Fall 2021: Robot Bandaging Project

Raghava Uppuluri [1]

Abstract

The goal of course project is to explore vision-based techniques for non-rigid object manipulation for longer horizon tasks by a robot arm. To accomplish this, the robot arm will complete a robot bandaging task, where the robot will learn to wrap a bandange around an object shaped like a wrist (simplified to a cylinder). A key motiviating factor behind this project's goal is that many day to day tasks include manipulation of 2D deformable objects such as cloth, wraps, etc. Also, developing a general approach to state estimation of the object that is robust to changes in the environment (i.e occlusion) and training a policy that can be evaluated under many testing scenarios to provide benchmarks and relibility were important learning objectives for this project.

1. Problem

Bandaging is a common task done in the medical field that consists of wrapping a bandage around an appendage with tension for added stability or wound care. Specifically, the task itself involves long-horizon manipulation of a highly deformable bandage around a wrist multiple times with a fixed tension of the bandage.

2. Prior work

While the combined difficulty of manipulating a 2D deformable object which is more prone to self-occlusions than just square cloth

2.1. Cable Manipulation with a Tactile-Reactive Gripper

2.1.1. Overview

This work utilizes visuo-tactile sensor feedback to control a gripper/arm system to follow along a cable using an LQR controller to center cable within gripper fingertips, while using a PD controller to control the gripping force and therefore friction along the cable. (She et al., 2020)

2.2. Review

This approach in manipulating highly deformable objects does not need to generate a state representation of the cable, but just controls features of the cable as seen through the visuo-tactile gripper.

While this has been shown to be robust in local control scenarios, it has trouble generate an scene understanding that would be necessary to grasp the bandage initially and make regrasps during the task execution. Additionally, validation in simulation is tricky without an accurate visuo-tactile sensor model represented in simulation.

Although, such an approach may be used reliably as a local grasp planner for grasp adjustments along the bandage.

2.3. Learning Rope Manipulation Policies Using Dense Object Descriptors Trained on Synthetic Depth Data

2.3.1. Overview

This work applies work done in (Florence et al., 2018) to manipulating rope, a 1D highly-deformable object.

The main contributions include (i) showing synthetic depth and image data of rope can effectively train a correspondence model to output a dense representation of rope that generalizes to real images of rope and is interpretable due to a focus on its geometric structure, named dense depth object descriptors (DDODs), (ii) a geometric-based policy can utilize the dense representation to complete non-trivial rope manipulation tasks such as tying knots and moving ropes to new configurations specified in demonstration videos not seen during training.

2.3.2. Training Dense Descriptor Mapping Function using Synthetic Data

In order to extend the dense representation to rope, the simulated mesh representation consists of ”over fifty thousand ordered vertices of known global coordinates and an underlying Bezier curve with $M = 12$ control points, $P_1, ..., P_M$,” adding reliability of having more reliable ground truth data and is more easily accessible (Sundaresan et al., 2020). Then, depth images of the scene and RGB images are used to generate a point-pair configuration of the scene with the 3D representation.

To collect the training data, the simulated rope configuration, described by pairs of points on the rope and synthetic depth and RGB pixels, is perturbed randomly to create a new configuration, represented by T_1 to T_2. The corresponding points in T_1 T_2 are ”mapped into the descriptor space and encoraged to be close together” (Sundaresan et al., 2020). This mapping $f_{dense}(*)$ is learned using a Siamese network with pixelwise contrastive loss.

2.3.3. Geometry-based policy for Visual Imitation using Learned Descriptors

Given a goal demonstration video with RGB and depth, each configuration of the rope is mapped into descriptor space, where it can be possible to compare the goal configuration (final configuration in the demonstration) to the current configuration of the rope. Taking the K-nearest-neighbors for given descriptors on current rope configurations and goal configurations, the points in the descriptor space with the greatest L2 distance error are selected and an action is taken on the 3D point corresponding to descriptor point of the current rope, then minimizing the error. This process is repeated for the highest error between pairs of points in descriptor space until a goal threshold is reached.

2.4. Review

This paper provides a strong baseline for utilizing synthetic data for non-rigid object representation. Although, a key assumption that needed more explanation was that ground truth correspondences are difficult to obtain for a real rope. Given object classes that cannot be easily represented like a rope such as gauze or materials that wrap around other objects, it causes synthetic data representation to fall through. Additionally, the geometric policy was finetuned and engineered for the specific task, making it more difficult to adapt for other problem configurations. Further, to potentially provide a better representation of the lower-level features that may be difficult for just a vision system as shown by situations such as occlusion and sparsity in the descriptor space, multi-modal data collection such as high-dimensional tactile data that may be mapped into the descriptor space may have potential.

3. Current Approach

From discussions of the prior work, gaps still exist in addressing a long-horizon, highly-deformable object manipulation problem such as resiliance to occlusions, having a policy learner that is not structurally engineered for a task, and using multimodal inputs.

From a systems perspective, the current approach fuses the prior works ((Florence et al., 2018),(She et al., 2020), and (Sundaresan et al., 2020)) to achieve bandage state estimation, high-level policy training using imitation learning, and local bandage control policy using visuo-tactile sensor.

This semester, bandage state estimation is being focused on, while setting up environments and training procedures for the high-level policy training and local control policy development.

Implementing the presented task will involve developing methods to robustly estimate the state of the non-rigid object which be observations for learning a imitiation learning policy from human demonstrations. For robust state estimation, results of (Florence et al., 2018) will be extended through the addition of a custom fabric dataset that is curated in the simulator to increase the diversity of the dataset.

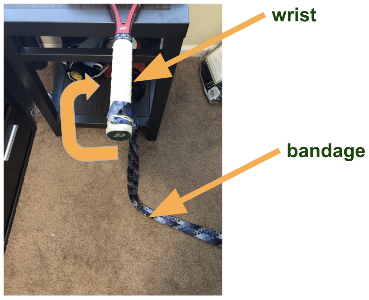

Figure 1. Preliminary experimental setup of the bandaging task

4. Checkpoints

4.1. Bandage Object State Estimation (this semester)

1. Generate a bandage dataset
2. Learn a dense correspondence mapping on the bandage dataset
3. Evaluate using pixel match error
4. Based on results, integrate visuo-tactile state estimation into network, or use point cloud representations, or utilize alternative latent representation

4.2. High-level Policy Learning

1. With the latent representation of bandage, collect training trajectories of successful task completions
2. Using human goal trajectories, train DAGGER imitation learning model and alternative models that remove distribution shift problem to create baselines

4.3. Local Control Policy

1. Reproduce results from (She et al., 2020) to have gripper follow bandage and control gripping force and therefore friction/tension of bandage

2. Integrate with high-level policy

5. Implementation

5.1. Curating the Custom Bandage Dataset

To represent the bandage, the bandage was represented as connected points that can each move freely in space, and each point in R^3 space on the fabric is mapped to pixel space to generate a set of pixels corresponding to each point on the bandage. Then, a unique skin was applied to the bandage that resulted in a change in appearance. A total of 7 skins were used and around 50 points were used to represent the bandage.

In order to reach the diversity in the dataset, the state of the cloth was randomized by initializing each point on the cloth by a *crumple factor* that is randomized. Then, after running the simulation until the cloth is at rest, the cloth is placed in a random position. In a similar fashion, the location of the camera was varied across each iteration. A sample is exemplified in Figure 2.

Developing the simulation was a result of generating synthetic representation of the fabric in Blender utilizing approaches by (Sundaresan et al., 2020) which used Blender for generating images of rope and then utilized domain randomization techniques for an efficent object representation that transfered well into the real world.

Then, a gym environment was created for the PyBullet environment that initialized the robot intrinsic vectors for representating the robot state, although state transitions (actions) were not implemented yet as the human controller is in development that would make it easier to test state transitions. Finally, generating a basic fabric reprentation was generated within pybullet and the gym environment.

As will be mentioned in the results, the PyBullet environment had difficulty in representing thin cloths and also a representation such that it was easy to change the cloth skin and visual, therefore the simulation enviornment was then ported to Mujoco as shown in Figure 2, where it became a lot easier to change the dimensions and initial position of the cloth without breaking the simulation (Todorov et al., 2012). Additionally, the simulation environment in Mujoco did a much better job in simulation of the cloth dynamics and the contact forces between the gripper and the cloth.

In total, 1000 unique cloths states were generated each segmented and domain-randomized using 5 different scenes. Other preprocessing technqiues include random rotations, and lighting changes.

5.2. Training Details

Introduced in (Florence et al., 2018),

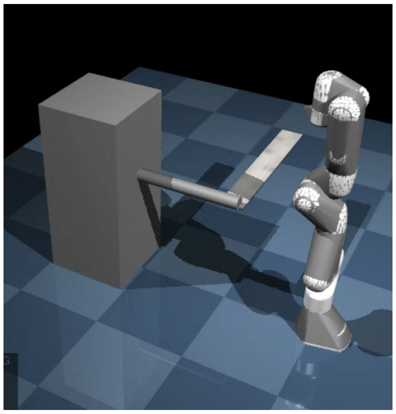

Figure 2. Preliminary simulated setup in MuJoCo of the bandaging task with the fabric-like object representing the bandage, a Franka Emika Panda robot, and a cylindrical wrist setup for simplicity

Described by pairs of points on the cloth and synthetic depth and RGB pixels, the cloth is perturbed randomly to create a new configuration, represented by T_1 to T_2. The corresponding points in T_1 T_2 are "mapped into the descriptor space and encoraged to be close together". This mapping $f_{dense}(*)$ is learned using a Siamese network with pixelwise contrastive loss, which was just shown.

Given that we aim to have the fabric representation be robust to many different views, colors, occlusion states, etc, we use the following loss function to evaluate the similarity of the fabrics. The similarity is represented by a match m and a non-match nm, which is applied to each pixel between two randomly selected images from the dataset. The match and non-match are propgated through the 3D representation of the fabric into pixel space, allowing for comparision directly in pixel space that is used at both training and test time, similar to how the robot would operate. Additionally, we choose a margin M hyperparameter that represents the buffer between a match and non-match in the latent representation outputted by $f_dense(*)$.

$$L_m(I_a, I_b) = \frac{1}{N_m} \sum_m D(I_a, u_a, I_b, u_b)^2$$

$$L_{nm}(I_a, I_b) = \frac{1}{N_{nm}} \sum_{nm} max(0, M - D(I_a, u_a, I_b, u_b)^2)$$

$$L(*) = L_{nm} + L_m$$

It should be noted that all fabrics are the same geometrically and similar visually, therefore the loss function reflects this by ensuring the points in 3D space are "close" in the network's latent representation of the object. If we define $f(*)$ as the network's mapping function from pixel space to its latent space, then each pixel mapping should be close together. This is validated by using L2 loss across the network's output.

Further, to further segment the latent space for each object, cross-object data representations were used to generate matches and non-matches.

For training the network, a pretrained 34-layer and stride-8 ResNet was used as backbone and then bilinearly upsampled until it reached the original image shape of 240 x 320 and has channel size D, the size of the descriptor space. Alternative methods that were intially trialed were using an encoder latent space representation using 3 convolution layers, each with pooling layers.

6. Results

6.1. Qualitative Evaluation

From Figure 2 (input visualization), Figure 3 (output descriptor visualization with D=3), and Figure 4 (output descriptor visualization with D=15), we can visually see the effectiveness of learning a descriptor representation. The relatively poor performance can be attributed to the reduced density of matches per pair of images. Additionally, performance can be improved by sampling non-matches from non-corresponding matches so that there is some separation between the projected points themselves. Also, performance decreased with a smaller descriptor space, which makes sense as large descriptor spaces should be able to encode more information about the object, although this has an added computational cost as it is very high dimensional.

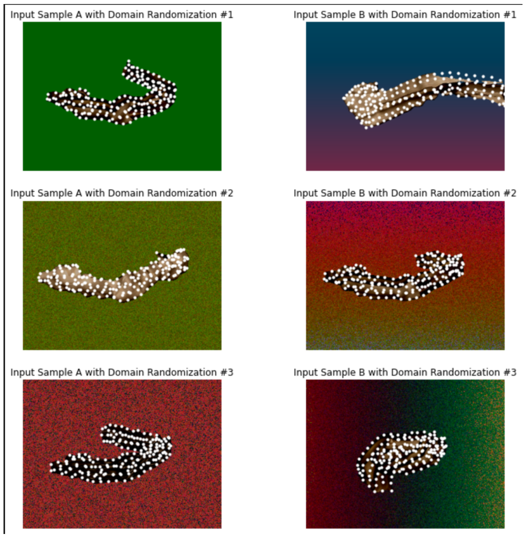
Input Sample A with Domain Randomization #1
Input Sample B with Domain Randomization #1
Input Sample A with Domain Randomization #2
Input Sample B with Domain Randomization #2
Input Sample A with Domain Randomization #3
Input Sample B with Domain Randomization #3

Figure 3. Domain randomized input samples into the dense correspondence mapping network

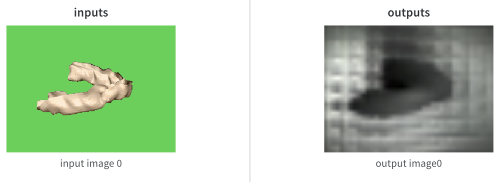

Figure 4. Left: input sample, Right: output of dense descriptor network using D=3

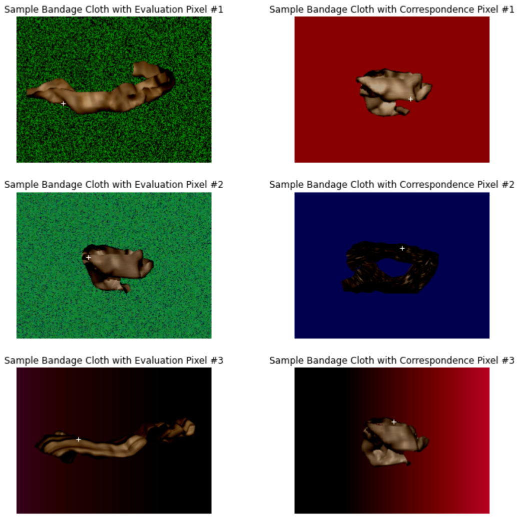
Sample Bandage Cloth with Evaluation Pixel #1
Sample Bandage Cloth with Correspondence Pixel #1
Sample Bandage Cloth with Evaluation Pixel #2
Sample Bandage Cloth with Correspondence Pixel #2
Sample Bandage Cloth with Evaluation Pixel #3
Sample Bandage Cloth with Correspondence Pixel #3

Figure 5. Left: Reference image with pixel, Right: interpolated location of corresponding pixel of dense descriptor through descriptor closest in descriptor space with D=15

6.2. Quantative Evaluation

After generating the dataset with sufficient diversity, a pair of samples are is fed into the network. After the output descriptor mapping of the images are normalized, the 180 descriptor pixels for each pair of images within a batched dataset are compared as follows to calculate *pixel match error*:

1. Index dense descriptor corresponding to a 2D index in descriptor image A
2. Find closest descriptor index in descriptor image B through querying closest L2-distance of all descriptors in descriptor image B
3. Count index successful if interpolated descriptor index is within margin M from actual pixel location from ground truth index projected from bandage 3D points into image space
4. Average successes/total across batches

Descriptor Space	Pixel Match Error
D = 15	50.3%
D = 3	40.3%

Table 1. For a descriptor space size of 15 and 3, the pixel match error is computed according to the steps listed.

7. Conclusion

All in all, this semester's results have shown that using dense descriptors to represent the object state is promising. Although, to get comparable results, there needs to be other representations tested as well to set some baselines such as (Simeonov et al., 2021), a recently published approach that is more sample efficient and robust as it uses point clouds as its base representation. Additionally it is probably more efficient to start moving ahead with policy learning using imitation learning and also developing a POC for the local control policy in parallel to the state estimation work as insights may be drawn across these methods.

References

Florence, P., Manuelli, L., and Tedrake, R. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. *Conference on Robot Learning*, 2018.

She, Y., Wang, S., Dong, S., Sunil, N., Rodriguez, A., and Adelson, E. Cable manipulation with a tactile-reactive gripper. In *Robotics: Science and Systems (RSS)*, 2020.

Simeonov, A., Du, Y., Tagliasacchi, A., Tenenbaum, J. B., Rodriguez, A., Agrawal, P., and Sitzmann, V. Neural descriptor fields: Se(3)-equivariant object representations for manipulation. *arXiv preprint arXiv:2112.05124*, 2021.

Sundaresan, P., Grannen, J., Thananjeyan, B., Balakrishna, A., Laskey, M., Stone, K., Gonzalez, J., and Goldberg, K. Learning rope manipulation policies using dense object descriptors trained on synthetic depth data. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9411–9418, 2020.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.