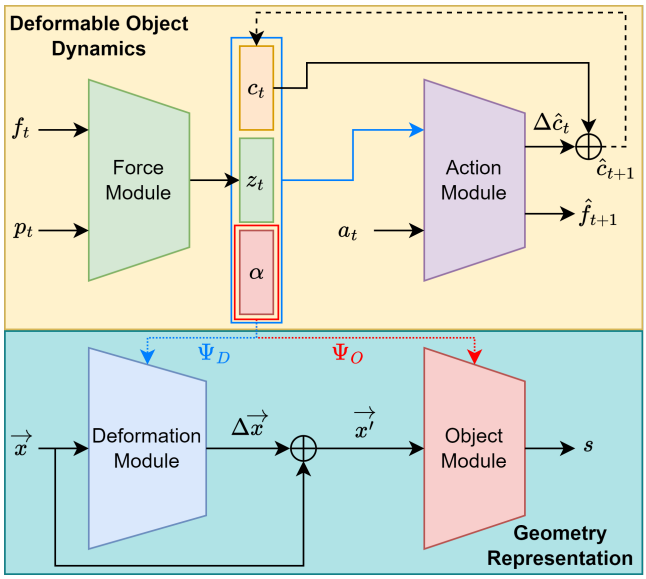

Fig. 1: Overview of VIRDO++ model architecture

The Force Module takes in a reaction wrench $f_t \in \mathbb{R}^6$ and the robot's end-effector pose $p_t \in \mathbb{R}^{d_p}$ and produces the force code z_t.

The Action Module takes in the force code z_t, object code α, contact embedding c_t, and the robot's end-effector action $a_t \in \mathbb{R}^{d_a}$ and predicts the reaction wrench after taking the action $\hat{f}_{t+1}$ as well as the change in the contact latent vector $\Delta\hat{c}_t \in \mathbb{R}^{d_c}$. This allows for a model rollout to predict future states based on a trajectory of actions.

B. Geometry Representation

The geometrical representation of the deformable object is handled by two separate modules.

The Deformation Module is a hyper-network, who's weights are conditioned on the object code α, force code z_t, and contact embedding c_t. The hyper network $\Psi_D(\alpha, z_t, c_t)$ predicts the weights for the Deformation Module, which takes in a set of query points $\vec{x} \in \mathbb{R}^3$ and predicts a point-wise deformation field $\Delta\vec{x}$, which when summed with the original query points results in the deformed object's nominal shape $\vec{x'}$.

The Object Module is also a hyper-network conditioned on the object code $\Psi_O(\alpha)$ and is a neural implicit representation of the signed-distance field of the nominal shape. It predicts a signed-distance s for each query point. This module, and the learnable object code are formulated similarly to [4], using an auto-decoder approach.

III. Data

Setting up the data for model training and inference was critical to achieving good performance in model prediction. In particular, this was challenging for this project because VIRDO and VIRDO++ are designed to handle representation of deformable objects held in an end-effector. However, for this project, we attempt to perform this same representation for an "object" that is the end-effector. Therefore, some choices had to be made in regards to handling the data such that it is still relevant to the formulation of VIRDO.

The data provided by the bubble sensors are depth images, which are filtered to reduce noise and then projected using camera intrinsics to obtain a pointcloud of the bubble surface. Due to the bubble surface being a surface and not an enclosed object as formulated in other neural SDF representations, here the signed-distance value for a point is corresponding to the signed distance of the point from the nominal surface along the camera axis. An alternative that was considered was to use a Three-Pole SDF [5] to represent the bubble surface because of its ability to capture open surfaces. This would require making the Object Module a classification head, slightly changing the way VIRDO is formulated. However, as further figures show, the approximated SDF values were good enough for training the Object Module and it is able to represent the bubble surface relatively well.

For training the VIRDO model, the contact patch Q is found by comparing the deformed depth image to a nominal reference depth image and filtering points with a depth imprint greater than a threshold value ε. $Q := \{p \in P \in \mathbb{R}^3 : |p - nom| > \varepsilon\}$

The dataset used for training and testing of the VIRDO and VIRDO++ models was a series of grasps by the Soft Bubble Grippers on a 10 mm diameter cylindrical rod. It consisted of 35 total trajectories, each of which having a length of 10 timesteps. 30 of the trajectories were used for training and the remainder were held out and used for testing.

IV. Results

A. VIRDO

To train the VIRDO model, a dataset was created as described above, and the left and right bubbles were treated as independent objects, each with their own object code.

Pretraining of the Object Module shows that the model is able to learn a good model of the nominal bubble surface shape as seen in Figure 2. The model is able to accurately predict the signed-distance field, although the predicted SDF has some issues at the surface due to a lack of ground-truth normals. This causes a somewhat "bumpy" predicted surface, as seen in the marching cubes reconstruction in Figure 3.

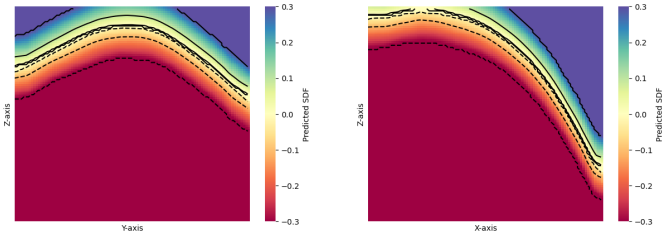

Fig. 2: Predicted SDF of bubbles for YZ-plane slice (left) and XZ-plane slice (right). Values are colored by clipped predicted SDF value.

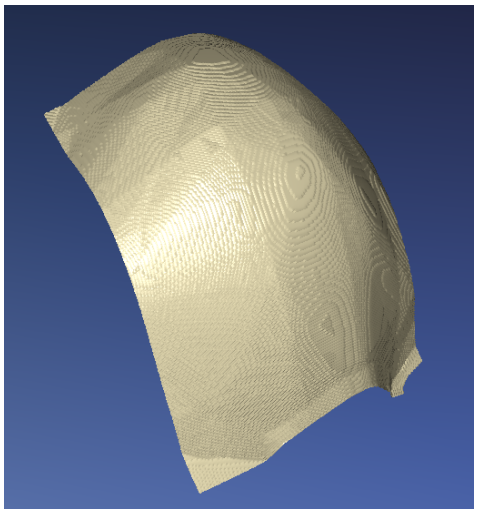

Fig. 3: Marching Cubes reconstruction of bubble surface using trained Object Module.

Additionally, the Deformation Module is able to learn the deformation field that corrects the deformed shape to the nominal shape. Over 10 trajectories, the mean chamfer distance (CD) to the nominal bubble point cloud is shown in Table I, indicating that the Deformation Module is able to predict the object deformation field.

TABLE I: Chamfer Distance to Nominal Shape

	Deformed	Undeformed
CD ($\times 10^3$)	70	1

B. VIRDO++

Before training the VIRDO++ model, the paper and available code first needed to be translated into a form that could be repeatably trained and evaluated. This took some time due to the differences between VIRDO and VIRDO++, but there is now a working version of VIRDO++ available here that can be used for anyone that wants to train a VIRDO++ model.

In order to train the model, a modified version of the VIRDO dataset was created that included all the additional inputs needed for the new Action Module. The data used

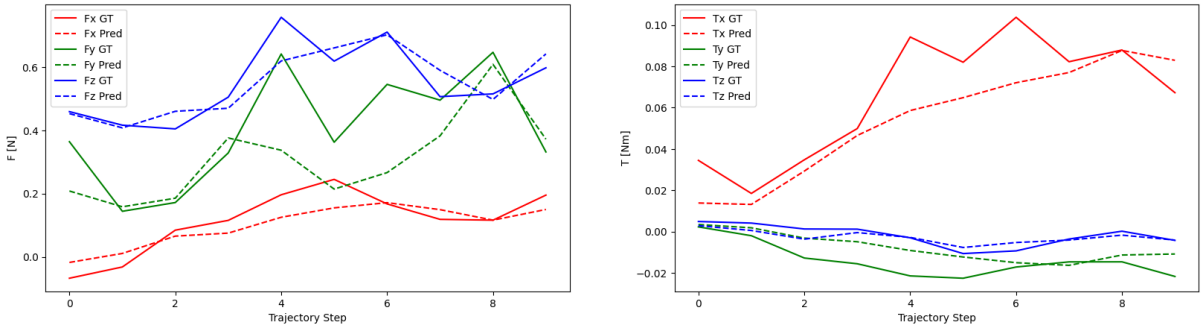

Fig. 4: Wrench predictions for a given trajectory. Ground truth and predicted values are shown for forces (left) and torques (right).

consists of trajectories of actions and measurements taken while using the Soft Bubble grippers to grasp a rod.

Pre-training follows an identical process to that of VIRDO, and almost identical results are produced as in Figure 2. While training the entire VIRDO++ model, we found that the model is able to accurately predict wrenches given the current state and action to be taken. Figure 4 demonstrates the predicted wrenches for a single trajectory. We observe that overall, the model performs well in making predictions and is able to follow the general trends of the wrist wrench.

Additionally, the trained VIRDO++ model is evaluated on a state estimation task, where the contact embedding is learned through a particle-filter like process (in the inference case, the contact embedding at the initial state would be unknown). Here, the n contact embedding samples are randomly initialized and propagated through the trained VIRDO++ model. Based on the calculated losses, backpropagation is performed to update the contact embeddings. Using the L1 loss between the predicted and ground truth wrench, weights are given to each contact embedding sample and used to re-sample with some added noise. The contact embedding with the highest weight is used for state estimation.

Figure 5 shows the results of this filtering-based state estimation for a single trajectory. We see that the model is able to learn how the deformations in the bubble surface change given only knowledge of the current state and the action to be taken. The model performs well in predicting how the deformation will change given knowledge of the current contact embedding and wrench because the Action Module is able to accurately use the action to predict the contact embedding and wrench prediction at the next timestep. This allows the Deformation and Object Modules to reconstruct the predicted deformed shape. The model performs particularly well for this task in the initial portion of the trajectory and is able to predict both the shape and location of the deformation well. However, it appears to not track the exact location of the deformation as well in the later parts of the trajectory, although it maintains the correct shape of the deformation. This could be due to the predicted contact embeddings starting to drift as the model goes through the trajectory, resulting in worse predictions for subsequent timesteps.

V. Conclusion

Overall, this project found that it is possible to model the behavior of the soft bubble gripper implicitly through a neural network and that VIRDO++ succeeds in capturing the dynamics of the bubble as well as its geometrical representation.

This work directly adapts VIRDO++ to represent membrane dynamics, however some modifications could be made for this specific task. Due to the bubble sensor directly providing a depth map, the true contact patch can be extracted, which is not true of many other objects. Therefore, an encoder structure could be used to encode the contact patch to a latent state instead of directly attempting to learn a contact embedding (similar to how it is handled in VIRDO).

Further work could be taken in constructing a dataset with a wider range of actions and objects to be gripped so that the trained model is more generalized to any type of deformation. This work only focused on a single dataset with fairly limited actions and an object with simple geometry. Additionally, comparisons to other methods of deformable object dynamics would help understand the strengths and weaknesses of this method.

References

[1] N. Kuppuswamy, A. Alspach, A. Uttamchandani, S. Creasey, T. Ikeda, and R. Tedrake, "Soft-bubble grippers for robust and perceptive manipulation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9917–9924.

[2] Y. Wi, P. Florence, A. Zeng, and N. Fazeli, "Virdo: Visio-tactile implicit representations of deformable objects," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 3583–3590.

[3] Y. Wi, A. Zeng, P. Florence, and N. Fazeli, "Virdo++: Real-world, visuo-tactile dynamics and perception of deformable objects," *arXiv preprint arXiv:2210.03701*, 2022.

[4] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 165–174.

[5] W. Chen, C. Lin, W. Li, and B. Yang, "3psdf: Three-pole signed distance function for learning surfaces with arbitrary topologies," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 522–18 531.

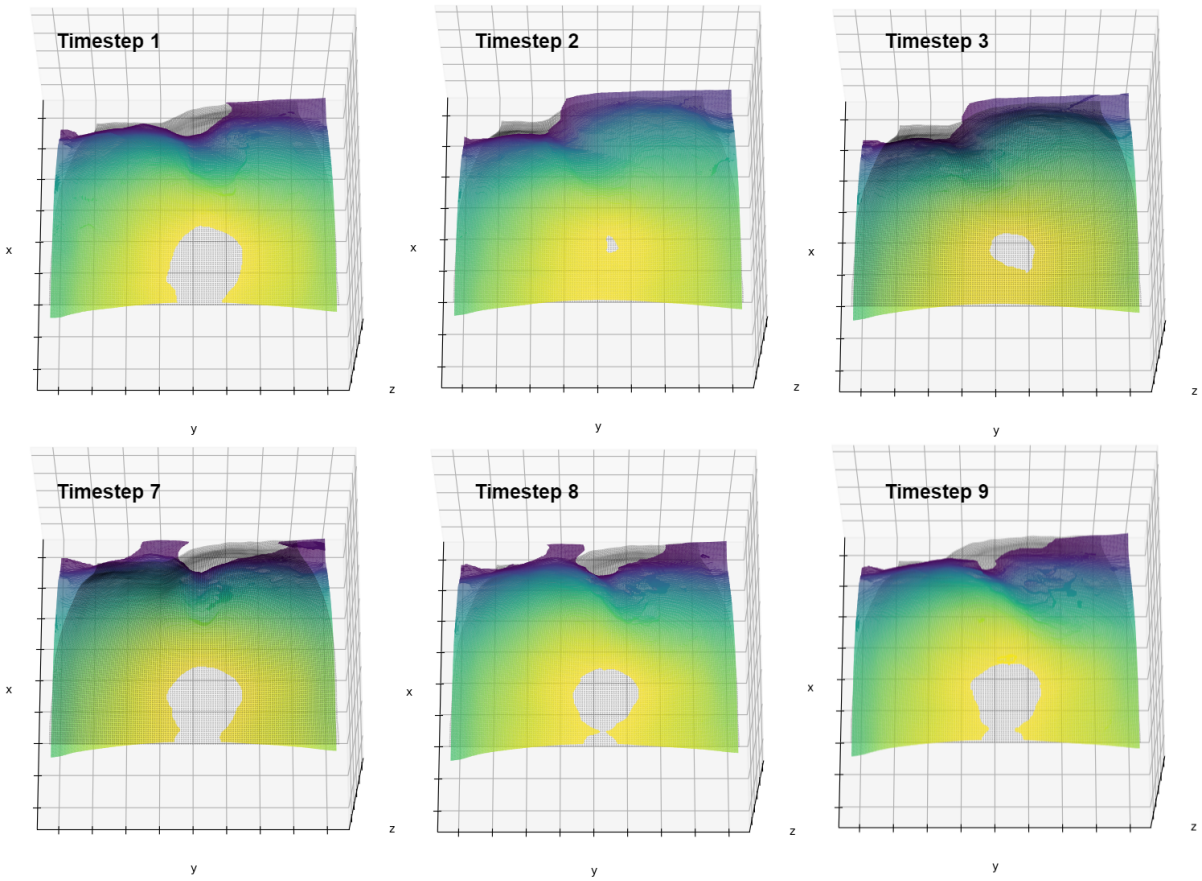

Fig. 5: State estimation using VIRDO++ for soft bubble. Predicted bubble surface mesh shown in color, ground truth deformed bubble at next timestep shown in black.