Learning Interpretable and Transferable Rope Manipulation Policies Using Depth Sensing and Dense Object Descriptors

Priya Sundaresan[1], Brijen Thananjeyan[1], Ashwin Balakrishna[1],
Michael Laskey[2], Kevin Stone[2], Joseph E. Gonzalez[1], Ken Goldberg[1]

***Abstract*— Robotic manipulation of deformable 1D objects such as ropes, cables, and threads is challenging due to the lack of analytic models and large configuration spaces. Furthermore, learning end-to-end manipulation policies directly from images and physical interaction requires significant time cost on a robot and can fail to generalize across tasks. We address these challenges using interpretable deep visual representations for rope extending recent work on dense object descriptors for robot manipulation. This facilitates the design of interpretable and transferable geometric policies built on top of the learned representations, decoupling visual reasoning and control. We present an approach that learns point-pair correspondences between rope configurations, which implicitly encodes geometric structure, entirely in simulation from synthetic depth images. We demonstrate that the learned representation can be used to manipulate a real rope into a variety of different arrangements either by learning from demonstrations or using interpretable geometric policies. In 50 trials of a knot-tying task with the ABB YuMi Robot, the system achieves 66% knot-tying success from unseen configurations. See `https://tinyurl.com/rope-learning` for supplementary material and videos.**

I. Introduction

Manipulating deformable objects is relevant to a wide variety of applications such as surgery, manufacturing, and household robotics [2, 8, 11, 14, 15, 19, 26, 37–39]. We specifically consider manipulation of 1D deformable structures, such as suturing thread in surgery or power cables in households/industrial settings. The infinite dimensional configuration space of these objects makes it difficult to build accurate dynamical models. They also pose significant perception challenges due to self occlusions, loops, and self-similarity [5]. There has been prior work successfully utilizing finite element models [13] and hard-coded representations for deformable manipulation [18, 25, 27, 40], but these techniques can fail to generalize to novel configurations.

These perception and modeling challenges motivate learning-based strategies. Past learning-based approaches have achieved impressive results on a variety of rope manipulation tasks, but require many hours of real-world data collection to learn action-conditioned visual dynamics models of the rope [28, 30, 41]. We address these issues by decoupling perception from planning and control. We learn abstract visual representations of rope by extending the techniques from [11, 34] to learn descriptors for the rope that are invariant across different configurations (see Figure 2). We then demonstrate that these representations can be leveraged to create both *interpretable* (visually intuitive and geometrically structured) and

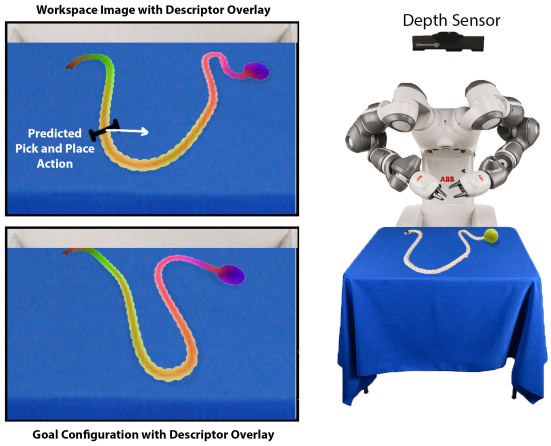

Fig. 1: The robot uses learned object descriptors to compare its current observation to an image of the desired configuration and plan actions to guide the rope to the goal configuration. We use this strategy to track video demonstrations of rope manipulation tasks and to define a geometric algorithm that ties knots from previously unseen starting configurations.

transferable (task and domain-agnostic) polices for learning from video demonstrations and achieving various planar and non-planar rope configurations. Shifting the representational load from the control policy to a separate perception module enables learning to encode information about rope geometry in simulation without real data. Furthermore, because the object descriptors are trained only on images of the rope in different configurations and are agnostic to the actions that generated them, accurate dynamic simulation of the rope is unnecessary.

This paper provides four contributions: (1) experiments suggesting that the dense object descriptors from Florence *et al.* [11] and Schmidt *et al.* [34], previously applied to learn representations for rigid bodies and slightly deformable objects using real data, can be extended to learning representations for highly deformable objects such as rope using only synthetic depth images; (2) a novel approach to achieve complex planar and non-planar rope configurations with a single video demonstration of the task by tracking the learned object descriptors; (3) a geometrically-motivated algorithm using dense object descriptors to tie knots from unseen rope configurations; and (4) experiments on an ABB YuMi robot suggesting the learned representation can be used to achieve a set of planar/non-planar rope configurations and 66% knot-tying success rate in 50 trials from previously unseen states.

II. Background and Related Work

There is recent work on tracking deformable objects in videos such as [8, 10, 29, 32, 34, 35]. There is also

[1]AUTOLAB at the University of California, Berkeley
[2]Toyota Research Institute

extensive literature on deformable manipulation [18, 25, 27, 36, 40]. We primarily focus on learning-based methods, which have been shown to generalize to a wide variety of tasks [28, 30, 41]. Due to the challenge of designing accurate analytical models for deformable objects, [28, 30, 41] provide effective learning-based algorithms for rope manipulation by either generating a visual plan or using an existing one from demonstrations, and then executing the plan by generating controls using learned dynamics models given a single video demonstration. However, these methods require tens of hours of real data collection to learn rope dynamics. These approaches also do not impose any structure on the learned visual representations, limiting the interpretability of the learned policies. In contrast, we impose geometric structure on the learned visual representations, learn them in simulation, and decouple them from robot actions. This accelerates training time substantially, and makes it easier to transfer the learned visual representation across domains.

We learn geometrically meaningful visual representations for rope by using dense object descriptors, introduced in the context of robotic manipulation by [11]. While task agnostic manipulation requires geometric understanding of the objects being manipulated, fine-grained understanding of the object configuration is often unnecessary to effectively grasp or push an object [7, 15, 19, 22–24]. We leverage dense descriptors for task-oriented manipulation, which often requires detailed geometric understanding to manipulate objects in the specific ways needed to achieve task success [9, 11]. There exists extensive literature on generating descriptors for keypoints in images [6, 20], but these approaches rely on image intensity gradients, which will not provide much signal in images where the pixel intensities and textures are largely homogeneous such as for a rope. This motivates a deep learning-based approach to utilize global information about the rope to generate descriptors and correspondences [4, 11, 16, 34].

Schmidt *et al.* [34] propose a deep learning approach to learn a function that maps pixels corresponding to the same point on an object to the same descriptor and pixels corresponding to different points to different descriptors. Florence *et al.* [11] use these dense object descriptors for task-oriented manipulation of rigid and slightly deformable objects. In contrast to prior work, we demonstrate that similar descriptors can be learned and leveraged for manipulation of very deformable 1D structures such as rope. While [11, 16, 34] learn descriptors using color image input, we use synthetic depth input to facilitate sim to real transfer of the learned representations [22, 37].

III. Simulator Design

We use Blender 2.8 [33] — an open-source 3D graphics, animation, and rendering suite — to model the rope in simulation and generate synthetic depth training data. The simulated rope is modelled by twisting four thin cylindrical meshes to produce a realistic braided twine appearance as in [3], with a sphere mesh added on one end to break the symmetry of the rope. The asymmetry reduces ambiguity in descriptor learning. This rope representation consists of a mesh with ordered vertices of known global coordinates and an underlying Bezier curve with M control points, $\mathbf{P}_1, ..., \mathbf{P}_\mathrm{M}$ (Figure 3). A larger M value enables higher manipulation fidelity and a larger configuration space for the rope. Producing varied synthetic depth training data requires simulating the rope in a variety of configurations and exporting the relevant ground truth data and rendered image. For the first step, we randomize the positions of a subset of the Bezier control points to produce varied deformations. Next, for a given scene, we utilize Blender's rendering capabilities to export a depth image from the scene's Z-Buffer output and a mapping $i \rightarrow (u_i, v_i), i \in (1, ..., N)$, which projects the world coordinates of N ordered mesh vertices to pixel coordinates in the synthetic camera frame. The parameter N specifies how many pixels to annotate on the image, so a higher value of N produces more dense pixel match sampling between images during training. In the exported mapping, each i is one of N systematically sampled rope mesh vertex indices. (u_i, v_i) is the pixel coordinate for p_i, the i^{th} mesh vertex, in the frame of the virtual camera in the scene. This raw projection mapping fails to account for complex rope geometries, since multiple mesh vertices can project to the same pixel coordinate at regions of self-intersection or occlusion. Thus, we reparent all pixels in a given region to the top-most mesh vertex in that region using a k-nearest neighbor algorithm with $k = 4$. Pixel matches can be sampled across images of varying configurations by pairing pixels by corresponding mesh vertex.

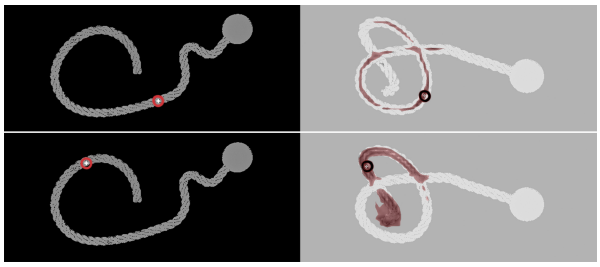

Fig. 2: A visualization of learned descriptors, where the right column images display predicted pixel correspondences (black cursors) relative to the left image source pixels (red cursors) and predicted best match regions (darkened) [11]. This is generated by applying the learned descriptor mapping: $\psi : \mathbb{R}_+^{W \times H \times 1} \rightarrow \mathbb{R}_+^{W \times H \times K}$ to both synthetic depth images, computing the pixelwise norm differences in descriptor space, and scaling these differences linearly $\in$ [0, 255] for the red channel. The reddish regions can be interpreted as a measure of uncertainty in predicted correspondences. Note that the predicted correspondences are sensitive to occlusions and self-intersections.

IV. Dense Descriptor Learning

A. Preliminaries

We consider an environment which consists of a static flat plane and a braided rope and learn policies to achieve specific planar and non-planar configurations. We do this by learning a structured visual representation of the rope to estimate point-pair correspondences between an overhead depth image of the rope and a subgoal image. These correspondences are then used to generate interpretable geometric policies which move the rope to better align it with the subgoal. For more details on how the policies are defined, see Section V.

For visual representation learning, we build on the work in [11, 34]. In Florence *et al.* [11], this is done by first sampling a variety of points on the surface a given object.

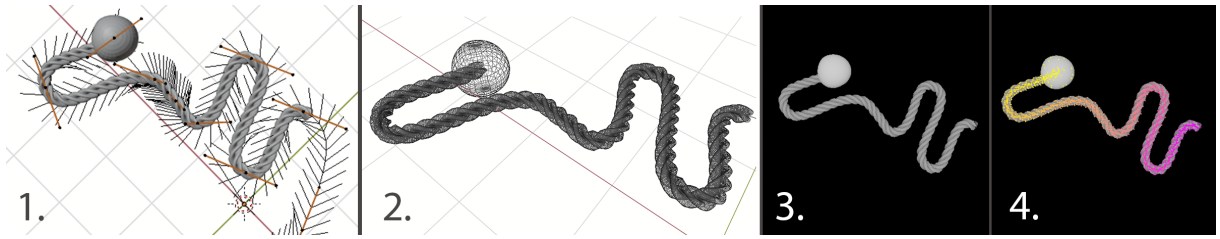

Fig. 3: Rope simulation design. 1) The underlying representation of the rope is a set of M=12 Bezier control points (visualized as black points with orange handles). These nodes can be randomly displaced along x, y, or z axes to produce arbitrary deformation or can be fixed according to a control polygon to produce structured deformation such as loops, overlaps, and knots. 2) The wireframe rope mesh with ordered vertices of known coordinates. 3) A rendered depth map. 4) A visualization of the densely annotated scene with N=1,465 pixels corresponding to N vertices sampled from the rope mesh in 3). The pixels are colored in a stream from yellow to pink starting with mesh vertices indexed at 0 and ending at 1,464, which allows for ordered and dense ground truth annotations in simulation.

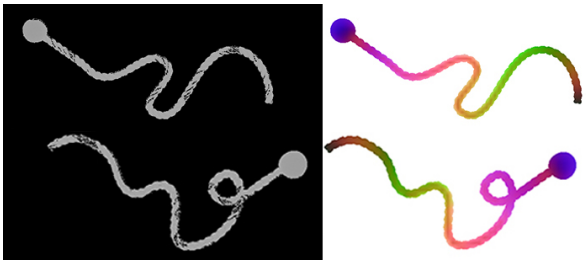

Fig. 4: A visualization of trained, normalized rope descriptors applied to synthetic depth images unseen during training. The first column shows examples of synthetic depth images of a rope in different configurations. The second column represents the output of the dense correspondence network, where for each pixel on the rope mask, the normalized 3D descriptor vector is visualized as a RGB tuple. The visualizations suggest descriptor consistency across deformations.

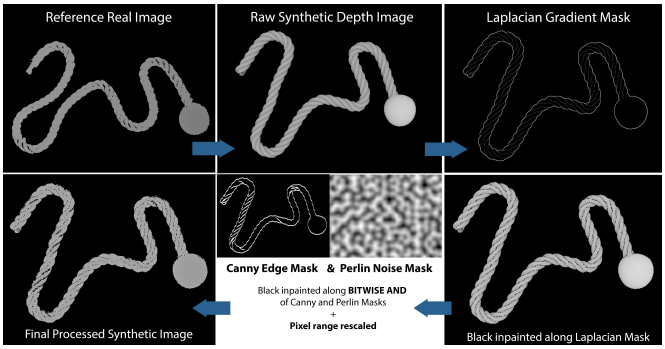

Fig. 5: Sim-to-Real Processing Pipeline. A raw synthetic depth image is post-processed to look like a real reference depth image (top left) by scaling the pixel range and strategically inpainting black along noise, edge, and gradient masks as described in IV-C. This post-processing of the simulation images models the noise and black pixel corruption in real depth images along regions of high gradient. A descriptor mapping is trained on these processed simulated images to enable sim-to-real transfer.

The camera pose is changed via a randomly sampled rigid body transformation and the sampled points are associated with corresponding points in the new view using standard static scene reconstruction techniques. These correspondences are then used to train a Siamese network [17] with pixelwise contrastive loss to learn the desired embedding space. See [11] for more details. Florence *et al.* [11] demonstrate that these descriptors can be used to pick up rigid and slightly deformable objects at specific grasp points from multiple views, even when the target grasp is only identified in one view. Unlike [11], since the rope is not rigid, it is insufficient to simply change the pose of the camera to learn object descriptors for manipulation. Thus, the rope must be manipulated into a variety of different possible configurations to generate useful correspondences. Since ground truth correspondences are difficult to obtain for a real rope, we leverage simulation to obtain point-pair correspondences, which are then used to learn dense object descriptors (Section IV). Unlike [11] which train descriptors on RGB images, we train on synthetic depth [11].

B. Descriptor Learning from Synthetic Depth Images

The training procedure involves sampling a random initial configuration of the rope ξ_1 in simulation and applying some transformation ϕ to yield a new configuration ξ_2. As in Florence *et al.* [11], the goal is to learn a mapping to a descriptor space in which corresponding points on ξ_1 and ξ_2 are encouraged to be close together while non-corresponding points are encouraged to be further apart.

We generate planar ϕ transforms by randomly translating the coordinates of a subsample of the rope's Bezier knots $\mathbf{P}_1,...,\mathbf{P}_M$ along the x and y axes to simulate pulling the rope arbitrarily along different directions. We also generate ϕ transforms that simulate more complex rope configurations including overlap, loops, and knots by geometrically arranging $\mathbf{P}_1,...,\mathbf{P}_M$ into the respective control polygons for these configurations as in [21], and then slightly perturbing knot coordinate positions for variation.

Note that while it is very difficult to obtain ground truth correspondences when arbitrary transformations are applied to a real rope, in simulation we can rapidly obtain ground truth correspondences, making it possible to learn efficiently and create a large, representative training set. We do this by sampling a set of N corresponding point pairs $p = (p_{1i}, p_{2i})_{i=1}^{N}$ on the rope between configurations ξ_1 and ξ_2. This allows us to sample a wide variety of possible rope deformations, making it easier to generalize to different tasks at test-time. Learning in simulation also makes it possible to inject noise to add robustness to varying experimental conditions as described in Section IV-C. Then, we utilize the same training procedure as in [11] to learn K-dimensional descriptors.

C. Domain Randomization

We leverage several domain randomization and image processing techniques to enable sim-to-real transfer by training on rendered synthetic depth images that are post-processed to match real images. In simulation, we slightly randomize over rope thickness $\in$ [0.05, 0.065], rope length $\in$ [14.3, 15], the coil length of the braided texture $\in$ [12.5, 14],

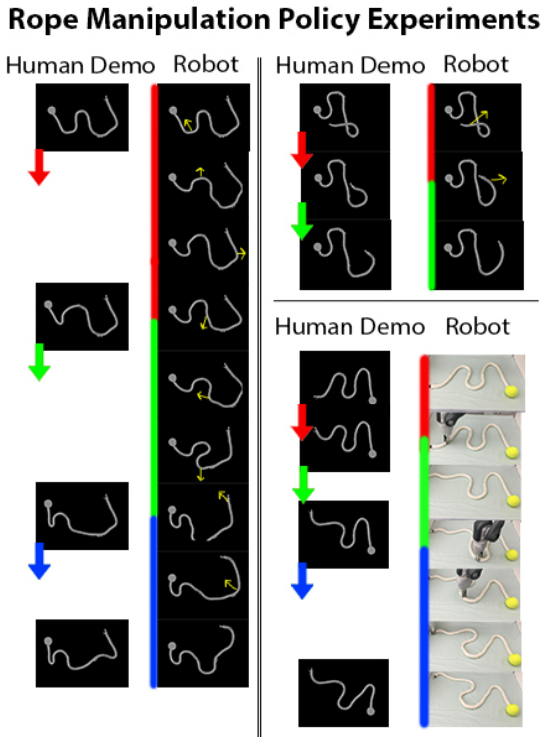

Fig. 6: Three examples of rope manipulation action sequences the YuMi robot performed by one-shot visual imitation of a demonstrated sequence of observations. Each demonstrated sequence consists of a starting configuration followed by pick-and-place actions performed by a human supervisor to produce a different final state. For each step in the demonstration, the YuMi is given a fixed number of pick-and-place attempts (1 for non-planar sequences, 3 for planar sequences) to produce the next sequential state, unless the IoU of the current workspace image and the goal state is below a hand-tuned threshold (0.67). For a single action, the YuMi executes a greedy policy by grasping the correspondence on the rope in the current image that is farthest from its pixelwise match in the goal image and placing it at that point. Qualitative results suggest the efficacy of the geometric policy defined over the learned descriptors. Note that the policies recover from poor actions to complete a sequence (see real rollout (left), actions $6 \rightarrow 7$).

and the radius of the attached sphere $\in$ [0.35, 0.37], all in Blender standard units. Experiments suggest that this provides robustness to slight dimension mismatch between domains. Additionally, we inject both zero-mean, unit variance Gaussian and Poisson noise in the simulated images to model the noise in real depth images. In real depth images of the rope, the corrupted pixels tended to occur along regions of high gradient, particularly on braided rope contours. To model this in the simulated images, we randomly color pixels black along areas of high Laplacian gradient and edges detected with a Canny edge detector [1] on the rope along a Perlin noise mask [31]. The Canny edges and Laplacian gradient provide rope contours and the Perlin noise provides realistic gradient noise. We rescale the pixel range of the simulated images to match the pixel range in real given a single reference real depth image. This process is illustrated in Figure 5.

V. Policy Design

Given the learned descriptors, we design interpretable geometric policies defined over the learned descriptors. We assume that the rope manipulation tasks considered can be performed by a sequence of pick and place actions by a single robotic arm as in prior work [28, 30]. We consider

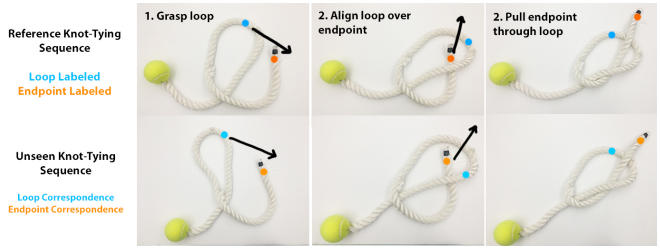

Fig. 7: To perform knot-tying, we label the centered loop point and endpoint of the rope in a reference image, and define two geometric pick-and-place actions in terms of the relative spacing of these points to generate a knot. To generalize to a new initial loop configuration, we recompute loop and endpoint correspondences and execute the sequence.

two algorithmic policies for rope manipulation tasks:

A. Algorithm 1: One-Shot Visual Imitation

In this setting, a human demonstrator makes sequential pick and place actions to arrange the rope arbitrarily. The robot observes the demonstration as a sequence of images from an overhead depth camera and attempts to mimic the sequence by taking actions from a geometric policy.

Actions are generated by using the frames in the provided demonstration as subgoals and using the descriptors to sparsely estimate point-pair correspondences between points on the current depth image of the rope at time t and the current subgoal, given by a demonstration frame (Figure 6). To find correspondences, we randomly sample a set of pixels on the rope mask in the current depth image subject to the constraint that the inter-pixel distance between any two points should be above a margin $M = 50$. For each of the sparsely sampled pixels, we compute their correspondence on the goal image by computing the 100 nearest neighbors in descriptor space and taking the best match to be the median of the associated 100 pixels. We choose the median correspondence due to its robustness to outliers.

Then, we find pairs of corresponding points with high discrepancy (large distance in R^3 between them), and take an action to align these points in 3D space. Specifically, the point-pair correspondence (p_1, p_2) with the maximum discrepancy is computed and the robot grasps the rope at point p_1 and places the rope at point p_2 to align the furthest points in the image. This process is repeated up to k times for each subgoal image or until the intersection-over-union (IoU) of the current and goal state image masks is below a hand-tuned threshold T=0.67. The IoU is a standardized metric across segmentation tasks [12] and provides an indication of the degree of alignment between two masks, which we use to judge the similarity of two rope configurations. We found the IoU to be a noisy measurement for alignment of current and subgoal rope masks, and set T to this relatively low value to account for this. This is likely caused by the long, thin geometry of the rope, which complicates pixelwise alignment of two otherwise very similar rope configurations.

B. Algorithm 2: Descriptor Parameterized Knot-Tying

In this setting, we use a single sequence of frames of a knot-tying task to parameterize a sequence of motion primitives for knot-tying that generalizes to unseen rope configurations. As in [28], we assume the rope contains a single loop initially.

The sequence is annotated with the two pick and place actions used to execute the task (Figure 7).

The first action involves picking the side of the loop close to the end of the rope without the ball and placing it around the endpoint of the rope. We record the descriptor vectors for the grasp point and the end of the rope and use it to define an action in terms of descriptors. When faced with a new, unseen rope configuration with a loop, the robot grasps the closest point in descriptor space to the grasp point in the reference image and pulls it in the direction of the end of the rope, which is also found by matching with the closest descriptor in the reference frame.

The next step involves grasping the end of the rope in the loop and pulling it to tighten the knot. To define this primitive, we record the descriptor vector for the end of the rope in the reference image. When executing this maneuver in a new configuration, the robot detects the end of the rope by finding the closest pixel in descriptor space to the end of the rope in the reference image. The robot grasps at this point and pulls to tighten the knot.

VI. Experiments

We assume access to observations from an overhead depth camera, that the starting configuration of the rope is the same as that in the demonstration, and the the entirety of the rope is supported on a flat plane and the endpoints are visible throughout the duration of the task. We further assume a relatively homogeneous background with limited depth variations and no distractor objects. Finally, to break symmetry in the rope, a tennis ball is tied to one end of the rope to resolve ambiguity between the two ends of the rope.

A. Simulated Experiments

In simulation, we train the deep network used in Florence *et al.* [11] to learn point-pair correspondences for a variety of rope deformations as described in Section IV. Specifically, we define a set of simple and complex deformations and train the network to learn point-pair correspondences for both tiers of deformations. Simple configurations consist of purely planar deformations, formed by picking random points along the rope and pulling arbitrarily along the x and y directions. Complex configurations include planar deformations in addition to randomized overlap, loops, and knots. For each network, we train on a set of 3,600 generated synthetic depth images and evaluate on a held-out test set of 100 pairs of unseen images. For each network, we use $M = 12$ control points to represent the rope. Descriptor quality is measured in terms of pixel-match error on the held-out test set as in [11]. Experiments suggest that the learned descriptors are able to accurately locate correspondences in images of rope in unseen configurations as seen in Figure 4. Furthermore, we note that the descriptors are relatively consistent across different rope configurations (Figure 4).

In Figure 8, we evaluate the quality of the learned descriptors when we vary the sensing modality (synthetic RGB/synthetic depth), the descriptor dimension, the number of annotated correspondences, the input image scale, and when we account for occlusions in the nonplanar datasets. Accounting for occlusions entails using the method described in Section III to reparent conflicting pixels in regions of self-intersection to the appropriate mesh vertex. On the other hand, a lack of occlusion handling uses the raw projection mapping of mesh vertex world coordinates to pixel coordinates. We see that the descriptor quality is largely invariant to small changes in the dimensionality of the descriptor space, the sensing modality used, the number of pixel match annotations, and the method of occlusion handling. One interesting observation is that for non-planar deformations, the gap in the pixelwise error for descriptors trained on RGB and depth data is significantly lower than for planar deformations. This is unsurprising, since for non-planar deformations, there is greater depth variation in the image, making the depth data more useful in differentiating parts of the rope. We also observe that the tennis ball at the end of the rope is necessary to differentiate between each end.

Type	Subgoal	Number of Trials with Improvement
Planar	1	28/32
Planar	2	28/32
Planar	3	23/32
Non-Planar	1	14/21
Non-Planar	2	13/21

TABLE I: Visual Imitation Number of Improved Trials: We report the number of trials that improve with respect to the loss defined in Section VI-B.1. We find that even in the non-planar case, the robot makes positive progress in most trials, but note that performance decreases as the task progresses.

TABLE II: Knot-Tying Failure Modes

Mode	Explanation	Count
A	wrong endpoint correspondence	4
B	wrong loop point correspondence	6
C	endpoint occluded after pull	3
D	loop pulled misaligned	3

B. Physical Experiments

We evaluate the learned representations for designing rope manipulation policies with an ABB YuMi robot equipped with one parallel jaw gripper. We train a simple tier net with a 3-dimensional descriptor and 1,400 labeled correspondences per image. Additionally, we train a complex tier net with a 16-dimensional descriptor and 557 labeled ground truth annotations per image. Both networks are trained on noise-injected simulation images to enable transfer to the real rope. We use the simple and complex networks to perform planar and non-planar manipulation tasks respectively, using the geometric policies from Section V.

1) Alg 1: We evaluate Algorithm 1 on its ability to track and repeat video sequences of both planar and non-planar rope manipulation as shown in Figure 6. Each planar and non-planar sequence consists of three or four frames respectively, including a starting configuration. For each of the subgoals, the robot executes up to 3 or 1 actions for planar and non-planar experiments respectively, and proceeds early to the next subgoal if the IoU threshold in Section V-A is met.

a) Evaluation Metric: To evaluate the agent's ability to track the subgoals in the video sequence, we define a loss function that takes in the realized image I_{real} and the goal image I_{goal}: $L(I_{real}, I_{goal})$. For each image I, a sequence of

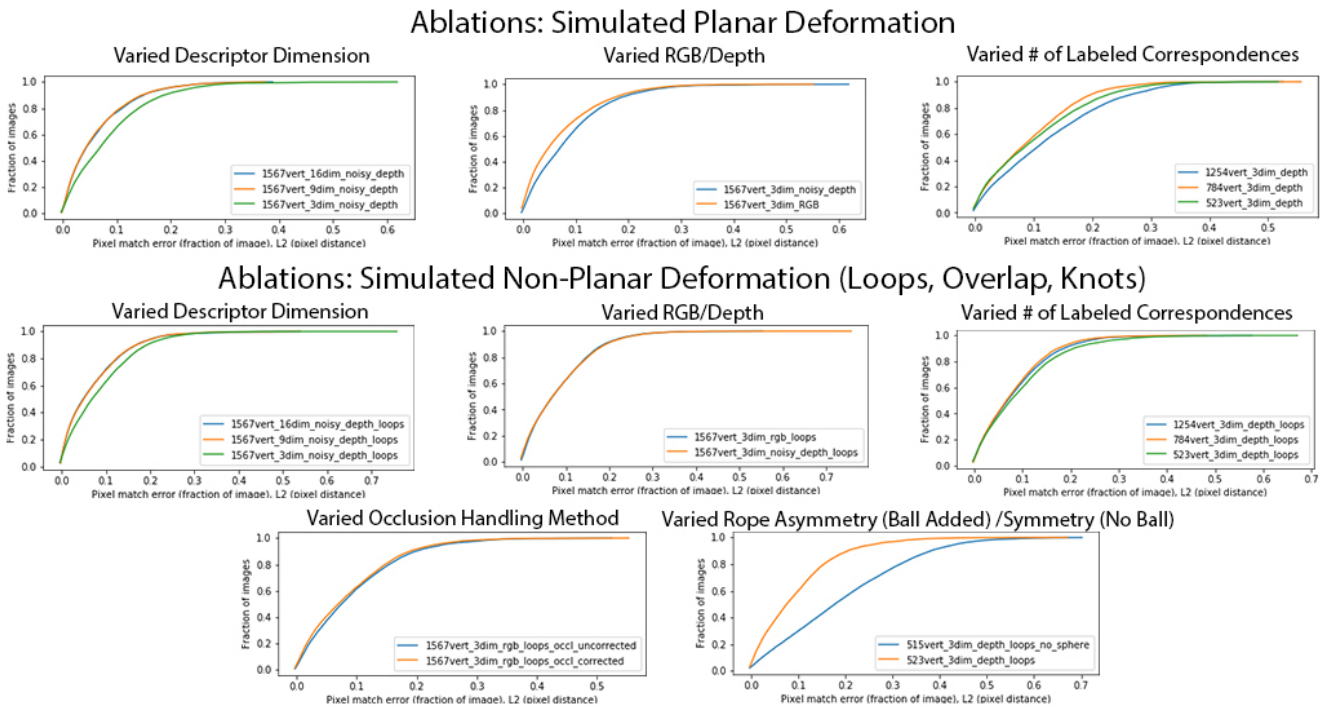

Fig. 8: Ablations measuring pixel-match error for the learned descriptors in simulation when descriptor dimension, sensing modality, number of correspondences used for training, and occlusion handling method are varied. Results suggest that the learned representations are largely insensitive to small changes in these parameters. However, it appears that adding a ball to the end of the rope to break symmetries is critical for good performance, as removing the ball results in a significant deterioration in performance as expected. Furthermore, we note that depth input is more helpful for non-planar configurations, which is unsurprising given the increased depth variation in non-planar settings.

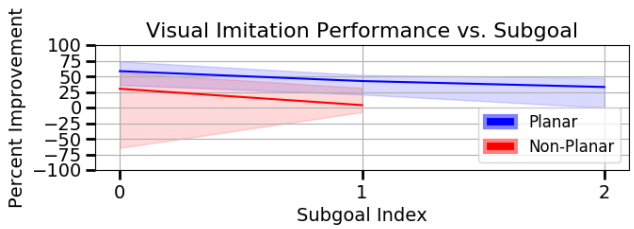

Fig. 9: Visual Imitation Percent Improvement: We report statistics for the percent improvement over each subgoal's starting configuration for the closest frame in the corresponding segment in terms of the loss function defined in Section VI-B.1. We find that the visual imitation policy using dense object descriptors is able to drive the rope to configurations close to the target configurations. We observe that performance deteriorates as the task progresses, which we hypothesize is due to compounding errors over time. We observe that non-planar manipulation is significantly more challenging.

points along the rope is manually annotated, and a parametric piecewise linear function $p_I(i)$ is fit to the points for $i \in [0,1]$. Then, the sum of squared errors is computed for a range of shifts and rotations of I_{real} for 100 evenly spaced points on the curve and the minimum is returned by L. For each subgoal in the demonstration trajectory, L is computed for all frames in the segment corresponding to it in the robot trajectory and report the percent improvement of the best frame over the segment's starting configuration (Figure 9).

2) Alg 2: We evaluate the method in Section V-B on a knot-tying task from 50 unseen configurations with the rope starting in a loop. As in prior work [28, 30], we report the success rate of the task by visually inspecting whether a knot was successfully tied. Figure 7 illustrates the knot-tying procedure used. The robot successfully ties a knot in 33/50 trials (66%). This rate is higher than the knot-tying accuracy reported in [28] (38%) and [30] (60%), and requires weaker supervision, although we do not provide a direct comparison. Failure modes include when the robot fails to accurately identify the loop and endpoint correspondences, fails to align the loop over the endpoint, or occludes the endpoint during alignment, preventing task completion.

VII. Discussion and Future Work

This work presents a new method for designing interpretable and transferable policies for rope manipulation by learning a geometrically structured visual representation entirely in simulation by building on the techniques from [11]. We use this representation to design intuitive geometric policies to track planar and non-planar manipulation from a demonstration and and to design a geometric algorithm for knot tying which performs the task with a 66% success rate. In future work, we will explore learning more complex manipulation primitives in descriptor space such as suturing and dynamic swinging. We will also investigate whether the learned descriptors provide appropriate representations for reinforcement learning.

VIII. Acknowledgments

This research was performed at the AUTOLAB at UC Berkeley with partial support from Toyota Research Institute, the Berkeley AI Research (BAIR) Lab and by equipment grants from PhotoNeo and NVidia. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsors. We thank our colleagues who provided helpful feedback, code, and suggestions, especially David Tseng, Aditya Ganapathi, Michael Danielczuk, and Jeffrey Ichnowski.

REFERENCES

[1] P. Bao, L. Zhang, and X. Wu, "Canny edge detection enhancement by scale multiplication", *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 9, pp. 1485–1490, 2005.

[2] J. P. van den Berg, S. Miller, D. Duckworth, H. Hu, A. Wan, X.-Y. Fu, K. Y. Goldberg, and P. Abbeel, "Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations", *2010 IEEE International Conference on Robotics and Automation*, pp. 2074–2081, 2010.

[3] blended. (2017). Blender tutorial - how to model a rope in blender, YouTube, [Online]. Available: https://youtu.be/xYhIoiOnPj4.

[4] A. Collet, D. Berenson, S. S. Srinivasa, and D. Ferguson, "Object recognition and full pose registration from a single image for robotic manipulation", *2009 IEEE International Conference on Robotics and Automation*, pp. 48–55, 2009.

[5] R. H. Crowell and R. Fox, *Introduction to knot theory*. Springer Science & Business Media, 2012.

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection", *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 886–893 vol. 1, 2005.

[7] M. Danielczuk, J. Mahler, C. Correa, and K. Goldberg, "Linear push policies to increase grasp access for robot bin picking", in *Proc. IEEE Conf. on Automation Science and Engineering (CASE)*, IEEE, 2018.

[8] D. G. Dansereau, S. P. N. Singh, and J. Leitner, "Interactive computational imaging for deformable object analysis", *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4914–4921, 2016.

[9] R. Detry, J. Papon, and L. H. Matthies, "Task-oriented grasping with semantic and geometric scene understanding", *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3266–3273, 2017.

[10] D. Du, H. Qi, W. Li, L. Wen, Q. Huang, and S. Lyu, "Online deformable object tracking based on structure-aware hyper-graph", *IEEE Transactions on Image Processing*, vol. 25, pp. 3572–3584, 2016.

[11] P. R. Florence, L. Manuelli, and R. Tedrake, "Dense object nets: Learning dense visual object descriptors by and for robotic manipulation", in *CoRL*, 2018.

[12] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask r-cnn", *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

[13] J. E. Hopcroft, J. K. Kearney, and D. B. Krafft, "A case study of flexible object manipulation", *I. J. Robotics Res.*, vol. 10, pp. 41–50, 1991.

[14] E. Jang, C. Devin, V. Vanhoucke, and S. Levine, "Grasp2vec: Learning object representations from self-supervised grasping", in *CoRL*, 2018.

[15] Y.-B. Jia, F. Guo, and H. Lin, "Grasping deformable planar objects: Squeeze, stick/slip analysis, and energy-based optimalities", *I. J. Robotics Res.*, vol. 33, pp. 866–897, 2014.

[16] M. Khoury, Q.-Y. Zhou, and V. Koltun, "Learning compact geometric features", *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 153–161, 2017.

[17] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition", in *ICML Deep Learning Workshop*, vol. 2, 2015.

[18] Y. Kuniyoshi, M. Inaba, and H. Inoue, "Learning by watching: Extracting reusable task knowledge from visual observation of human performance", *IEEE Trans. Robotics and Automation*, vol. 10, pp. 799–822, 1994.

[19] H. Lin, F. Guo, F. Wang, and Y.-B. Jia, "Picking up a soft 3d object by "feeling" the grip", *I. J. Robotics Res.*, vol. 34, pp. 1361–1384, 2015.

[20] D. G. Lowe, "Distinctive image features from scale-invariant keypoints", *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[21] Y. L. Ma and W. T. Hewitt, "Point inversion and projection for nurbs curve and surface: Control polygon approach", *Computer Aided Geometric Design*, vol. 20, no. 2, pp. 79–99, 2003.

[22] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics", *Proc. Robotics: Science and Systems (RSS)*, 2017.

[23] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg, "Dex-net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning", in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, IEEE, 2018, pp. 1–8.

[24] J. Mahler, F. T. Pokorny, B. Hou, M. Roderick, M. Laskey, M. Aubry, K. Kohlhoff, T. Kröger, J. Kuffner, and K. Goldberg, "Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards", in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, IEEE, 2016, pp. 1957–1964.

[25] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, and P. Abbeel, "Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding", *2010 IEEE International Conference on Robotics and Automation*, pp. 2308–2315, 2010.

[26] H. G. Mayer, F. J. Gomez, D. Wierstra, I. Nagy, A. Knoll, and J. Schmidhuber, "A system for robotic heart surgery that learns to tie knots using recurrent neural networks", *Advanced Robotics*, vol. 22, pp. 1521–1537, 2006.

[27] T. Morita, J. Takamatsu, K. Ogawara, H. Kimura, and K. Ikeuchi, "Knot planning from observation", *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol. 3, 3887–3892 vol.3, 2003.

[28] A. Nair, D. Chen, P. Agrawal, P. Isola, P. Abbeel, J. Malik, and S. Levine, "Combining self-supervised learning and imitation for vision-based rope manipulation", *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2146–2153, 2017.

[29] R. A. Newcombe, D. Fox, and S. M. Seitz, "Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time", *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 343–352, 2015.

[30] D. Pathak, P. Mahmoudieh, G. Luo, P. Agrawal, D. Chen, Y. Shentu, E. Shelhamer, J. Malik, A. A. Efros, and T. Darrell, "Zero-shot visual imitation", *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2131–21 313, 2018.

[31] K. Perlin, "Improving noise", in *ACM transactions on graphics (TOG)*, ACM, vol. 21, 2002, pp. 681–682.

[32] C. Qian, X. W. Sun, Y. Wei, X. Tang, and J. Sun, "Realtime and robust hand tracking from depth", *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1106–1113, 2014.

[33] T. Roosendaal, "Blender foundation", *The essential Blender: guide to 3D creation with the open source suite Blender*, 2007.

[34] T. Schmidt, R. A. Newcombe, and D. Fox, "Self-supervised visual descriptor learning for dense correspondence", *IEEE Robotics and Automation Letters*, vol. 2, pp. 420–427, 2017.

[35] J. Schulman, A. Gupta, S. Venkatesan, M. Tayson-Frederick, and P. Abbeel, "A case study of trajectory transfer through non-rigid registration for a simplified suturing scenario", *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4111–4117, 2013.

[36] J. Schulman, J. Ho, C. Lee, and P. Abbeel, "Learning from demonstrations through the use of non-rigid registration", in *ISRR*, 2013.

[37] D. Seita, N. Jamali, M. Laskey, R. Berenstein, A. K. Tanwani, P. Baskaran, S. Iba, J. Canny, and K. Goldberg, "Deep transfer learning of pick points on fabric for robot bed-making", in *ISRR*, 2019.

[38] B. Thananjeyan, A. Balakrishna, U. Rosolia, F. Li, R. McAllister, J. E. Gonzalez, S. Levine, F. Borrelli, and K. Goldberg, "Extending deep model predictive control with safety augmented value estimation from demonstrations", *CoRR*, vol. abs/1905.13402, 2019. arXiv: 1905.13402.

[39] B. Thananjeyan, A. Garg, S. Krishnan, C. Chen, L. Miller, and K. Goldberg, "Multilateral surgical pattern cutting in 2d orthotropic gauze with deep reinforcement learning policies for tensioning", in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 2371–2378.

[40] H. Wakamatsu, E. Arai, and S. Hirai, "Knotting/unknotting manipulation of deformable linear objects", *I. J. Robotics Res.*, vol. 25, pp. 371–395, 2006.

[41] A. Wang, T. Kurutach, K. Liu, P. Abbeel, and A. Tamar, "Learning robotic manipulation through visual planning and acting", *RSS*, vol. abs/1905.04411, 2019.