Leveraging Appearance Priors in Non-Rigid Registration, with Application to Manipulation of Deformable Objects

Sandy H. Huang Jia Pan George Mulcaire Pieter Abbeel

***Abstract*— Manipulation of deformable objects is a widely applicable but challenging task in robotics. One promising non-parametric approach for this problem is *trajectory transfer*, in which a non-rigid registration is computed between the starting scene of the demonstration and the scene at test time. This registration is extrapolated to find a function from $\mathbb{R}^3$ to $\mathbb{R}^3$, which is then used to warp the demonstrated robot trajectory to generate a proposed trajectory to execute in the test scene. In prior work [1] [2], only depth information from the scenes has been used to compute this warp function. This approach ignores appearance information, but there are situations in which using both shape and appearance information is necessary for finding high quality non-rigid warp functions.**

In this paper, we describe an approach to learn relevant appearance information about deformable objects using deep learning, and use this additional information to improve the quality of non-rigid registration between demonstration and test scenes. Our method better registers areas of interest on deformable objects that are crucial for manipulation, such as rope crossings and towel corners and edges. We experimentally validate our approach in both simulation and in the real world, and show that the utilization of appearance information leads to a significant improvement in both selecting the best matching demonstration scene for a given test scene, and finding a high quality non-rigid registration between those two scenes.

I. Introduction

In robotic manipulation of deformable objects, a key challenge is operating in high-dimensional, continuous state and action spaces. Accounting for the complicated dynamics of deformable objects is also difficult. Despite these challenges, recent work has shown promising results in manipulating deformable objects through *learning from demonstration* (LfD). In LfD, the robot generalizes from human demonstrations of a given task in order to perform that task autonomously in new scenes. One approach to LfD uses *trajectory transfer* [1] [2], which consists of first selecting a demonstration to apply and then finding a non-rigid warp from the demonstration scene to the test scene. This warp is then applied to the human-demonstrated gripper trajectory, to generate a proposed trajectory to execute in the test scene.

A state-of-the-art approach to finding this non-rigid warp in trajectory transfer uses Thin Plate Spline Robust Point Matching (TPS-RPM), which takes in each scene in the form of 3D point clouds. TPS-RPM iteratively solves for (i) soft point correspondences between the source and target points, and (ii) a warp function that balances non-rigidness of the warp with effective matching of corresponding target and warped source points [3]. Regularizing the warp to be as rigid as possible increases the likelihood of the transferred trajectory succeeding in the test scene.

Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, CA, USA. {shhuang, jia.pan, gmulcaire, pabbeel}@cs.berkeley.edu

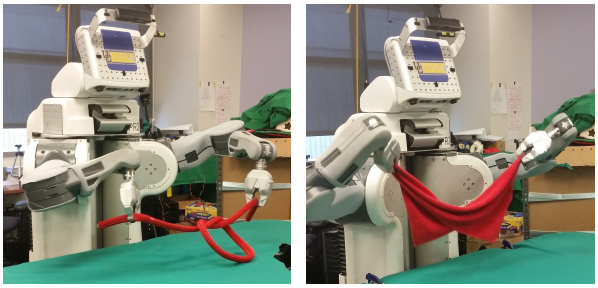


Fig. 1: Objects often have regions that can be clearly distinguished with appearance information, but not with only shape information. Our approach uses learned appearance information to encourage better matching of crucial points between source and target scenes, which produces a higher quality warp function from source to target points, and thus more reliably successful execution. We apply our approach to two challenging deformable object manipulation tasks: tying knots and folding towels. Our approach enables the first successful application of LfD to perform towel folding when starting from a fully crumpled state.

However, this approach only considers the locations of the source and target points; it has no mechanism of ensuring that, for example, towel corners and edges in the source scene are registered to those in the target scene, and rope crossings and endpoints in the source are registered to those in the target. This additional information is necessary for generating high-quality warps in certain situations, and can be captured by local appearance descriptors.

The contributions of this paper are: (i) An approach to learning feature descriptors invariant to small non-rigid transformations, and (ii) A method of using these descriptors to incorporate appearance of deformable objects into non-rigid registration. We apply our approach to two instances of deformable object manipulation: tying overhand knots in ropes and folding towels (Figure 1). Our experiments show that, in general, integrating appearance information into the TPS-RPM algorithm produces higher quality warps from demonstration scenes to test scenes, which improves demonstration selection and makes the transferred trajectory more likely to succeed in the test scene. Our approach enables the first successful application of LfD to perform towel folding starting from a fully crumpled state, which can be seen at `http://rll.berkeley.edu/iros2015ap/`.

II. Related Work

A. Deformable Object Manipulation

Robotic manipulation of deformable objects has been studied in various contexts, ranging from surgery to industrial

environments. An extensive survey is available in [4].

One line of previous work uses motion planning techniques for deformable object manipulation. These methods apply traditional planning algorithms such as Probabilistic Roadmap [5] [6] or Rapidly-exploring Random Trees [7] to compute a set of feasible motions for manipulating an object. Such techniques have been successfully applied to a variety of manipulation tasks, including knot tying [5] and needle insertion [8]. However, these approaches require an explicit deformation model for the object, and rely on physical simulation during planning. Recent works [9] [10] extend the planning framework to not rely on modeling or simulation of the objects, but these approaches cannot handle complicated tasks such as knot tying or cloth folding.

Towel folding from a fully crumpled state has been accomplished in the past through executing a series of manually-defined trajectories based on the locations of detected grasp points [11]. However, this approach requires manually defining trajectories and a state machine of transitions between states of the towel, which is expensive and limits the failure states that the robot can recover efficiently from.

Another line of work for deformable object manipulation uses human demonstrations to teach robots how to perform complicated tasks, without requiring modeling or simulation of the object. In the so-called learning from demonstration (LfD) paradigm [12], a human expert demonstrates the task one or more times, and a learning algorithm generalizes these demonstrations so that the robot can perform the tasks autonomously under new, yet similar, situations. This approach has shown promising results for a range of robotic manipulation tasks involving deformable objects, including pizza dough flattening [13], pancake flipping [14], knot tying [15] [16] [17], and cloth folding [2] [18]. Prior applications of LfD to cloth folding require that the cloth starts out in a flat spread-out state, whereas our approach enables folding towels that start from fully crumpled states, by leveraging appearance information.

B. Local Appearance Descriptors

Defining or learning local appearance descriptors is a well-studied problem that is applicable to many tasks. In the context of non-rigid registration, one approach is labeling corresponding landmarks in both the source and target images, and using those fixed correspondences to calculate the registration [19] [20]. These landmarks can be determined manually [19], semi-automatically [20], or automatically. SIFT Flow [21] is a method for establishing correspondences across two distinct scenes; it produces a flow field by matching densely computed scale-invariant feature transform (SIFT) descriptors [22].

Previous work on hand-engineered local appearance descriptors often focuses on making these descriptors invariant with respect to viewpoint changes (i.e., rigid transformations) or illumination [23]. Examples include shape context [24], SIFT [22], histogram of oriented gradients (HOG) [25], and DAISY [26], which have proven to be effective in a variety of applications. By contrast, for our task, we seek descriptors that are also invariant to small *non-rigid* transformations of the object. In our work, rather than hand-engineering new local appearance descriptors for deformable objects, we will follow a deep learning approach. We will compare these deep-learned local appearance descriptors against existing hand-engineered descriptors.

C. Deep Learning

Recently deep learning methods, particularly in the form of Convolutional Neural Networks (CNNs), have emerged as an alternative to hand-engineered features, and have achieved impressive results in a variety of computer vision tasks. In particular, Krizhevsky et al. [27] achieved breakthrough results on the ImageNet classification task [28]. Girshick et al. [29] developed R-CNN, which achieved state-of-the-art performance on the task of object detection on PASCAL VOC [30]. Importantly, they showed substantial improvement by first using a pre-trained network on the large ILSVRC12 classification dataset and then fine-tuning on the smaller PASCAL VOC object detection dataset.

Most previous work falls within the contexts of classification [27] and detection [29]. However, neither is sufficient for complex manipulation tasks, especially those involving deformable objects. For such tasks, it is important to accurately determine the configuration of each deformable component of the object. Recent work has applied deep learning towards the classification of deformable objects, but not their manipulation [31] [32].

III. Background

A. Trajectory Transfer through Non-Rigid Registration

(i) Non-Rigid Registration with Known Correspondences

Non-rigid registration from a source to a target scene computes a warping function $\hat{\mathbf{f}}$ that minimizes both the registration error and a regularization term. Given a set of point correspondences $(\mathbf{x}_i, \mathbf{y}_i)$ between the two scenes and a regularizer r, the goal is to find the warping function $\hat{\mathbf{f}} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ such that:

$$\hat{\mathbf{f}} = \underset{\mathbf{f}}{\operatorname{argmin}} \quad \sum_i ||\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i||^2 + \lambda r(\mathbf{f}) \tag{1}$$

where λ is a hyper-parameter that trades off between the registration error and regularization of $\mathbf{f}$.

We set $r(\mathbf{f})$ to be the Thin Plate Spline (TPS) regularizer [33] [34], a commonly-used effective regularizer that calculates the *bending cost* of $\mathbf{f}$:

$$r(\mathbf{f}) = ||\mathbf{f}||^2_{\text{TPS}} = \int d\mathbf{x} ||\mathrm{D}^2\mathbf{f}(\mathbf{x})||^2_{\text{Frob}} \ , \tag{2}$$

where $\mathrm{D}^2\mathbf{f}(\mathbf{x})$ is the Hessian of $\mathbf{f}$ at $\mathbf{x}$, and $||\cdot||_{\text{Frob}}$ denotes the Frobenius norm. With this choice of $r(\mathbf{f})$, the solution to the non-rigid registration problem in Equation 1 can be expressed as an affine transformation plus a weighted sum of basis functions $\sigma(\cdot)$ around the source points $\mathbf{x}_i$. More concretely, for $\mathbf{x} \in \mathbb{R}^3$, $\hat{\mathbf{f}}$ has the form

$$\hat{\mathbf{f}}(\mathbf{x}) = \sum_i \mathbf{a}_i \sigma(\mathbf{x} - \mathbf{x}_i) + \mathbf{B}\mathbf{x} + \mathbf{c} \ , \tag{3}$$

where $\sigma(\mathbf{x}-\mathbf{x}_i) = -||\mathbf{x}-\mathbf{x}_i||_2$, $\mathbf{a}_i \in \mathbb{R}^3$, $\mathbf{B} \in \mathbb{R}^{3\text{x}3}$, and $\mathbf{c} \in \mathbb{R}^3$. In addition, $\mathbf{a}_i$ must satisfy $\sum_i \mathbf{a}_i^d x_i^d = 0$ and $\sum_i \mathbf{a}_i^d = 0$ for all dimensions $d \in \{1, 2, 3\}$. Using this known structure of $\mathbf{f}$, Equation 1 can be efficiently solved analytically [34].

(ii) Non-Rigid Registration with Unknown Correspondences

When point correspondences are unknown, the Thin Plate Spline Robust Point Matching (TPS-RPM) algorithm [3] solves the problem by iteratively (i) estimating soft correspondences between the point clouds of two scenes and (ii) fitting the optimal TPS transformation $\hat{\mathbf{f}}$ based on these estimated scene correspondences. This is equivalent to coordinate descent on the following joint optimization problem, where $\mathbf{M}$ is the correspondence matrix with $m_{ij} \in [0, 1]$ indicating the degree of correspondence between point $\mathbf{x}_i$ and $\mathbf{y}_j$:

$$\begin{aligned} \underset{\mathbf{f},\mathbf{M}}{\text{minimize}} \quad & E(\mathbf{f}, \mathbf{M}; T, \zeta) + \lambda||\mathbf{f}||^2_{\text{TPS}} \\ \text{subject to} \quad & \sum_{i=1}^{N+1} m_{ij} = 1, \sum_{j=1}^{N'+1} m_{ij} = 1, m_{ij} \geq 0 \ , \end{aligned} \tag{4}$$

$$\begin{aligned} E(\mathbf{f}, \mathbf{M}; T, \zeta) = & \sum_{i=1}^{N}\sum_{j=1}^{N'} m_{ij}||\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_j||_2^2 \\ & + T\sum_{i=1}^{N}\sum_{j=1}^{N'} m_{ij}\log m_{ij} - \zeta\sum_{i=1}^{N}\sum_{j=1}^{N'} m_{ij} \end{aligned}$$

$m_{i(N'+1)}$ represents the likelihood of $\mathbf{x}_i$ being an outlier, and $m_{(N+1)j}$ represents the same for $\mathbf{y}_j$. As before, λ controls the tradeoff between how well $\mathbf{f}$ minimizes the energy function E and the bending cost of $\mathbf{f}$. The temperature T controls how soft the correspondences are, with a low temperature favoring a harder m_{ij}. The parameter ζ controls the preference for matching points to non-outliers.

This objective is non-convex, so the solution obtained from coordinate descent is only guaranteed to be locally optimal. In coordinate descent, minimizing with respect to $\mathbf{M}$ is equivalent to the update

$$\hat{m}_{ij} \propto \exp\left(-\frac{1}{T}||\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_j||^2\right) \ , \tag{5}$$

followed by iterative row and column normalization. Given these correspondences $\hat{m}$, minimizing with respect to $\mathbf{f}$ is equivalent to the update

$$\hat{\mathbf{f}} = \underset{\mathbf{f}}{\text{argmin}} \quad \sum_{i=1}^{N} \mathbf{w}_i||\mathbf{f}(\mathbf{x}_i) - \bar{\mathbf{y}}_i||_2^2 + \lambda||\mathbf{f}||^2_{\text{TPS}} \ , \tag{6}$$

where $\mathbf{w}_i = \sum_{j=1}^{N'} \hat{m}_{ij}$ and $\bar{\mathbf{y}}_i = \frac{\sum_{j=1}^{N'} \hat{m}_{ij}\mathbf{y}_j}{\mathbf{w}_i}$. Note that $\mathbf{w}_i$ also equals $1 - \hat{m}_{i(N'+1)}$, due to row and column normalization of $\mathbf{M}$.

TPS-RPM embeds coordinate descent within deterministic annealing: it iteratively alternates between performing the two update equations 5 and 6, while gradually reducing the temperature T.

(iii) TPS-RPM for Trajectory Transfer

Using a point cloud representation for the demonstration starting scene and for the test scene, Schulman et al. [2] use the TPS-RPM algorithm to jointly find point correspondences and a warping between the two scene point clouds. The resulting warp function is then used to warp the path traced by the end effector of the robot in the demonstration.

However, it is challenging to compute a high quality warp function based solely on the location of points, because this only encodes low level shape information of the corresponding objects, and thus lacks important semantic knowledge. For instance, a good correspondence between ropes should take into account the topology of the ropes; in other words, the endpoints and/or crossings of the two ropes should match. For towels, corners and edges should be considered during the warp function computation, as well as the presence of wrinkled as opposed to flat surfaces. We use convolutional neural networks to learn these high level semantic features for deformable objects. We show incorporating this information improves the warps found by TPS-RPM, thus enabling more successful trajectory transfer.

B. Convolutional Neural Networks

To learn local appearance descriptors, we use deep convolutional neural networks (CNNs), which have led to breakthrough results on a variety of pattern recognition problems [27] [29]. In general, neural networks are function approximators and are trained in an end-to-end paradigm. In our application, we use CNNs to classify image patches into regions-of-interest on towels and ropes, as further explored in Section IV.

CNNs are comprised of layers of convolutional filters that are automatically learned from the training data. A single convolutional filter looks for a specific pattern from its input layer, expressed as a linear combination on the inputs of a small spatial window. For example, filters in the first convolutional layer, which operate on pixels when the input is an image, can identify low-frequency patterns such as different blobs of colors or high-frequency patterns such as edges in various orientations. Multiple convolutional filters are used in each layer, so that many patterns can be captured and stored. Filters in subsequent layers can then learn to identify increasingly complex structures, such as textures and shapes. Through the network, spatial resolution is gradually traded for increased semantic understanding of the input. The convolutional nature of the net regularizes the model so that expressive features can be learned through end-to-end training with a relatively small number of parameters.

Convolutional layers are followed by non-linear units, typically rectified linear units (ReLU). For image classification, the last layer of the network is a multinomial logistic regression classifier. The whole network is trained through backpropagation, commonly in the form of batch stochastic gradient descent. Techniques to accelerate learning and reduce overfitting include momentum and dropout, respectively.

Our CNN is inspired by the Alexnet architecture, which won the 1000-category ILSVRC12 challenge [28]. We train

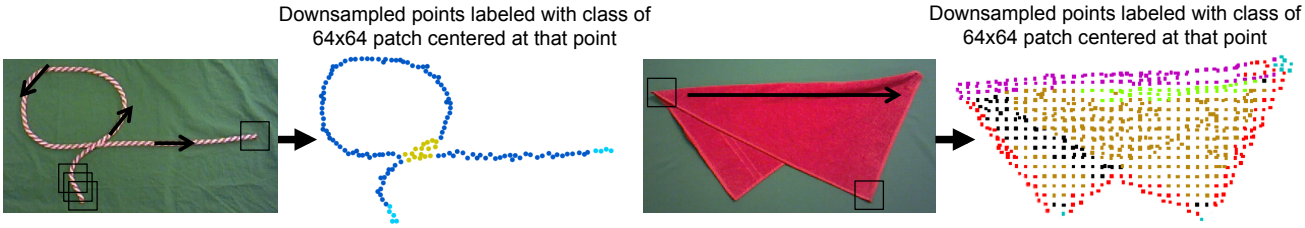


Fig. 2: Given a rope or towel image, we first use a color filter to remove points that are the background color, and downsample using a voxel size of 2.5 centimeters. Then, we use the learned deep CNN to predict the label for the 64x64 window centered around each point, resulting in a labeled point cloud. For the rope, endpoints, crossings and regular sections are labeled as teal, blue, and yellow, respectively. For the towel, corners-against-background, edges-against-background, edges-against-interior, folds-against-background, flat interior, and wrinkled interior are labeled as cyan, red, black, purple, brown, and green.

		Linear SVM	Alexnet [27], finetuned	3 conv, 3 fc CNN (1)	2 conv, 3 fc CNN (2)	2 conv, 3 fc CNN (3)	3 conv, 2 fc CNN (4)	1 conv, 3 fc CNN (5)
Number of Weights to Train		17,769	20,184,452	1,673,994	1,171,268	127,818	1,266,474	175,914
Rope	Validation Accuracy	0.969	0.989	**0.993**	0.989	0.991	**0.993**	0.850
	Time to Train	516s	6600s	3880s	3510s	3480s	3860s	1810s
	Time to Test, on Validation	0.128s	18.6s	7.62s	7.32s	7.30s	7.61s	5.59s
Towel	Validation Accuracy	0.894	**0.940**	0.935	0.938	0.935	0.653	0.616
	Time to Train	1390s	6110s	1780s	1590s	1570s	1730s	835s
	Time to Test, on Validation	0.390s	15.8s	6.40s	6.04s	6.00s	6.34s	4.72s

TABLE I: Comparison of using a linear SVM (with L2 penalty, L2 loss, C = 0.001) and different neural architectures for supervised learning of rope and towel 64x64 patch labels. The input features for the linear SVM were HOG and DAISY features calculated from each patch. Alexnet (finetuned from weights trained on ILSVRC12) achieves the highest validation set accuracy. We choose to use a smaller neural net, CNN (3), for more efficient classification with essentially on-par performance for both rope and towel.

this CNN on our rope and towel datasets by using the open source Caffe library [35].

IV. Acquiring Appearance Information

For both the rope and towel, we use an RGBD camera to acquire images and 3D point clouds of the object in a variety of configurations. We capture appearance information by training a classifier to classify patches of the RGB image as different parts of the object. We tried several different classifiers: a linear SVM trained on HOG and DAISY features of the patches, the full Alexnet architecture, as well as several simpler deep CNN architectures inspired by Alexnet. We found one of the simpler CNNs worked best for this task.

A. Labeling Data

For each image, we manually label 64-by-64 patches with the category that each patch's center point belongs to. These patches are selected to have adequate representation for each category. For the rope there are four types of labels: *crossing*, *endpoint*, *regular section* (i.e., any part that is neither a crossing nor an endpoint), and *background*. For the towel there are ten types of labels: *corner-against-background*, *corner-against-towel*, *corner-against-background-and-towel*, *edge-against-background*, *edge-against-towel*, *fold-against-background*, *fold-against-interior*, *flat towel interior*, *wrinkled towel interior*, and *background*. The *background* refers to the surface on which the object lies, which is a solid color distinct from that of the object. We distinguish towel edges from folds as well as flat from wrinkled interiors because they often contain valuable information about which demonstration should be used, and where the robot should grasp during execution. This is also why we distinguish between whether a corner, edge, or fold lies directly on top of the surface or on top of the towel. Figure 2 shows examples of the patches for ropes and towels.

B. Training and Choosing a Classifier

We label patches for 100 and 121 images of rope and towel configurations, respectively. We randomly split these images into training and validation sets: the training set consists of 7,645 patches for rope and 5,668 for towel, and the validation set consists of 1,611 for rope and 1,356 for towel.

We tried using several different classifiers (Table I). First, we used a linear SVM trained on hand-engineered local appearance descriptors (specifically, HOG and DAISY features) of the patches, and compared that to finetuning Alexnet from weights trained on ILSVRC12. The latter results in better classification accuracy on the validation set for both towel (0.940 vs. 0.894) and rope (0.989 vs. 0.969). [1]

However, classifying image patches is much simpler than the ImageNet task that Alexnet was designed for, so we also explored CNN architectures with fewer layers and filters (Table II). We found CNN (3) maintains high validation set accuracy while classifying patches in 40% of the time as Alexnet requires (Table I). Thus, we use CNN (3) as the classifier in our experiments.

C. Overall Pipeline

Given an RGBD point cloud of a rope or towel, we first use a color filter to remove points that are the background color.

[1]Note that while Alexnet is run on 227x227 images (in the Caffe reference implementation), the first 5 layers, which are convolutional, can be readily adapted to operate on our smaller 64x64 patches. The feature map at this stage is reduced to a 1x1 spatial resolution, rather than 6x6 for a conventional 227x227 input image. Due to this difference, finetuning can only be done for the convolutional layers; the three fully connected layers are trained from scratch.

TABLE II: The CNN architectures we experimented with for learning local appearance descriptors. The simpler CNNs are based off Alexnet, and their convolutional layers have the same filter size and stride as the corresponding layer in Alexnet. The last fully connected layer, fc8, has either 4 or 10 filters, corresponding to the 4 classes of rope patches or 10 classes of towel patches. The simpler CNN architectures do not have the first pooling layer of Alexnet, and do not pad images—we found that these changes improved their performance.

	Alexnet [27]	**CNN (1)**	**CNN (2)**	**CNN (3)**	**CNN (4)**	**CNN (5)**
conv1	96	64	64	64	64	64
conv2	256	32	32	32	32	-
conv3	384	32	-	-	32	-
conv4	384	-	-	-	-	-
conv5	256	-	-	-	-	-
fc6	4096	1024	1024	256	1024	1024
fc7	4096	1024	1024	256	-	1024

Then we downsample the point cloud using a voxel size of 2.5 centimeters, in order to speed up computation of non-rigid registrations. For each remaining point, we use CNN (3) to determine the labeling for the 64-by-64 patch centered at that point. The end result is a probability distribution over patch classes, at each downsampled point (Figure 2).

V. TPS-RPM with Appearance Information

A. TPS-RPM with Priors

We incorporate appearance information by building on Combès et al.'s use of a prior on point correspondences in TPS-RPM [36]. This prior encodes the probability that a given source point and target point should be matched, independent of the registration function and the spatial proximity between the warped source point and target point. Combès et al. show that by viewing TPS-RPM as a variation of the Expectation-Maximization (EM) algorithm, incorporating a prior on point correspondences reduces to defining a new soft point correspondence matrix $\mathbf{M}'$ such that $m'_{ij} = \pi_{ij} m_{ij}$, where π_{ij} is the prior probability that the source point x_i and target point y_j should be matched [36]. Thus, incorporating this prior in TPS-RPM only impacts solving for soft point correspondences, not the optimization of the warp function.

In this work, we set $\pi_{ij} \propto e^{\beta s(\mathbf{x}_i, \mathbf{y}_j)}$, where is β is a hyperparameter that controls the degree of influence of the prior on the point correspondences, and $s(\mathbf{x}_i, \mathbf{y}_j)$ is a non-negative function that encodes the similarity between points $\mathbf{x}_i$ and $\mathbf{y}_j$. Our new update for $\mathbf{M}$ is now

$$\hat{m}_{ij} \propto \exp\left(-\frac{1}{T}||\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_j||^2 + \beta s(\mathbf{x}_i, \mathbf{y}_j)\right). \quad (7)$$

B. TPS-RPM with Learned Labels

To use predictions from the CNN as a prior, we set

$$s(\mathbf{x}_i, \mathbf{y}_j) = \sum_{c \in C} \min(p_c(\mathbf{x}_i), p_c(\mathbf{y}_i)) \quad (8)$$

where C is the set of all possible patch classes, and $p_c(\mathbf{x}_i)$ and $p_c(\mathbf{y}_j)$ are the class probabilities of the 64-by-64 patch centered at $\mathbf{x}_i$ and $\mathbf{y}_j$, respectively. For the rope, $p_c(\mathbf{x}_i)$ is a probability distribution over four values, and for the towel it is over ten values. This similarity measure is equivalent to the histogram intersection between the two probability distributions, which has been used successfully as a measure of image similarity in prior work [37].

VI. Experiments

We evaluate the benefit of using appearance priors with TPS-RPM in the context of manipulation of rope and cloth. In particular, we are interested in whether our approach is able to find higher quality warp functions. Higher quality warp functions result in a more accurate ranking of demonstrations for a given test scene, and the transferred trajectory will be more likely to succeed in the test scene.

The purpose of ranking demonstrations is to determine which demonstration to transfer to the test scene. Schulman et al. [2] first calculate the warp from each demonstration scene to the test scene, and then rank demonstrations by increasing warp bending cost. We use a similar approach, but instead use TPS-RPM with appearance priors to calculate the warps, which are then ranked based on a linear combination of bending cost and appearance-based metrics.

A. Experiments with Rope

(i) Experimental Setup

Our dataset contains 322 rope configurations, which correspond to starting states for the three steps in tying an overhand knot. 32 of the 322 rope configurations in our dataset are randomly selected to be target rope configurations, and the rest are source configurations. We obtain ground truth point correspondences and warp functions for this dataset by first manually labeling the overcrossings, undercrossings, and endpoints located sequentially along each rope configuration. These points define a given rope's *crossing configuration* [5], which uniquely determines the topology of the rope.

For each (source, target) pair with the same crossing configuration, we then automatically calculate the ground truth warp as follows. For each pair, we divide each rope into several segments with boundaries as either endpoints or crossings, and then resample each segment to have the same number of points. After this resampling, each pair of ropes will have the same number of points and the correspondences between the two point clouds are completely specified. Once we have these ground truth point correspondences, we find the warp function that minimizes the TPS objective (Equation 1), and set that as the ground truth warp function.

(ii) Matching Ground Truth Point Correspondences

For each (source, target) pair of rope configurations, we compare the warping function learned by TPS-RPM with and without appearance information.

Given M correspondence points between a source s and target t rope configuration, we define the measure $D_{\text{points}} = \sum_{i=1}^{M} ||\mathbf{f}(p_i^s) - p_i^t||_2$, where (p_i^s, p_i^t) are the i^{th} pair of ground truth point correspondences. D_{points} measures the registration error of the computed warp function $\mathbf{f}$ on matching the corresponding points. Table III shows the average registration error and bending cost across all pairs of source and target

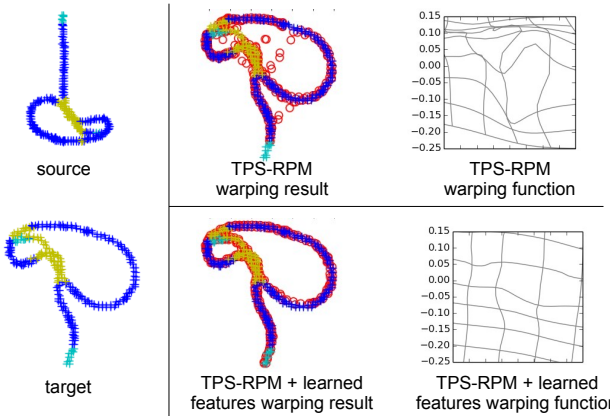


Fig. 3: Our approach uses learned appearance information, in the from of appearance priors, to find a higher quality warp function from source to target points—as shown for this pair rope configurations, in which regions of interest are endpoints and crossings. The colors cyan, yellow, and blue indicate the predicted endpoints, crossings, and normal regions in the rope, respectively. Red circles indicate locations of the warped source points. The ideal warp from source to target includes a rotation by 180°, and warping with appearance priors includes this rotation. By contrast, warping without appearance priors reaches a poor local optimum, as seen by the warped source points that are not close to any target points (top row, middle picture).

rope configurations. As expected, using appearance information improves matching of ground truth corresponding points, and decreases the bending cost of the final warp found by TPS-RPM—making it more likely that the transferred trajectory will succeed when executed in the test scene.

Figure 3 is an example of the improvement gained by using appearance priors when calculating the warp between two RGBD rope point clouds: the points corresponding to the crossings and endpoints of the two ropes are better matched, and the warp function is more rigid as well.

Knot Demo State	**Warp Quality Measure**	**TPS-RPM**	**TPS-RPM + Prior**
Step 1	D_{points} (in cm)	1.42	**1.08**
	bending cost	0.69	**0.64**
Step 2	D_{points} (in cm)	1.00	**0.62**
	bending cost	1.05	**0.91**
Step 3	D_{points} (in cm)	3.91	**2.64**
	bending cost	1.98	**1.74**

TABLE III: A comparison of the quality of warping functions calculated using TPS-RPM, with and without incorporating appearance priors. D_{points} represents the accuracy of the warping function $\mathbf{f}$ in matching ground-truth point correspondences between the source and target rope configurations. The bending cost measures how non-rigid $\mathbf{f}$ is; a more rigid warp (and thus lower bending cost) is preferred. These numbers are averaged over all (source, target) pairs belonging to each of the three demonstration steps in tying an overhand knot.

(iii) Demonstration Selection

For a given test scene, we obtain a ground truth ranking of the source configurations by ordering them by increasing bending cost of the ground truth warps from each source to the test scene. We use this to evaluate interpolated precision and recall for the rankings produced by TPS-RPM with and without appearance information, as follows. For each ranking V, interpolated precision is calculated for the top k elements, with k ranging from one to the total number of elements. Let $V(k)$ denote the set of top k elements, and let V_g denote the set of source demonstrations which are of the same topology as the target rope configuration. Then for a given ranking V and for the top k elements, we calculate interpolated precision and recall as $\text{Precision}(k) = \max_{j \leq k}(|V(j) \cap V_g|)/j$, and $\text{Recall}(k) = (|V(k) \cap V_g|)/|V_g|$ [38]. For both variants of TPS-RPM, precision and recall are averaged across all 32 targets for each value of k.

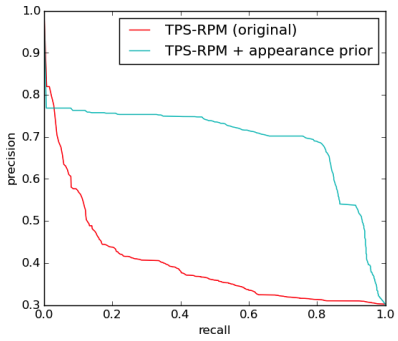


Fig. 4: Precision-recall curve for knot-tying demonstration selection, using TPS-RPM with and without appearance priors.

As seen in the resulting precision-recall curve (Figure 4), using appearance priors significantly improves demonstration ranking and thus selection.

B. Experiments with Cloth

(i) Ground Truth

We collected images and point clouds for 121 towel configurations that are commonly seen while folding the towel, with at least 15 towel configurations from each of the eight steps (Figure 6). We randomly designated 84 (70%) of these configurations as source (i.e., "demonstration" scenes) and the rest as target configurations.

We compare the same versions of TPS-RPM as for the rope: without any appearance information versus with learned labels. However, we are not able to calculate ground truth warp functions for pairs of towel configurations because it is difficult to accurately define ground truth point correspondences from one towel configuration to another.

(ii) Matching Ground Truth Corner Correspondences

Since we cannot obtain ground truth correspondences for all points between two towel configurations, we instead consider ground truth correspondences between points in regions of interest. Manipulation steps in folding a towel generally involve grabbing corners and/or edge midpoints of the towel. Thus, we define regions of interest to be the corners and midpoint of each edge of the towel point cloud. Table IV shows the following measures for TPS-RPM with and without learned labels, averaged over the pairs of configurations for each towel category: $D_{\text{corners}} = \sum_{i=1}^{4} ||\mathbf{f}(c_i^s) - c_i^t||_2$ and $D_{\text{midpoints}} = \sum_{i=1}^{4} ||\mathbf{f}(m_i^s) - m_i^t||_2$, where $\{c_i^s\}$ and $\{c_i^t\}$ are the four corners for source and target towels, respectively; $\{m_i^s\}$ and $\{m_i^t\}$ are the four

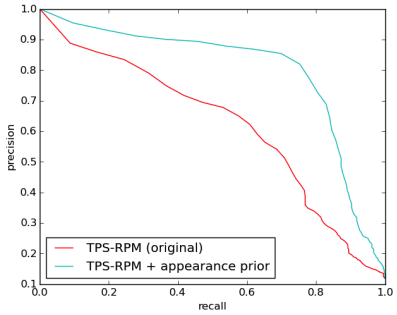


Fig. 5: Precision-recall curve for towel-folding demonstration selection, using TPS-RPM with and without appearance priors.

edge midpoints for source and target towels, respectively. As expected, using appearance information generally results in a better match of corners and edge midpoints, and a more rigid warp.

Towel Folding Demo State	Warp Quality Measure	TPS-RPM	TPS-RPM + learned labels
decrumple 1 (36 pairs)	corners	5.40	**5.23**
	midpoints	-	-
	bending cost	0.00355	**0.00349**
decrumple 2 (40 pairs)	corners	8.08	**8.02**
	midpoints	-	-
	bending cost	**0.00267**	0.00272
pick up corners (36 pairs)	corners	**14.30**	14.61
	midpoints	-	-
	bending cost	0.00240	**0.00238**
triangles (36 pairs)	corners	**3.96**	4.00
	midpoints	2.44	**2.42**
	bending cost	0.00106	**0.00105**
re-lay down (56 pairs)	corners	4.12	**4.10**
	midpoints	3.59	**3.57**
	bending cost	**0.00085**	0.00086
first fold (63 pairs)	corners	5.18	**4.82**
	midpoints	6.93	**6.46**
	bending cost	0.00108	**0.00100**
second fold (48 pairs)	corners	3.18	**2.90**
	midpoints	3.90	**3.02**
	bending cost	0.00058	**0.00053**
third fold (54 pairs)	corners	3.50	**3.20**
	midpoints	3.57	**3.10**
	bending cost	0.00077	**0.00073**

TABLE IV: A comparison of towel region-of-interest matching and warp bending cost for TPS-RPM, with and without appearance information. Please refer to Figure 6 for examples of each of these towel demonstration scenes, labeled with corner and midpoint points.

(iii) Demonstration Selection

Adding learned labels as appearance priors in TPS-RPM also leads to better towel folding demonstration rankings. Figure 5 shows the precision-recall curve, averaged across the 37 target configurations. Precision and recall are calculated in the same way as for the rope, with V_g denoting the set of towels of the same category as the target.

C. Evaluation of End-to-End Execution with a PR2 Robot

Ultimately, we are interested in whether our approach enables using LfD for real-world execution of a challenging deformable object manipulation task: towel folding. The original approach of considering only point clouds (and ignoring appearance information) in TPS-RPM runs into difficulties for towel folding because the resulting warp often does not match important regions such as corners and edges. When these mismatches happen, trajectory transfer is very likely to fail, since the steps in folding a towel depend on successful grasping of the corners and edges of the towel. In addition, when selecting the demonstration for a test scene with a flat, non-wrinkled towel, TPS-RPM without appearance information will not be able to distinguish among the first fold, second fold, and third fold starting states, without hardcoding scaling penalties.

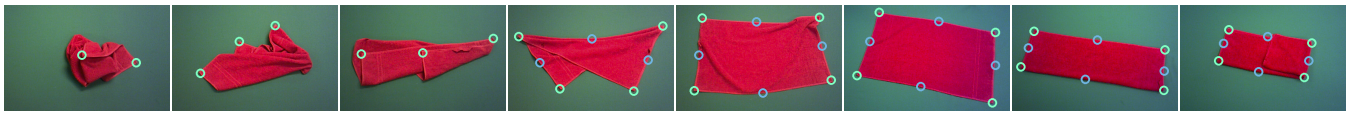
Incorporating appearance priors in TPS-RPM makes it possible to fold towels through LfD by matching regions of interest more accurately. Using our approach, a Willow Garage PR2 robot folded a towel successfully in six out of ten trials starting from fully crumpled towel configurations. Our website, mentioned in Section I, contains a recording of one of the successful executions. Trajectory transfer is done using TPS-RPM with learned labels, and corners are downsampled using a smaller voxel size, to improve precision while not increasing runtime significantly.

On the website is also a video containing examples of failure cases. The primary failure cases are caused by the lack of a demonstration that is similar enough to the test scene, which results in a poor warp found by TPS-RPM. When this happens, the robot either grasps at an incorrect location or completely misgrasps. The robot is generally able to recover from failures, unless the failure caused the towel to move out of view, or neither gripper grasped the towel.

VII. Conclusion and Future Work

We have presented a method for encoding appearance information for deformable objects and incorporating these learned labels to improve the performance of learning from demonstration. The labels are learned by a deep CNN trained on labeled patches sampled from demonstration images, and are then used as a prior on matching object point clouds in TPS-RPM. We evaluated our approach in the context of two typical instances of deformable object manipulation: tying knots in ropes and folding towels. We find using appearance priors in TPS-RPM improves demonstration selection, and can reduce error in trajectory transfer by guiding TPS-RPM towards more reasonable point correspondences. Our approach enables the first successful application of LfD to folding a towel from a fully crumpled state.

In future work, misclassification of patches (especially in the case of cloth) could be reduced by combining both depth and color images in patch classification. It would also be interesting to investigate the performance of our approach on objects with a variety of different patterns and textures, and under different lighting conditions; our approach should be able to be readily adapted to these variations. In addition, the method we proposed relies on supervised learning and requires a relatively large set of labeled patches. State-of-the-art unsupervised feature learning techniques such as sparse coding could help reduce dependency on labeled data.



Decrumple 1 Decrumple 2 Pick Up Corners Triangles Re-lay Down First Fold Second Fold Third Fold

Fig. 6: The eight demonstration steps we use for folding a fully crumpled towel. Green and blue circles denote the corners and edge midpoints, respectively, labeled for ground-truth point correspondences. These correspondences are used to evaluate our method in Table IV.

VIII. Acknowledgments

This research was supported in part by DARPA under a Young Faculty Award and under Award # N66001-15-2-4047. Sandy Huang was supported by a Chancellor's Fellowship.

References

[1] 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," in *IROS*, 2013.

[2] J. Schulman, J. Ho, C. Lee, and P. Abbeel, "Generalization in robotic manipulation through the use of non-rigid registration," in *ISRR*, 2013.

[3] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding*, vol. 89, no. 2-3, pp. 114–141, 2003.

[4] F. F. Khalil and P. Payeur, *Dexterous Robotic Manipulation of Deformable Objects with Multi-Sensory Feedback - a Review*. Robot Manipulators, Trends and Development, 2010, ch. 29, pp. 587–621.

[5] M. Saha, P. Isto, and J.-C. Latombe, "Motion planning for robotic manipulation of deformable linear objects," in *Experimental Robotics*, ser. Springer Tracts in Advanced Robotics, 2008, vol. 39, pp. 23–32.

[6] B. Frank, C. Stachniss, N. Abdo, and W. Burgard, "Efficient motion planning for manipulation robots in environments with deformable objects," in *IROS*, 2011, pp. 2180–2185.

[7] S. Rodriguez, X. Tang, J.-M. Lien, and N. Amato, "An obstacle-based rapidly-exploring random tree," in *ICRA*, 2006, pp. 895–900.

[8] S. Patil, J. Burgner, R. Webster, and R. Alterovitz, "Needle steering in 3-d via rapid replanning," *Robotics, Transactions on*, vol. 30, no. 4, pp. 853–864, Aug 2014.

[9] J. Smolen and A. Patriciu, "Deformation planning for robotic soft tissue manipulation," in *International Conferences on Advances in Computer-Human Interactions*, 2009, pp. 199–204.

[10] D. Berenson, "Manipulation of deformable objects without modeling and simulating deformation," in *IROS*, 2013, pp. 4525–4532.

[11] 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." in *ICRA*, 2010, pp. 2308–2315.

[12] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer Berlin Heidelberg, 2008, pp. 1371–1394.

[13] S. Calinon, T. Alizadeh, and D. G. Caldwell, "On improving the extrapolation capability of task-parameterized movement models," in *IROS*, 2013.

[14] P. Kormushev, S. Calinon, and D. G. Caldwell, "Reinforcement learning in robotics: Applications and real-world challenges," *Robotics*, vol. 2, no. 3, pp. 122–148, 2013.

[15] H. Mayer, I. Nagy, A. Knoll, E. Braun, R. Lange, and R. Bauernschmitt, "Adaptive control for human-robot skilltransfer: Trajectory planning based on fluid dynamics," in *ICRA*, 2007, pp. 1800–1807.

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

[17] 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," in *IROS*, 2013.

[18] B. Balaguer and S. Carpin, "Combining imitation and reinforcement learning to fold deformable planar objects," in *IROS*, 2011, pp. 1405–1412.

[19] H. Lombaert, Y. Sun, and F. Cheriet, "Landmark-based non-rigid registration via graph cuts," in *Image Analysis and Recognition*, ser. Lecture Notes in Computer Science, M. Kamel and A. Campilho, Eds. Springer Berlin Heidelberg, 2007, vol. 4633, pp. 166–175.

[20] K. Rohr, H. Stiehl, R. Sprengel, T. Buzug, J. Weese, and M. Kuhn, "Landmark-based elastic registration using approximating thin-plate splines," *Medical Imaging, IEEE Transactions on*, vol. 20, no. 6, pp. 526–534, June 2001.

[21] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 5, pp. 978–994, May 2011.

[22] D. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, vol. 2, 1999, pp. 1150–1157.

[23] V. Lepetit, P. Lagger, and P. Fua, "Randomized trees for real-time keypoint recognition," in *CVPR*, vol. 2, June 2005, pp. 775–781 vol. 2.

[24] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 4, pp. 509–522, Apr 2002.

[25] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, vol. 1, June 2005, pp. 886–893.

[26] E. Tola, V. Lepetit, and P. Fua, "Daisy: An efficient dense descriptor applied to wide-baseline stereo," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 5, pp. 815–830, May 2010.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[28] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *CoRR*, vol. abs/1409.0575, 2014.

[29] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *CoRR*, vol. abs/1311.2524, 2013.

[30] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[31] W. Ouyang and X. Wang, "Joint deep learning for pedestrian detection," *ICCV*, pp. 2056–2063, 2013.

[32] R. B. Girshick, F. N. Iandola, T. Darrell, and J. Malik, "Deformable part models are convolutional neural networks," *CoRR*, vol. abs/1409.5403, 2014.

[33] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans, "Reconstruction and representation of 3d objects with radial basis functions," in *SIGGRAPH*, 2001, pp. 67–76.

[34] G. Wahba, *Spline Models for Observational Data*. Philadelphia: Society for Industrial and Applied Mathematics, 1990.

[35] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[36] B. Combès and S. Prima, "A new efficient em-icp algorithm for non-linear registration of 3d point sets," INRIA, Research Report RR-7853, Jan. 2012.

[37] A. Barla, F. Odone, and A. Verri, "Histogram intersection kernel for image classification," in *International Conference on Image Processing*, vol. 3, Sept 2003, pp. 513–16.

[38] C. D. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.