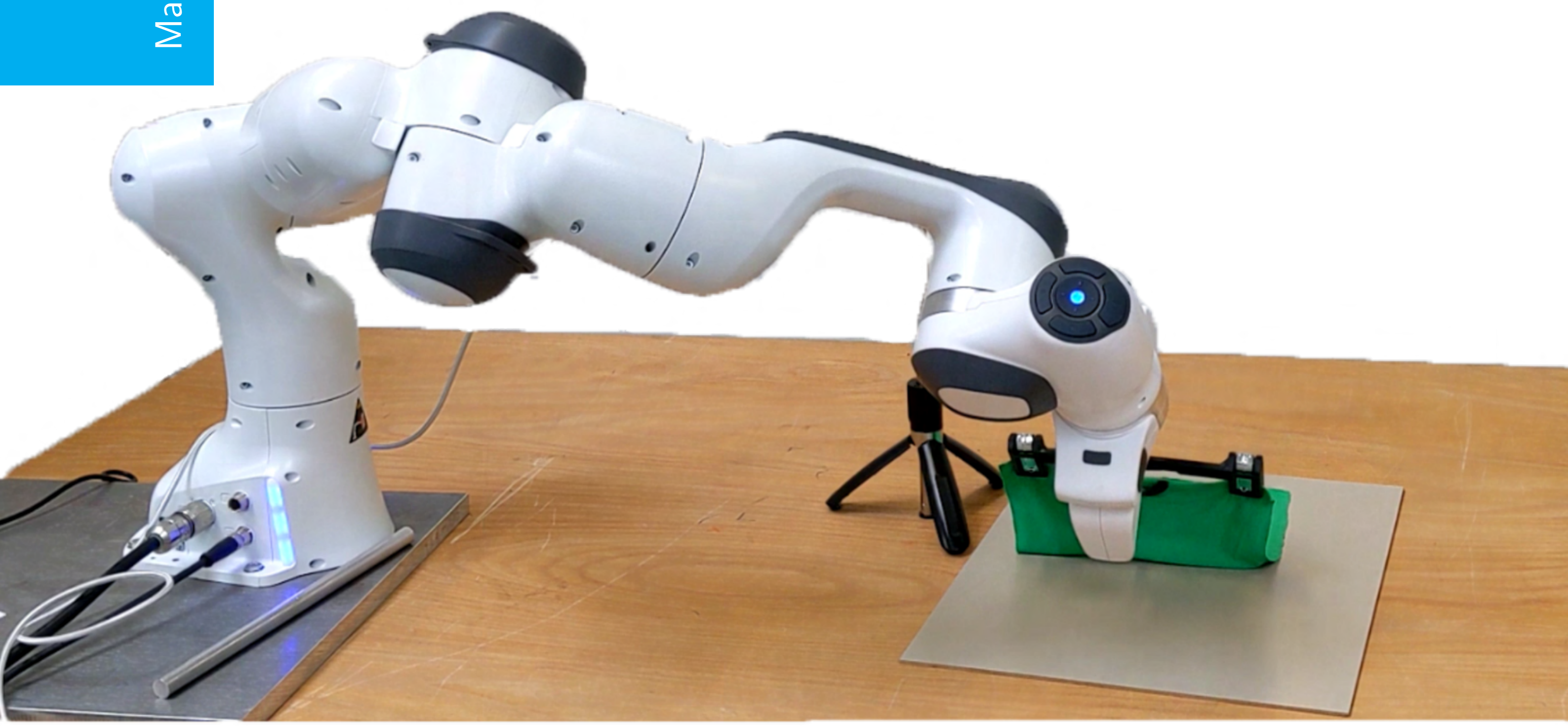
Wrinkle contraction direction: a useful feature for learning robotic fabric manipulation from demonstration

Chia-Yu Tsai

Master of Science Thesis



TU Delft Delft University of Technology

Wrinkle contraction direction: a useful feature for learning robotic fabric manipulation from demonstration

Master of science Thesis

For the degree of Master of Science in Mechanical Engineering at Delft University of Technology

Chia-Yu Tsai

October 5, 2021

Student number:	5004268	
Project duration:	November 1, 2020 – October 20, 2021	
Thesis committee:	Dr. Jens Kober,	TU Delft, Chair & Supervisor
	Dr. Jihong Zhu,	TU Delft, Daily supervisor
	Dr. Cosimo Della Santina	TU Delft, External member

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

This work was supported by the Honda Research Institute Europe GmbH as part of the project: Learning Physical Human-Robot Cooperation Tasks.

Abstract

Deformable objects manipulation (DOM) is largely considered an open problem in robotics. The complexity stems from the high degrees of freedom and nonlinear nature of the object configurations. In this thesis, we consider placing and flattening tasks for cloth-like objects. We propose a practical framework to place a cloth on a surface based on visual perception and human demonstrations. We present a novel feature, Wrinkle cOntRaction Direction (WORD), which extracts a stretching direction to flatten clothes from image and depth data. Furthermore, we integrate WORD and demonstrations into Gaussian Processes to learn a cloth placing policy. Simulation and robot experiment results are used to validate the performance of WORD and the proposed learning framework in this study. The results show that WORD efficiently captures wrinkles on the contact part of the cloth in the simulation as well as the real robot experiment. Besides, the proposed learning framework performs successful results in cloth placing and flattening.

A video of the experiments and execution of the tasks is available at https://youtu.be/iV2mAPqL7mA.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I am grateful to have Dr. Jens Kober to be my master thesis supervisor. I would like to thank my daily supervisor Dr. Jihong Zhu for his assistance during the writing of this thesis. It is my pleasure to work with both of you. During my master thesis project, I have learned so much more than I expected. The motivation and inspiration I got from this journey are extremely valuable. I would like to express my gratitude to my family and friends for their constant support. It would not have been possible without all of you.

蔡佳佑

Delft, University of Technology
October 5, 2021

Chia-Yu Tsai

Chapter 1

Introduction

In this chapter, we provide an introduction to this thesis. Section 1-1 introduces motivation, The current state-of-the-art is presented in section 1-2. The contributions of this thesis is mentioned in section 1-3. Finally, the thesis outline is presented in section 1-4.

1-1 Motivation

Robotic technology changes industrial automation and our daily lives significantly. Still, on most occasions, robots only work on simple repetitive tasks. Currently, most of the automation tasks are related to object manipulation and it is one of the key areas of research in robotics. Most robotic manipulation problems consider rigid object manipulation. The mature robotic manipulation of rigid objects brings valuable applications. However, in the real world, as shown in Figure 1-1, object rigidity does not always hold, for instance, cables, food products, clothes, etc. In order to manipulate deformable objects reliably, research is extensively done in recent years. In this thesis, we focus on investigating manipulation tasks of cloth-like objects.

In contrast to 6 degrees of freedom of rigid objects, due to infinite degrees of freedom of cloth-like objects, new challenges emerge. For example, it becomes more complex to sense and describe the configuration of the cloth in real-time, since a cloth-like object has infinite degrees of freedom and self-occlusion usually happens when observing a garment. Besides, it is also challenging to control the deformation of the cloth to complete the goal such as placing or folding.

Deformable objects manipulation (DOM) usually breaks down into sensing and control. Sensing is to obtain states to describe the object's configuration, control use this information to guide robot motion. Humans possess the intuition to efficiently infer the complex configuration of a deformable object and perform a suitable manipulation strategy. However, it is difficult to model the entire human behavior. Therefore, learning based on human demonstration data seems to be a promising approach.

(a) **(b)**

Figure 1-1: (a) Cloth folding [1], (b) Fruit picking [2]

The high number of DoF is another challenge for DOM. A simple way to constrain the shape and the degree of freedom of cloth is using surface contact. For example, when people are folding a cloth, it becomes a lot easier when the cloth is place flat on a surface instead of hanging in the air. With a good initial condition, humans and robots can recognize the cloth state easily and apply a simple strategy to manipulate the cloth into the desired configuration. If we could provide the initial condition that cloths are placed flat on a surface for robots, a more robust performance can be expected. A method to pick up cloth on the desired corners in the air from a random crumpled state is proposed by Cusumano et al. [3]. It is thus beneficial to develop a framework to place the cloth flat on a surface starting from picking the two consecutive corners in the air to provide a good initial condition for any kind of cloth manipulation task such as folding, ironing, etc. Developing this framework is the main focus of this thesis.

1-2 Literature review

For the thesis, a broad literature survey of current cloth state extraction methods and recent advances in cloth manipulation was performed, details of which are provided in chapter 2. In this study, we aim to develop a framework that is feasible in a real-world scenario with common sensors. Therefore, we summarize different cloth manipulation tasks correspond to minimal required sensing based on the literature survey in Table 1-1.

Task	Sensing techniques
Cloth state extraction [8][9][10]	Vision
Grasping points detection [11][12][13]	
Cloth unfolding [3][14][15][16]	
Cloth placing [17]	
Cloth flattening [4][18]	
Cloth ironing [19]	
Cloth folding [5][20][6][21]	
Clothing assistance [22][23][24][25]	Vision + Force

Table 1-1: Cloth manipulation tasks correspond to minimal required sensing

Some of the key findings and summaries of the survey are as follows:

- Vision is the main sensing modality in cloth manipulation.
- Visual descriptors are commonly used to extract the information from RGB-(D) images in order to get essential low-dimensional features efficiently.
- Techniques such as visual servoing and learning from demonstration have been used to execute the manipulation policy since these techniques are suitable for encoding complex behaviors.

Throughout the extensive literature survey, we found that although most of the robotic tasks require using cloth-like objects in contact with the table surface, yet no framework has specifically included the contact information in the task so far. Although Sun et al. [4] proposed a method using a dual-arm robot to flatten a crumpled napkin on a table based on the RGB-D sensors feedback and simulations, it did not solve the issue to place a garment flat on a surface starting from holding it in the air. Balaguer et al. [5] presented a method that a dual-arm robot executes a momentum fold that places a towel flatly on a surface then folds it. Combining imitation learning and reinforcement learning, several human demonstrations provide a good initial policy and make the learning algorithm converge efficiently without a deformable object model. The research successfully executes the momentum fold, including placing a towel flat on a surface. The main drawback of this research is that 28 reflective trackers are installed on the edges of the towel to track the positions. It is not applicable in the real world. From the extensive literature survey, we conclude that a research gap of cloth-like object manipulation is how to endow the robot the ability to place a garment on the surface as flat as possible in a cost-effective and easily applicable way.

Placing a garment flatly on a surface is an important topic that completes the pipeline of cloth folding. In addition, a flat garment on a surface creates a proper initial condition that increases the robustness of different manipulation tasks since flat clothes can be recognized and bring to the desired state easily.

1-3 Contributions

In this thesis, we aim to provide a generalized cloth placing framework based on LfD, visual descriptor, and surface contact information. The proposed method should be practical to apply to the real-world scenario with inexpensive sensors.

A common challenge in the field of deformable objects manipulation is state representation. Since the final goal of the cloth placing task is to lay it flat on a surface without any wrinkles and maximize the contact area, we present a visual descriptor, Wrinkle cOntRaction Direction (WORD), which extracts a stretching direction to flatten clothes from image data. WORD efficiently extracts useful information for cloth placing and flattening tasks. Since a model-based control strategy is not applicable in deformable manipulation, we propose using an LfD framework that utilizes human demonstrations to learn a generalized placing policy. This is because humans are good at perceiving the state of garments and manipulate them into the desired state. An LfD framework provides an easy way for non-experts to encode a sequence of complex movements. Our proposed method combines WORD and the LfD framework to generalize the cloth placing task for different cloth conditions.

The main contributions of this thesis are as follows:

- We present a novel feature, WORD, to describe the state of cloth based on wrinkles and contact information.
- We present an LfD framework based on GPs that takes WORD as input to generalize the cloth placing task.
- The proposed framework is applicable in a real-world situation with inexpensive devices.
- The proposed framework shows robustness against disturbance when performing a cloth placing task.

1-4 Thesis outline

The thesis is organized as follows: in the chapter 2, the background and related works for the study are covered. Chapter 3 introduces the novel feature, wrinkle contraction direction, and its experiment results in simulation and on a real robot. Chapter 4 introduces the proposed learning framework for cloth placing tasks and its experiment results. Finally, the study will be summarized with conclusions and potential future research directions in chapter 5.

Chapter 2

Background and related works

In this chapter, we provide background information about recent developments in cloth manipulation and the different approaches applied to manipulation tasks. A challenging and essential part of cloth manipulation is state extraction. According to the literature, vision is the most efficient and widely used sensing modality in deformable objects manipulation. In this study, we focus on low-dimensional cloth state extraction methods which extract useful states for specific tasks.

In section 2-1, recent advances in several cloth manipulation tasks such as cloth picking, placing, and folding, are presented.

In section 2-2, several visual descriptors commonly used in cloth manipulation are introduced.

After extracting the configuration of the garment, a control strategy corresponding to the manipulation task must be developed. Regardless of what kind of manipulation task it is, the behavior is usually complex for robots. LfD is able to generate robot motion from demonstration data without the need of explicitly programming the task. Since human manipulation strategy is usually difficult to model, but the data of the manipulation is easy to collect. LfD provides a convenient way to learn the manipulation pattern directly from the data. Thus attractive in DOM task. In section 2-3, we provide an extensive literature review of LfD.

2-1 Cloth manipulation tasks

The cloth manipulation tasks are complex and include multiple subsequential subtasks. In this section, we introduce some cloth manipulation tasks related to cloth placing and flattening as follows: cloth unfolding, cloth placing, cloth flattening, and cloth folding.

Cloth unfolding

The cloth unfolding task is bringing a cloth from a random state to a specific configuration so that it is easier to execute the following tasks. It is a common step for humans before manipulating a garment. Endowing robots this ability is an essential step for robotics cloth manipulation. Researchers have been focusing on this particular task.

Bersch et al. [14] presented research that a bimanual robot is able to grasp a T-shirt with fiducial markers from a random crumpled state and manipulated it into the goal state (grasp on the shoulders). A score function is defined to evaluate the current configuration and goal configuration based on a set of geometric features. The score function is trained using a support vector machine (SVM).

Cusumano et al. [3] proposed a markerless method to bring different clothing articles into desired configurations. A hidden Markov model (HMM) is used to estimate and track the configuration of a cloth-like object throughout a sequence of manipulations and observations. A dual-arm robot is able to manipulate the cloth into a known configuration then bring it into the desired configuration successfully. The block diagram of the unfolding process is shown in Figure 2-1. The final state of this research is the dual-arm robot holding the cloth in the air. To continue the folding process, it is necessary to place the cloth on a surface flatly. However, we found that the research aiming at cloth placement attracts less attention.

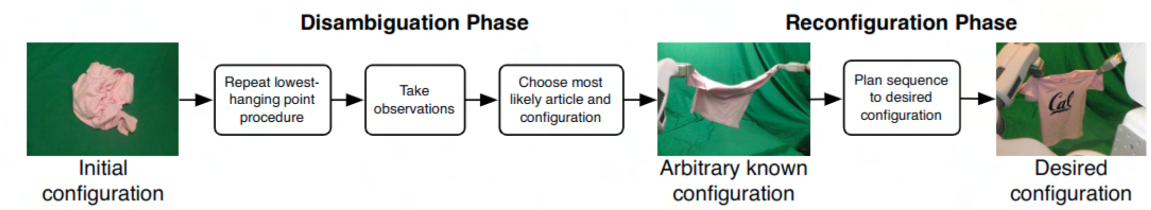


Figure 2-1: Block diagram of the unfolding process proposed by Cusumano et al. [3]

A more efficient method is presented by Doumanoglou et al. [15]. First, the robot grasps a random initial point of a crumpled cloth and the trained Random Forest algorithm classifies the target cloth. Next, the other arm grasps on the lowest point. Using a partially observable Markov decision process (POMDP) to recognize the first grasping point then grasp the second point. Finally, it is completely unfolded. However, it requires a robot to grasp the lowest point of the cloth and it might not be applicable for a large fabric or a small robot.

Cloth placing

Endowing robots the ability to place a cloth on a surface properly provides a decent initial state for subsequent tasks, such as cloth folding. Jangir et al. [17] used a deep reinforcement learning approach to solve dynamic cloth placing tasks. With few demonstrations, their approach successfully learned to do diagonal folding, sideways folding, and placing in a simulation environment. However, the proposed approach requires getting the positions and velocities of certain points on the edge of the cloth. Although these states are possible to extract through vision, self-occlusion usually happens in cloth manipulation. Therefore, this approach is not practical to use in a real-world situation.

Cloth flattening

Starting from a crumpled cloth lying on a surface, Sun et al. [4] presented a method for the dual-arm robot with high-quality RGB-D sensors to detect wrinkles and flatten the cloth. Using the feedback from these sensors to construct shape, topology, and wrinkle analysis then flattening parameters are estimated. Dual-arm planning is done to flatten the cloth by pulling on the side of the cloth. The figures of the flattening process and the detected wrinkles with planned displacements are shown in Figure 2-2.

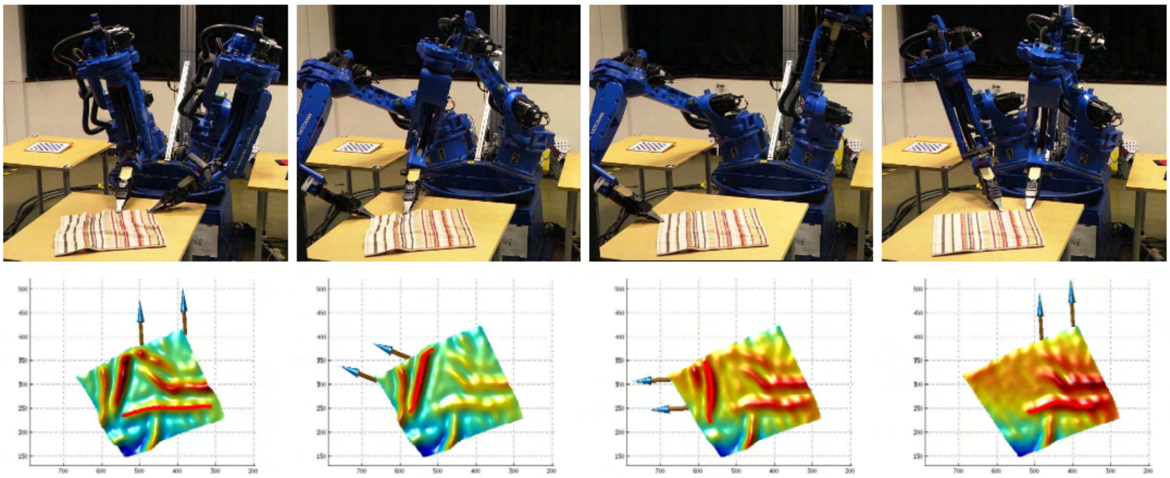


Figure 2-2: The first row shows a robot flattening a towel. The second row shows the detected wrinkles and the planned displacements. [4]

Seita et al. [18] applied deep imitation learning to learn policies to flatten and smooth fabric. Due to the complexity of fabric states, the trained policies are given color (RGB), depth (D), or color-depth (RGB-D) images, estimate pick points, and pull vectors to spread the fabric to maximize coverage. Simulation and robot experiments show that effective fabric smoothing policies can be learned. Comparing the performances of different trained policies, they found that depth sensing is a valuable addition to color alone.

From these studies, we found that depth information is an essential sensing modality to improve cloth flattening performance. Besides, we notice that current developments of robotic cloth flattening involve regrasping. However, regrasping is a relatively complex behavior for robots. If we could come up with a flattening strategy that only uses dragging and stretching, it could decrease the complexity of trajectory planning and possibly improves the robustness.

Cloth folding

Cloth folding is a common household chore. Research in robotic cloth folding approaches is presented in this section. To precept the state of the clothes, vision is widely used since it is able to classify the cloth-like object, detect wrinkles and grasping points, and verify the performance of folding. In most cases, cloth folding is done when the cloth-like object is placed flat on a surface.

Balaguer et al. [5] proposed a new learning algorithm that combines imitation and reinforcement learning to perform a towel folding task, which places half of the towel on a surface then folds it in half. The result is shown in Figure 2-3. A few human demonstrations are captured with the 28 tracker around the edges of the cloth. With an imitation learning algorithm, an initial folding policy is learned. Next, a reinforcement learning algorithm improves the placing policy. Due to imitation learning providing a good starting policy, the reinforcement learning algorithm converges extremely fast. Although the research presented a successful result, there are trackers around the edge of the towel and it requires a motion capture system to acquire data which is not a practical solution for cloth placing and folding tasks in the real world.

Figure 2-3: The successful motion of a dual-arm robot folding a towel with markers. [5]

Due to the high complexity to describe the configuration of a cloth-like object, Miller et al. [20] utilized the parameterized shape models to recognize the cloth articles, including towels, pants, short-sleeved shirts, and long-sleeved shirts. Each cloth article is represented as polygons and the desired sequence of folds is defined. Depending on the category of the cloth, the predefined fold sequence will be executed by the dual-arm robot PR2. During the folding process, according to the predefined folding sequence and visual feedback, new polygons models are fitted to track its progress over folds. A new polygonal model of a garment is introduced by Stria et al. [6]. The proposed algorithm (takes under 5 seconds) is more efficient than the parameterized shape model by Miller [20] (takes 30-150 seconds). The process of a robot folding a T-shirt with its polygonal model is shown in Figure 2-4.

In addition to focusing on the process of cloth folding, the trajectory optimization for cloth folding is proposed by Li et al. [21]. Cloth folding trajectories without optimization may create more wrinkles or cause the cloth to move away from the target location. A simulation environment that is comparable to the real world is created. In the simulation, minimizing the quadratic objective function that measures dissimilarity between simulated folded shape and user-specified shape iteratively, an optimized trajectory is obtained. The optimal trajectory obtained from the simulation is then applied to the real robot and gets successful results.

As stated in this section, we introduce different cloth manipulation tasks. Comparing with rigid object manipulation, deformable object manipulation is more challenging in both perception and manipulation strategy due to deformation. We notice that current research mainly uses cameras as sensors. Choosing the appropriate feature descriptors and combining them with a learning algorithm are widely used for cloth manipulation tasks. Besides, in order to gather more data, simulations with high fidelity are commonly used. As the complexity of manipulating cloth-like objects is significantly high, it is infeasible to formulate a model of the whole system to manipulate a cloth into the desired configuration. Therefore, low-dimensional visual descriptors that capture essential features are useful for cloth manipulation tasks. We will present an extensive review of visual descriptors in the next section.

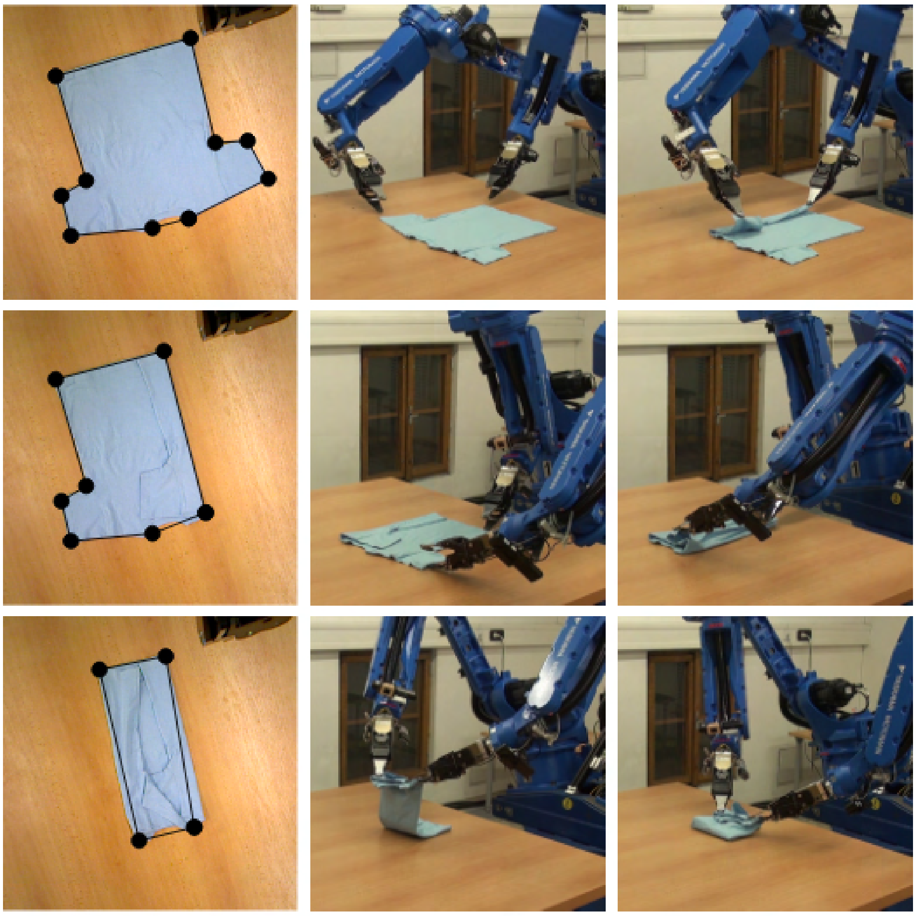


Figure 2-4: Robot folds a short-sleeved shirt. The left column shows the view from the camera with a fitted polygonal model. [6]

2-2 Deformable object representation with visual descriptors

In recent research, generally, the perception of a cloth-like object mainly utilizes vision. In special cases such as clothing assistance, involve working in close proximity to people, even touching humans. To ensure safety, sensory feedback both on tactile and force are required. In this section, we present some recent advances in vision-based low-dimensional cloth state extraction methods.

With the development of computer vision algorithms, more and more information can be extracted from RGB-(D) images. Visual feedback makes robots capable of identifying the location of a cloth-like object, selecting possible grasping points, estimating current configuration, etc. Moreover, selecting a suitable descriptor can efficiently extract essential information for the specific task and reduce the dimension. We introduce some useful visual descriptors for extracting cloth states with several applications in the following.

- **Scale-Invariant Feature Transform (SIFT)**

  SIFT is a feature detection algorithm proposed by David G. Lowe [26][27]. SIFT is able to extract distinctive invariant features from images that are robust and reliable to match different conditions, such as different viewpoints, additional noise, or changes in illumination. In addition to matching features between different images, SIFT is also useful for object recognition. Willimon et al. [7] proposed a clothing classification method using various vision descriptors. SIFT is one of the descriptors that gathers useful 2D local texture information. As shown in Figure 2-5, after background removal, SIFT algorithm is applied to the target object. SIFT feature points can be found in Figure 2-5 with arrows. The arrows represent the orientation and magnitude of the feature point.

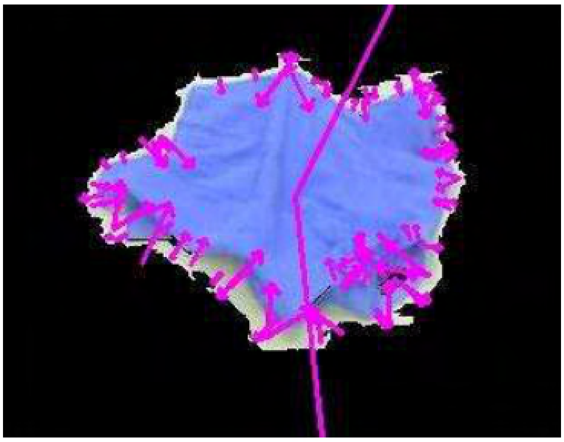


Figure 2-5: SIFT features overlay on the image. [7]

- **Histogram of Oriented Gradients (HOG)**

  HOG features are commonly used after Dalal and Triggs [28] presented their research on HOG for human detection. HOG is a powerful feature descriptor that can capture the edge of objects based on the gradient of the intensity of each pixel. Kampouris et al. [29] presented research that a dual-arm robot equipped with different sensors manipulates unknown fabrics and acquires multi-modal data. By applying various machine learning algorithms combined with Random Forests on HOG, it is able to recognize the fabric type, fabric pattern, and material.

- **Histogram of Oriented Wrinkles (HOW)**

  Jia et al. [30][31] presented the algorithm to compute the HOW features. HOW features are useful to describe the shape variation of highly deformable objects. HOW features are computed by applying Gabor filters with multiple orientations and wavelengths then extracting low-frequency and high-frequency components. Finally, the HOW features can be represented as a single-column vector. Jia et al. [30] used HOW features for the offline training to construct a visual-feedback dictionary, which stores a mapping between the HOW features and the velocity of the end-effector. During the run-time, the visual servoing combining with the visual-feedback dictionary realizes human-robot collaborative cloth manipulation tasks. The dual-arm robot is able to manipulate the garment into the desired configuration in collaboration with a human.

As mentioned above, we introduce three feature descriptors that are commonly used in the cloth manipulation field. SIFT features are suitable for cloth classification since they can efficiently capture distinctive features under different conditions. HOG features capture the edge of objects but clothes are highly deformable and the shapes are variant. Research shows that after bringing a garment into the desired configuration, HOG features combining with machine learning algorithms can successfully recognize the garment type [29]. HOW features mainly capture the deformation and wrinkles of objects. Based on the HOW features, robots are able to control the deformation of clothes by minimizing the error between the HOW features of the current configuration and the desired configuration. These descriptors successfully compute the low-dimensional features of cloth-like objects from RGB-(D) data, otherwise it is infeasible to represent the infinite degrees of freedom of a garment. Utilizing the suitable feature descriptor based on the cloth manipulation task could accurately capture the essential information and increase computational efficiency. Besides, integrating different feature descriptors properly might increase the success rate of the task.

2-3 Learning from demonstration (LfD)

Traditionally, if we want a robot to perform a task, it is required to encode all the desired movements manually. LfD provides a much easier way to teach robots from human demonstrations. It facilitates non-expert robot programming and enables robots to learn how to react differently depending on different scenarios.

A well-established LfD approach is trajectory-based methods. These methods encode robot movement policies by extracting trajectory patterns from demonstrations using various probabilistic methods, e.g. Dynamical Movement Primitives (DMPs) [32], Gaussian Mixture Regression (GMR) [33], or Gaussian Process Regression (GPR) [34]. These approaches have proved successful at learning and generalizing the desired trajectories in various scenarios.

DMPs use multiple primitive actions to form a complex movement. Complex movements consist of point attractors using DMPs. Therefore, DMPs guarantee convergence to a given target. Besides, DMPs are flexible to create complex behaviors by tuning the weight coefficients of basis functions. Kober et al. [35] use DMPs which incorporate perceptual coupling to learn ball-in-a-cup motion from demonstrations. However, due to its definition of trajectory dynamics, DMPs introduce many open parameters including the number of basis functions and their weighting coefficients.

GMR utilizes the Gaussian conditioning theorem to predict the distribution of output data given input data [36]. Unlike DMPs, GMR avoids defining trajectories via specific functions. Research shows that GMR is not a well-established technique adapting learned skills [37].

Gaussian process regression is a probability distribution over possible functions that fits a set of points [38]. It is a Bayesian non-parametric regression method which provides the means for inferring prediction and uncertainty with a specific mathematical definition. Comparing with GMR, which provides a generative model, GP generates a discriminative model. Generative models require fewer data to train, however, their generalization performance is often poorer than the performance of discriminative models [39]. Therefore, in the field of trajectory learning, GP is a more suitable method for modulating the learned policy than GMR [40]. Schneider et al. [34] presented an LfD framework based on GP. With few demonstrations, the robot learns to grasp a cup under the coffee machine and press the button on the machine with good generalization.

According to the result of an extensive literature survey on LfD, we select GP as the LfD method to encode cloth placing and flattening trajectories. We will give an introduction of GP and our proposed learning framework in section 4-1.

Chapter 3

Wrinkle cOntRaction Direction (WORD)

In this chapter, we provide an introduction of the proposed feature, WORD in section 3-1. In section 3-2 and section 3-3, we present the simulation experiment and the robot experiment results.

3-1 Methods

In this section, we present the motivation of WORD and the steps to compute the WORD from the depth camera stream.

3-1-1 Motivation

WORD is inspired by the HOW features presented by Jia et al [30]. HOW features are useful to describe the shape variation of highly deformable objects. The features are computed by applying 2D Gabor filters [41] on RGB images with multiple orientations and wavelengths then extracting low-frequency and high-frequency components. Gabor filter is a linear filter used for texture analysis. It is a Gaussian kernel function modulated by a sinusoidal plane wave oriented at an angle. A Gabor filter can be represented as:

$$g(x, y; \lambda, \phi, \sigma, \gamma, \theta) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \sin\left(2\pi \frac{x'}{\lambda} + \phi\right), \tag{3-1}$$

where $x' = x\cos\theta + y\sin\theta, y' = -x\sin\theta + y\cos\theta$, λ represents the wavelength, ψ is the phase offset, σ is the standard deviation of the Gaussian envelope, γ is the spatial aspect ratio, and θ is the orientation of the normal to the parallel stripes of a Gabor function.

From the processes of computing HOW features, we found that Gabor filters are useful in extracting patterns of images in the selected wavelength and orientation. This results in the enhancement of the ridges, wrinkles, and edges. HOW features utilize multiple Gabor filters and stack the feature matrix to a vector with hundreds of elements. From the HOW features, we are able to infer the number of wrinkles, the directions of wrinkles, and the distribution of wrinkles in the image.

Intuitively, if there is a wrinkle on a cloth and its direction is parallel to the x axis, a person will probably pull or stretch the cloth in the perpendicular direction of the wrinkle, which is the y axis, to flatten the cloth. Based on this intuition, we develop WORD to gives the proper stretching direction based on the majority of wrinkles direction by using parts of the HOW features algorithm. WORD simplifies the cloth configuration into a 2-dimensional vector based on the wrinkles' orientation. In addition, WORD takes the contact information as input to extract the wrinkles on the contact part of the cloth. Reducing the feature vector dimension from a few hundred to 2 increases the computation efficiency and avoids additional noise. We present the processes of computing WORD in the following section.

3-1-2 WORD computation

In this section, we provide an introduction to the processes of computing the proposed visual descriptor, WORD. It consists of four steps: foreground segmentation, contact detection, wrinkles extraction, and WORD computation.

Foreground segmentation

Since we aim to detect the wrinkles on the cloth, it is essential to extract the cloth from images. In practice, we can apply an HSV thresholding to extract the cloth by its color in the image.

Contact detection

After detecting where the cloth is in the image, the next step is to find which part of the cloth contacts the surface. In order to capture the contact part of the cloth, we can set a depth threshold on the depth map captured by the depth camera to detect contact. Combining with the previous step, we can successfully extract the contact part of the cloth from images. An example is shown in Figure 3-1. Figure 3-1a shows the original image captured by the camera and Figure 3-1b shows the contact part of the cloth in the original image.

Wrinkles extraction

The purpose of this step is to extract wrinkles in different orientations on the contact part of the cloth. Similar to the HOW features, we apply n Gabor filters with orientations in θ_n on the image to extract wrinkles. After applying Gabor filters, this results in the enhancement of the ridges, wrinkles, and edges in a specific direction, as shown in Figure 3-2. However, we are only interested in the wrinkles on the contact part of the cloth, we use the results

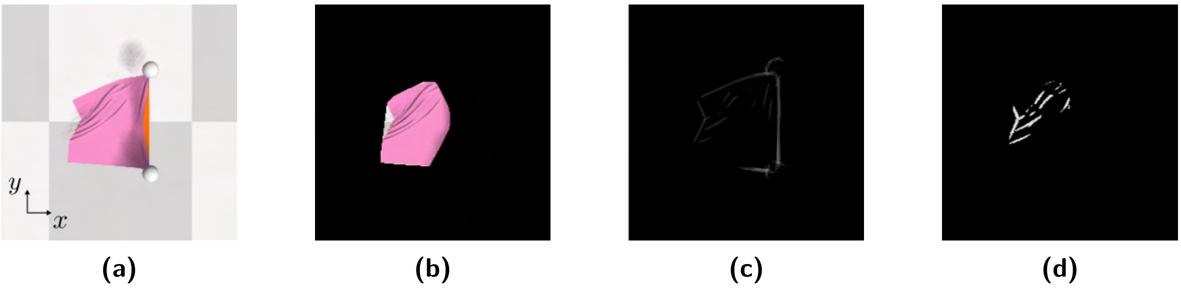


Figure 3-1: (a) Original image, (b) The contact part of the cloth in the original image, (c) Wrinkle extraction image in 8 directions with parameters: $\sigma = 4,\ \lambda = 8,\ \gamma = 0.1,\ \psi = 10,$ kernel size $=$ 12, (d) Extracted wrinkles on the contact part of the cloth. WORD is $[0.960, -0.280]$ in this image.

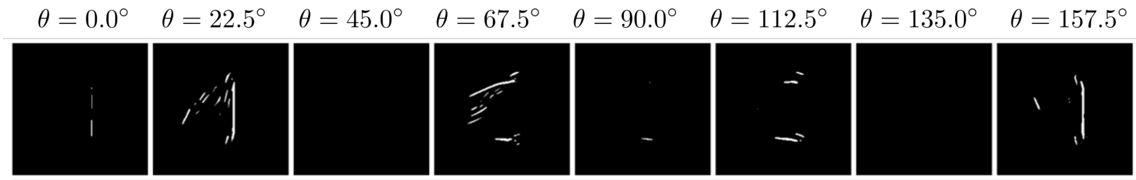


Figure 3-2: Raw output images after applying 8 Gabor filters in multiple orientations

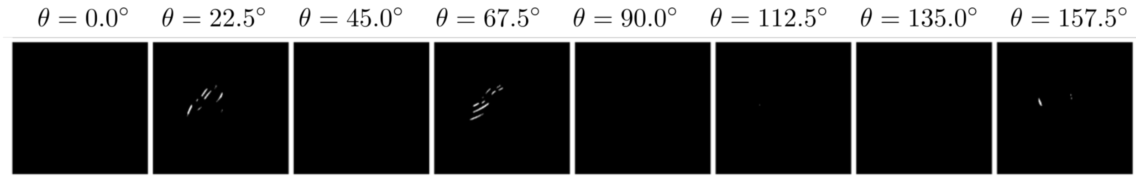


Figure 3-3: Wrinkles extraction of each direction on the contact part of the cloth

from previous steps to extract wrinkles in the specific area, as shown in Figure 3-3. The output image shows a larger pixel value if the pixel contains wrinkles. If there are no wrinkles detected in that orientation, all pixels have zero value in the output image. To visualize all the extracted wrinkles in an image, we stack n output images from Gabor filters. The result of wrinkles extraction of the original image is shown in Figure 3-1c. We can notice that the ridges, wrinkles, and edges are highlighted in white. Since we only want to detect the wrinkles on the contact part of the cloth, we apply the mask from the previous step again to capture the wrinkles in the region of interest, as shown in Figure 3-1d.

WORD computation

Once we get n output images from Gabor filters, we can infer the number of wrinkles in each direction from the summation of pixel values. The summation becomes larger when there are more wrinkles in that direction. Thus we can stack all the summations of pixel values in a n-dimensional vector $\boldsymbol{L}$ and formulate a weight vector $\boldsymbol{w}$ as:

$$\boldsymbol{w} = \frac{\boldsymbol{L}}{\max(\boldsymbol{L})}. \tag{3-2}$$

As mentioned in the previous section, intuitively, stretching in the direction perpendicular to the wrinkle can effectively flatten the cloth. Therefore, if a direction contains most of the wrinkles, it is reasonable to assign a larger weight to stretch in its perpendicular direction. We can formulate the WORD ($\boldsymbol{d}$) as:

$$\boldsymbol{d} = \sum_{i=1}^{n} w_i \boldsymbol{v}_i(\theta_n), \tag{3-3}$$

where w_i represents the i^{th} element in $\boldsymbol{w}$ and $\boldsymbol{v}$ is a unit vector perpendicular to θ_n direction. Finally, the WORD is a normalized vector $\boldsymbol{d}$. After applying the WORD computation method, the WORD of the Figure 3-1 is $[0.960, -0.280]$. The WORD is approximately perpendicular to the direction of the majority of wrinkles. The result shows that WORD successfully indicates a proper stretching direction for flattening the cloth.

In summary, in order to compute WORD, the first step is to capture the cloth and contact information for the image. Secondly, we extract wrinkles in multiple orientations and get the enhancement of wrinkles in each direction. Next, we compute a weight for each orientation based on the summation of pixel values. Finally, the summation of weight times the corresponding direction gives the direction of WORD. The algorithm for computing WORD is shown in Algorithm 1.

Algorithm 1 Computing WORD

Input: image $\boldsymbol{I}$ of size $(w_{\boldsymbol{I}},\ h_{\boldsymbol{I}})$, cloth mask $\boldsymbol{C}$ and depth mask $\boldsymbol{D}$ of size $(w_{\boldsymbol{I}}, h_{\boldsymbol{I}})$, number of orientations n.
Output: WORD vector $\boldsymbol{W}$
for $i = 0, 1, \ldots, n$ **do**
  $\theta_n = i \times \pi/n$
  Get Gabor kernel in direction θ_n
  Apply the Gabor kernel on the image $\boldsymbol{I}$ to get wrinkles in direction θ_n
  Apply the cloth mask and depth mask on the output image to get wrinkles on contact part of the cloth
  Stack the summation of pixel values in an n-dimensional vector $\boldsymbol{L}$
end for
$\boldsymbol{w} = (\boldsymbol{L}/\max(\boldsymbol{L}))$ ▷ $\boldsymbol{w}$: weight vector in each direction
$\boldsymbol{d} = \sum_{i=1}^{n} w_i \boldsymbol{v}_i(\theta_n)$ ▷ $\boldsymbol{v}$: unit vector perpendicular to θ_n direction
$\boldsymbol{W} = \boldsymbol{d}/\mathrm{norm}(\boldsymbol{d})$
return $\boldsymbol{W}$

Compared with HOW features, WORD significantly reduces the dimension to describe the cloth state. In our problem, the goal of cloth placing/flattening is to minimize the wrinkles and maximize the contact area, combing WORD and the contact portion of the cloth can provide sufficient information of the cloth state for this specific task. Later in subsection 3-2-2 we will present simulation results which compare WORD with baseline approaches to show its efficiency in removing the wrinkles.

3-2 Simulation experiment

We validate our proposed approaches in an open-source simulation environment with high-fidelity. In this section, the simulation environment and the validation tasks are introduced in detail.

3-2-1 Simulation experiment setup

In this section, we present the introduction of the simulation environment, SoftGym, and the customized cloth manipulation environment.

Simulation environment: SoftGym

We use SoftGym [42] as the main simulation environment to validate our proposed approaches. The SoftGym is a benchmark environment for deformable manipulation, including several fluid, cable, and cloth manipulation tasks. To simulate the behavior of deformable objects, SoftGym builds on top of the Nvidia FleX physics simulator. The Nvidia FleX physics simulator models deformable objects in a particle and position-based dynamical system. Objects are represented by a set of particles and the internal constraints among these particles. Using this generalized framework, the physics simulator can model different categories of deformable objects with corresponding parameters.

Cloth manipulation environment

An environment with an infinitely large surface is created in the SoftGym. Two white balls represent two independent pickers picking on two consecutive corners of the cloth. Initially, both pickers pick the cloth perfectly and hanging in the air, as shown in Figure 3-4.

In order to move the picker and pick up the cloth, an action space is defined in SoftGym. For our experiment, the action space for each picker is (d_x, d_y, d_z, p), where d_x, d_y, d_z determine the movement of the picker and p determines whether the picker is activated (picking cloth, $p \geq 0.5$) or deactivated (not picking cloth, $p < 0.5$). In this study, we only consider the case that the pickers constantly picking the cloth. We do not consider regrasping the cloth in this study. Therefore, in our experiment, we set $p = 1$ so that pickers always hold on the corner of the cloth.

3-2-2 Validation of WORD performance

In section 3-1, we introduce WORD in detail and show successful results in Figure 3-1. In this section, we perform an experiment in SoftGym to compare WORD with two baseline approaches to show its efficiency in removing wrinkles.

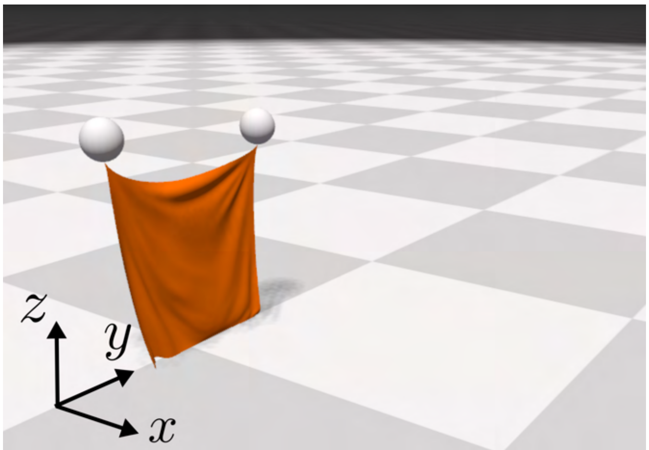


Figure 3-4: Cloth manipulation environment in SoftGym

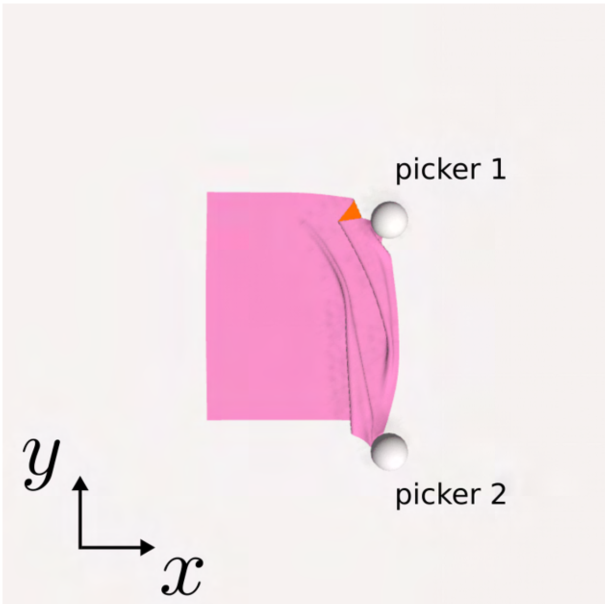


Figure 3-5: The initial state of the cloth

Wrinkle removal in a 2D plane using WORD

Knowing that WORD is able to indicate a proper stretching direction according to wrinkles, we perform an experiment using WORD to flatten a crumpled cloth on a surface. Initially, two independent pickers hold two consecutive corners of the cloth hanging in the air. We execute a series of hard-coded actions to place the cloth on the surface with several wrinkles. From the ground truth positions of cloth particles, the initial contact portion is around 74.00%. The initial condition of the cloth is shown in Figure 3-5.

The goal is to use WORD to pull the cloth to the right ($+x$ direction) and flatten it. We design a wrinkle removal controller based on WORD to test the performance of WORD. Algorithm 2 shows the control logic of the wrinkle removal controller based on WORD.

Algorithm 2 Wrinkle removal controller based on WORD

while Contact portion $< 100\%$ **do**
  $\boldsymbol{W} \leftarrow$ WORD of the current image
  $\boldsymbol{W} = (W_x, W_y)$ ▷ Decompose $\boldsymbol{W}$ in x and y direction
  if $W_y > 0$ **then**
    Move picker 1 in $\boldsymbol{W}$ direction
  else
    Move picker 2 in $\boldsymbol{W}$ direction
  end if
end while

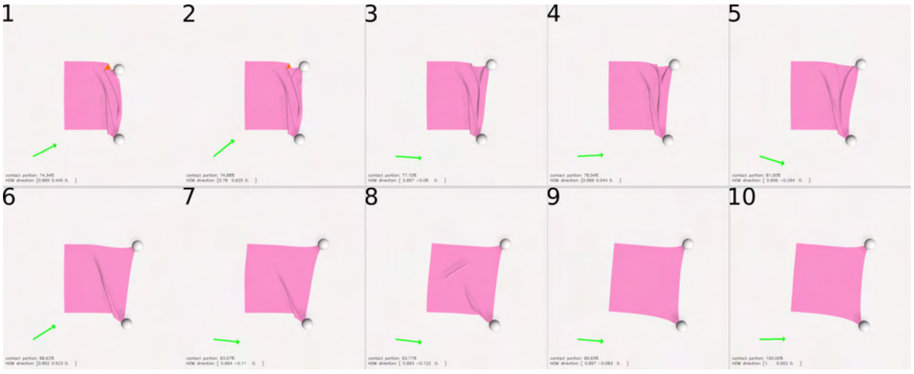


Figure 3-6: Proposed method: Wrinkle removal based on WORD (green arrows indicate the WORD of each step)

Two baseline approaches are performed to compare the flattening performance. The first baseline approach to flatten a crumpled cloth is using both pickers to stretch to the $+x$ direction. The other baseline approach is executing random movements on both pickers.

The simulation experiment results of using wrinkle removal controller based on WORD is shown in Figure 3-6. In the 1^{st} frame of Figure 3-6, there are several wrinkles and the contact portion is 74.34%. The controller detects the WORD of each frame and moves the corresponding picker in that direction. After a few movements, the WORD-based controller reduces the wrinkles and the contact portion reaches 100.00% in the 10^{th} frame. With a similar initial condition, the experiment result of the first baseline approach (stretching toward the right) is shown in Figure 3-7, the final contact portion is 92.67% caused by an obvious wrinkle between pickers. Figure 3-8 shows the result of the second baseline approach (executing random movement on both pickers). It shows that the cloth state stays almost the same and the final contact portion is 73.33%. The results show that using WORD descriptor can effectively flatten a crumpled cloth.

3-3 Robot experiment

Besides the validation in a simulation environment, we also perform robot experiments to test the performance of our proposed methods. In this section, we introduce the robot experiment in detail.

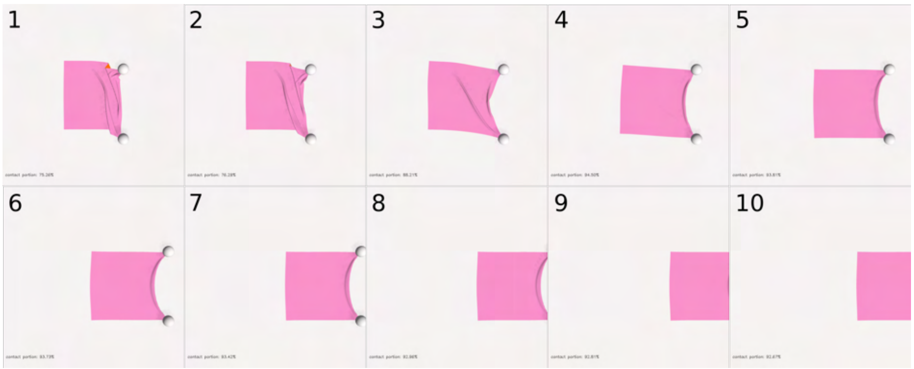


Figure 3-7: Baseline method 1: Pickers move toward the right to remove wrinkles

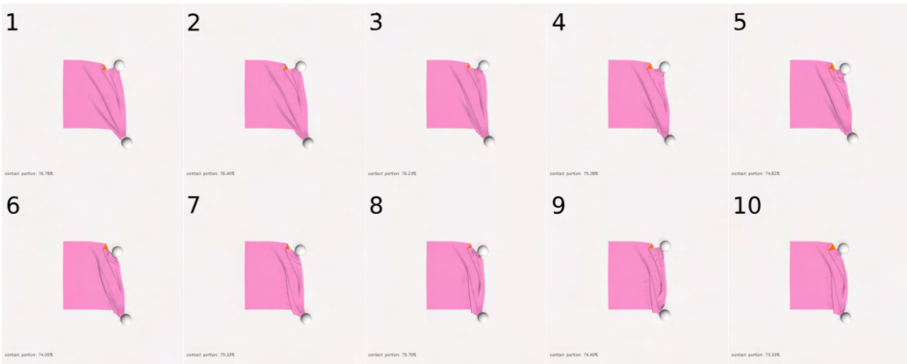


Figure 3-8: Baseline method 2: Pickers move randomly to remove wrinkles

3-3-1 Robot experiment setup

We use a 7 DoF Franka-Emika Panda with an impedance controller to manipulate the cloth and an Intel RealSense depth camera D435 to capture the state of the cloth. The translational stiffness is set as $400N/m$. A ROS communication network is used for integrating the feedback and computing desired commands. Due to the practical limitations, only a single robotic arm is used for this study. Therefore, we utilize a cloth hanger with two clips to pick the two consecutive corners of the cloth, and the center of the hanger is mounted on the gripper of Franka-Emika Panda. In addition, to decrease the friction between the cloth and the table during the cloth placing task, smooth white cardboard is placed on the table. The experiment setup is shown in Figure 3-9. We use two rectangular clothes to perform robot experiments. The dimension of cloth 1 is $35cm \times 35cm$ and the dimension of cloth 2 is $21.5cm \times 27.5cm$. The material of cloth 1 is stiffer than cloth 2 and cloth 2 has a thinner thickness which makes it easier to have wrinkles. In the following experiments, we will use cloth 1 and cloth 2 to validate the generalization performance of the proposed method.

3-3-2 Validation of WORD performance

In the previous sections, we present the result that the WORD algorithm works in the simulation environment, SoftGym. To prove WORD is a practical visual descriptor, we perform

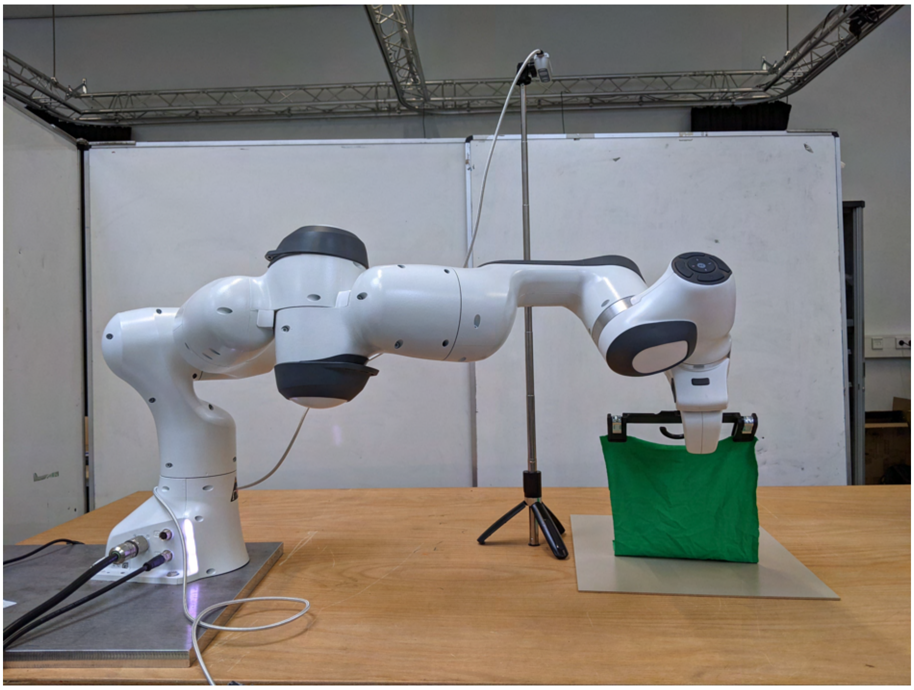
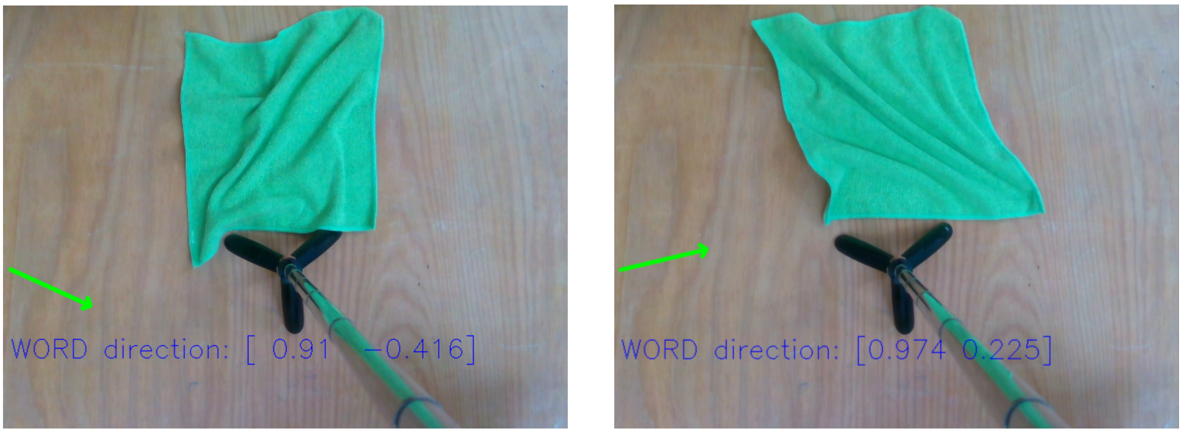


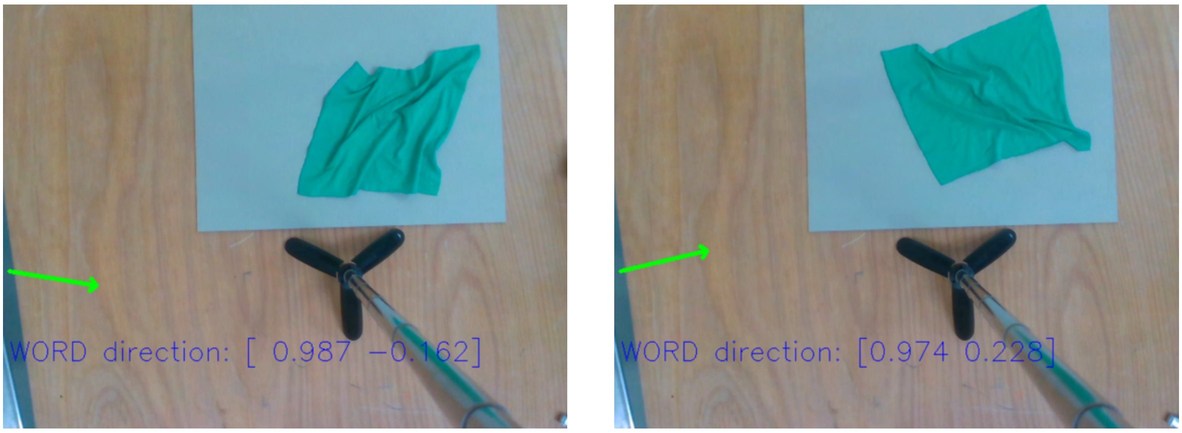
Figure 3-9: Robot experiment setup



(a) **(b)**

Figure 3-10: WORD extraction of cloth 1 with two wrinkles patterns (green arrows indicate the WORD of each image)

validations on the robot experiment setup. We place cloth 1 and cloth 2 on the table and manually create several wrinkles. Camera captured images and WORD vectors of two different scenarios are shown in Figure 3-10 and Figure 3-11. The results show that WORD effectively indicates a proper stretching direction which is perpendicular to the majority of wrinkles. Showing that using a depth camera can capture the required information to compute WORD and it works successfully.



(a) (b)

Figure 3-11: WORD extraction of cloth 2 with two wrinkles patterns (green arrows indicate the WORD of each image)

Chapter 4

Learning from demonstration using Gaussian processes

In this chapter, we provide an introduction of the proposed learning framework for cloth placing tasks in section 4-1. In section 4-2 and section 4-3, we present the simulation experiment and the robot experiment that validate the proposed learning framework.

4-1 Methods

Once the configuration of the cloth is detected, a robot has to execute a particular series of motions to reach the goal configuration. Deformable objects manipulation involves complex motions so it is difficult to encode such motions. We select an LfD as the main method to encode the cloth placing and flattening policy because it allows the user to program the motion without explicitly programming the motion. Besides, the placing task is a common task and the demonstrations are very easy to collect via kinesthetic teaching. A Gaussian processes regression is selected as the LfD method in our study. GP provides a parameter-free learning method and it enables a good generalization near the region of demonstrations. In addition, GP predictions interpolate the observations and provide the confidence level of each prediction point. We will give an introduction to GP in the next section.

4-1-1 Gaussian processes (GPs)

A Gaussian process is a probability distribution over possible functions that fits a set of points [38]. It is a Bayesian non-parametric regression method which provides the means for inferring prediction and uncertainty with a specific mathematical definition. It is defined by its mean function and covariance function. The mean function $m(\mathbf{x})$ and the covariance function (i.e. kernel function) $k(\mathbf{x}, \mathbf{x}')$ of a real process $f(\mathbf{x})$ are defined as:

$$
\begin{aligned}
m(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})] \\
k\left(\mathbf{x}, \mathbf{x}^{\prime}\right) &= \mathbb{E}\left[(f(\mathbf{x})-m(\mathbf{x}))\left(f\left(\mathbf{x}^{\prime}\right)-m\left(\mathbf{x}^{\prime}\right)\right)\right].
\end{aligned}
\tag{4-1}
$$

The kernel function $k\left(\mathbf{x}, \mathbf{x}^{\prime}\right)$ is a measure of similarity between two input vector $\mathbf{x}$ and $\mathbf{x}^{\prime}$, and it is an essential design choice in GP regression. A GP regression is denoted as:

$$
f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k\left(\mathbf{x}, \mathbf{x}^{\prime}\right)\right). \tag{4-2}
$$

Consider $\mathbf{y}=\{y_1, \ldots y_n\}$ as a set of noisy observations of a function $f(\mathbf{x})$ at input location $\mathbf{x}_i$

$$
y_i = f\left(\mathbf{x}_i\right)+\epsilon_i, \tag{4-3}
$$

where the inputs $\mathbf{x}_i \in \mathbb{R}^{D \times 1}$ are vectors and the outputs $y_i \in \mathbb{R}$ are scalars. ϵ_i represents a Gaussian noise with variance ${\sigma_n}^2$.

All the collected observations can be represented in a data set $\mathcal{D}=\{\mathbf{X}, \mathbf{y}\}$, where $\mathbf{X}=\left[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\right]^T \in \mathbb{R}^{N \times D}, \mathbf{y}=\left[y_1, y_2, \ldots, y_N\right]^T \in \mathbb{R}^{N \times 1}$ and N is the number of data points available. We are interested in incorporating the knowledge that the observed data set $\mathcal{D}$, provides about $f(\mathbf{x})$.

Let $\mathbf{m}(t)$ be the vector of the mean function evaluated at all observed points $\mathbf{x}$, and $K(t, t)$ be the matrix of the covariances evaluated at all pairs of training and prediction points $\mathbf{x}^*$. We can write the joint distributions of the observed values $\mathbf{y}$ and the predicted function values $\mathbf{f}^*$ under the prior as

$$
\left[\begin{array}{c}
\mathbf{y} \\
\mathbf{f}^*
\end{array}\right] \sim \mathcal{N}\left(\left[\begin{array}{c}
\mathbf{m}(\mathbf{x}) \\
\mathbf{m}\left(\mathbf{x}^*\right)
\end{array}\right],\left[\begin{array}{cc}
K(\mathbf{x}, \mathbf{x})+\sigma_n^2 I & K\left(\mathbf{x}, \mathbf{x}^*\right) \\
K\left(\mathbf{x}^*, \mathbf{x}\right) & K\left(\mathbf{x}^*, \mathbf{x}^*\right)
\end{array}\right]\right). \tag{4-4}
$$

By conditioning the joint Gaussian prior distribution on the observations, we can get the posterior distribution over functions $p\left(\mathbf{f}^* \mid \mathbf{x}, \mathbf{y}, \mathbf{x}^*\right) \sim \mathcal{N}\left(\mu^*, \Sigma^*\right)$, where

$$
\begin{aligned}
\mu^* &= \mathbf{m}\left(\mathbf{x}^*\right)+K\left(\mathbf{x}^*, \mathbf{x}\right)\left[K(\mathbf{x}, \mathbf{x})+\sigma_n^2 I\right]^{-1}[\mathbf{y}-\mathbf{m}(\mathbf{x})] \\
\Sigma^* &= K\left(\mathbf{x}^*, \mathbf{x}^*\right)-K\left(\mathbf{x}^*, \mathbf{x}\right)\left[K(\mathbf{x}, \mathbf{x})+\sigma_n^2 I\right]^{-1} K\left(\mathbf{x}, \mathbf{x}^*\right)
\end{aligned}.
\tag{4-5}
$$

As shown in Equation 4-5, GP regression is highly related to the kernel function. The mean and the variance of a GP prediction are affected by the selected kernels. Therefore, the choice of the covariance function depends on the prior knowledge and the training data we have. There exist different kinds of kernels that can be applied to GPs, such as standard kernel, linear kernel, periodic kernel, etc. Users can select a suitable kernel according to prior knowledge of the target. Based on different cases, it is also possible to combine different kernels. In fact, any symmetric and positive semi-definite function can be a kernel function. However, some kernels are commonly used in practice. For example, the squared exponential (SE) kernel, also known as the radial basis function (RBF) kernel. Its mathematical definition is as follow:

$$k_{\mathrm{SE}}\left(x, x'\right) = \sigma^2 \exp\left(-\frac{\left(x - x'\right)^2}{2\ell^2}\right) \tag{4-6}$$

where ℓ is the length scale of the kernel which determines the length of the fluctuation in the function, and σ is the output variance. Both ℓ and σ are the hyperparameters of the SE kernel. In practice, we can optimize the hyperparameters of the kernel using gradient-based optimization approaches.

4-1-2 Cloth placing learning framework

The cloth placing setup in our study has two independent robot pickers. Each picker holds on to a corner of the cloth. The proposed framework consists of multiple GP models, trained to take the 3D Cartesian positions of both pickers, the WORD descriptor, and the contact portion as input, and the desired picker displacement along each of the three principal directions as output. Using the 3D Cartesian positions of both pickers as input of GP models allows GPs to know the positions of pickers. It prevents GPs from predicting displacements that cause pickers to stretch the cloth too much. The WORD descriptor and the contact portion provide the cloth state to GPs.

Since a picker can move in 3 axes, our framework consists of 6 GP models to predict the displacement of each picker in x, y, and z directions, Δx, Δy, and Δz respectively.

In this study, the squared exponential kernel was selected within the GP models. The hyperparameters of the kernel are automatically determined through Expectation Maximization with the L-BFGS method using the GPy package [43].

4-2 Simulation experiment

We validate our proposed approaches in an open-source simulation environment with high-fidelity. In this section, the simulation environment and the validation tasks are introduced in detail.

4-2-1 Simulation experiment setup

The simulation experiment setup is the same as subsection 3-2-1.

4-2-2 Cloth placing

This experiment utilizes the proposed learning framework to perform a cloth placing task. In order to learn from demonstrations to place a cloth flat on the surface, we use a Sony PS4 DualShock controller to do demonstrations in SoftGym and record the cloth placing trajectories, WORD, and the contact portion. Then we use the proposed GP framework to train GPs. To compare the performance of the proposed GP framework, we train a baseline GP framework that does not take WORD and contact portion as input. The baseline approach only takes the positions of two pickers as input.

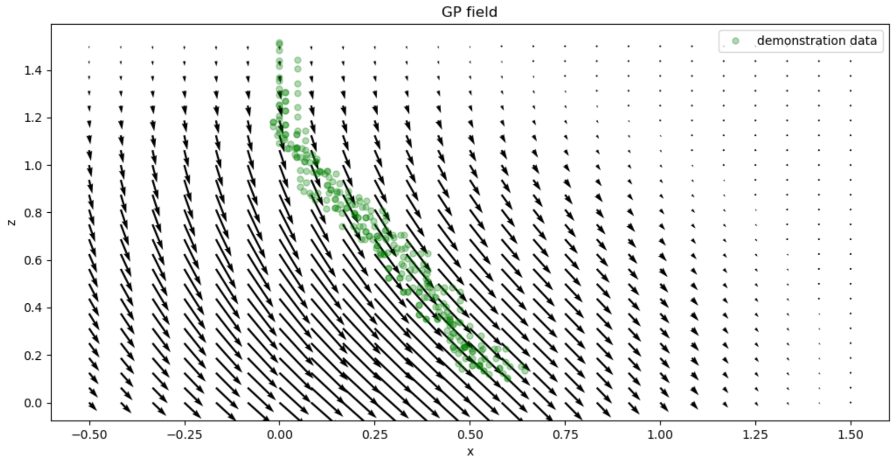


Figure 4-1: A quiver plot of the cloth placing task with multiple demonstrations (baseline GP framework)

Starting from picking a cloth perfectly in the air, the pickers' positions are randomized by giving each picker a random move for 20 steps. Figure 4-1 shows the recorded trajectories of picker 1 in the xz plane (where the y position has a fixed value of 0) and the GP attractor field of the baseline GP framework. The size of the arrow is proportional to the GP output displacement. Five demonstrations are performed to train the baseline GP framework, we start the cloth placing in the upper left corner and end in the lower right corner. As expected, the GP attractor field shows a tendency to attract the picker positions to the lower right corner following a similar curve of demonstrations. Moreover, GPs predict larger displacement near the region of demonstrations. When a picker position is far from the demonstrations region, the GP output displacement decreases.

Figure 4-2 presents the results of cloth placing using the baseline GPs (only takes pickers' positions as input). Figure 4-3 shows the results of using the proposed learning framework. Both method successfully learns to place the cloth on a surface. In the 8^{th} frame of Figure 4-2 and Figure 4-3, the cloth reaches a contact portion above 90%. Finally, in the 10^{th} frame, the contact portions of both cases reach 100%. The experiment results show that our proposed method and the baseline method perform similarly in doing the cloth placing task. Including WORD and contact portion into GP does not have a significant improvement in this case. In the next experiment, we will test the disturbance rejection performances of both methods.

4-2-3 Disturbance rejection

In this section, we carry out an experiment to test the disturbance rejection. We would like to know how well the trained models can react to external disturbance during the cloth placing process. The disturbance rejection performance of the baseline GP model and the proposed learning framework are compared.

The initial condition for the cloth placing task is the same as the cloth placing experiment in subsection 4-2-2. Previous experiment results show when a picker position is far from the demonstrations region, the GP output displacement decreases and its variance becomes larger. Therefore, we set a threshold of the GP output to serve as a stopping criterion. If the

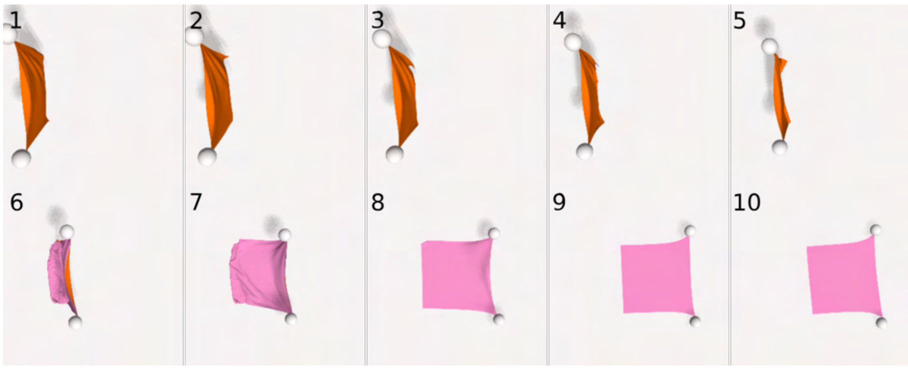


Figure 4-2: Baseline method: Experiment results of cloth placing using position based GPs in SoftGym

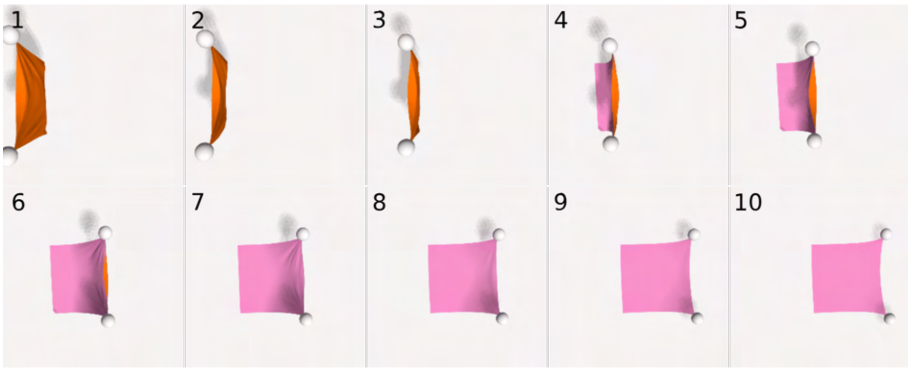


Figure 4-3: Proposed method: Experiment results of cloth placing using proposed learning framework in SoftGym

norm of the GP output $(\Delta x_1, \Delta y_1, \Delta z_1, \Delta x_2, \Delta y_2, \Delta z_2)$ becomes lower than the threshold, the GP stops predicting pickers' displacements and thus the cloth placing movement stops. After the GP stops, we use a mouse to select and drag a part of the cloth in SoftGym to perturb it and create wrinkles.

The experiment results of the baseline method are shown in Figure 4-4. Starting from the 1^{st} frame, the baseline GP models reproduce the placing trajectory until the GP predicted displacement is below the threshold. Eventually, the GP stops in the 5^{th} frame and reaches a 100% contact. In the 6^{th} frame, we manually perturb the cloth to create wrinkles. We can notice that the cloth state and pickers' positions stay the same from 6^{th} to the 10^{th} frame. The results show that the baseline method does not react to the external perturbation since it only takes pickers' positions as input. Figure 4-5a shows the results of the proposed method, which takes WORD and contact portion as additional inputs for GPs. Same as the figure of the baseline method results, the cloth placing task is completed in the 5^{th} frame and the cloth reaches a 100% contact. We perturb the cloth in the 6^{th} frame to create wrinkles. The proposed method reacts to the external disturbance and the GP starts to predict a larger displacement that is greater than the threshold. Therefore, both pickers move towards the right and remove some wrinkles, as shown in 6^{th} to 10^{th} frame of Figure 4-5a. Figure 4-5b shows the zoom-in view of the wrinkle removal processes. We notice that pickers stop moving in the 10^{th} frame when there are still some wrinkles. The reason for

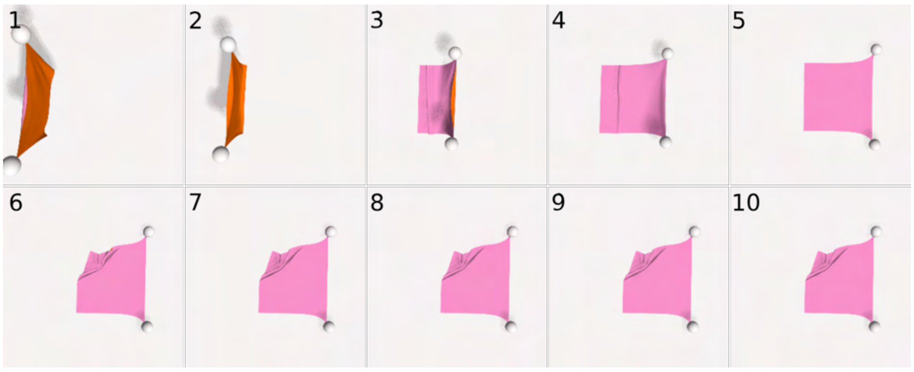


Figure 4-4: Baseline method: Experiment results of cloth placing disturbance rejection in Soft-Gym

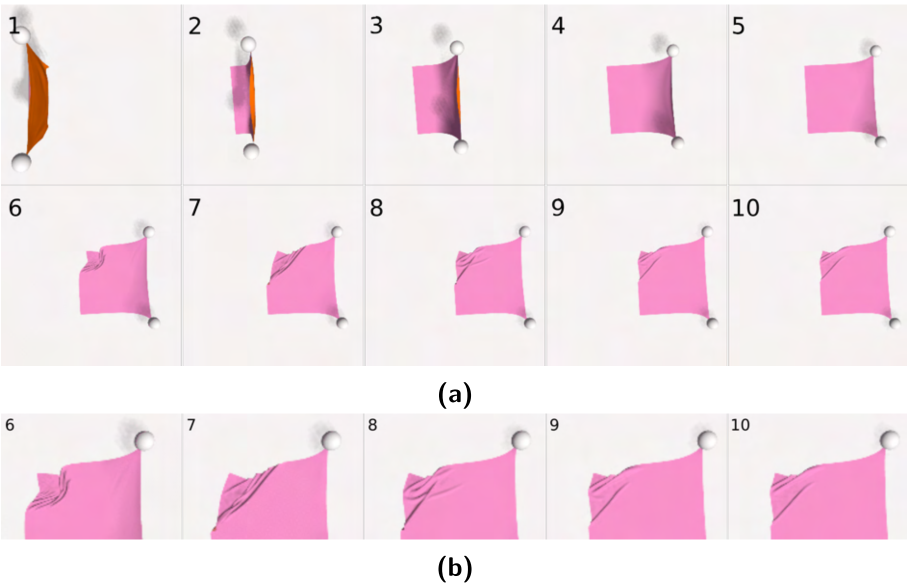


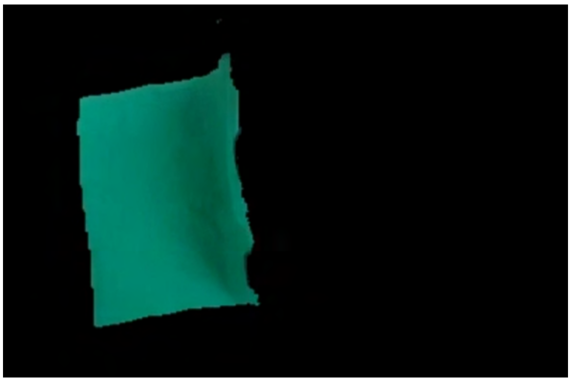
Figure 4-5: Proposed method: Experiment results of cloth placing disturbance rejection in SoftGym

this phenomenon is that the GP output displacement becomes less than the threshold. As a result, the proposed learning framework is unable to completely flatten the cloth after the disturbance. In conclusion, we can notice an obvious difference in disturbance rejection between the baseline method and the proposed method. Including the state of the cloth (WORD and contact portion) helps the trained model know the current cloth configuration. When the cloth state changes, our method reacts to it and removes some wrinkles on the cloth.

4-3 Robot experiment

Besides the validation in a simulation environment, we also perform robot experiments to test the performance of our proposed methods. In this section, we introduce the robot experiment

(a) Original image



(b) Extracted contact part of the cloth

Figure 4-6: Contact portion extraction, the extracted contact portion is 52.28%

in detail.

4-3-1 Robot experiment setup

The robot experiment setup is the same as subsection 3-3-1.

4-3-2 Cloth placing

In order to validate the performance of the proposed learning framework on robot experiments. Kinesthetic teaching is used to do cloth placing demonstrations. During the demonstrations, the robot end-effector positions, WORD, and contact portion are recorded to train GPs.

To get the contact portion of the cloth, we utilize the RGB image and depth map from the RealSense depth camera. We define the contact portion as:

$$C = \frac{P_{contact}}{P_{flat}}, \tag{4-7}$$

where C represents the contact portion, $P_{contact}$ represents the pixel number containing the contact part of the cloth, and P_{flat} is the pixel number containing the whole cloth when it is completely flat on a surface. We apply an HSV thresholding to extract the cloth by its color in the image. By applying a threshold to the depth map, we create a depth mask that only shows the part of the image within a specific depth. Combing both results, it is practical to detect the contact portion in a real robot experiment. Figure 4-6 shows the original image and the extracted contact part of the cloth. In this case, we use the robot to place approximately half of the cloth on the surface and the extracted contact portion is 52.28%. The results show that using this method can get a good estimation of the contact portion. However, due to the limited accuracy, we notice that the RealSense depth camera often includes wrinkles as the contact part. Therefore, WORD is included in our proposed learning framework to capture wrinkles information.

The GP model is simplified since we use a single robotic arm setup for robot experiments. A hanger constrains the distance between two picked corners. Therefore, we eliminate the

	Baseline method	Proposed method
Max final contact portion	99.09%	98.30%
Mean final contact portion	96.62%	96.51%
Min final contact portion	94.48%	91.46%

Table 4-1: Performance in cloth placing

y positions of pickers from the GP input and the displacement in the y axis from the GP output. As a result, the proposed learning framework takes the x and z positions of the robot end-effector, the WORD descriptor, and the contact portion as input. The output of the proposed framework is the desired picker displacement along x and z directions. A baseline model is trained to compare with the proposed learning method. The baseline model does not take the WORD descriptor and the contact portion as input. It only takes the x and z positions of the robot end-effector. The output of the baseline model stays the same as the proposed model. The proposed model and the baseline model are trained with 10 demonstrations. Figure 4-7 and Figure 4-8 show experiment results of the baseline method and the proposed method performing the cloth placing task. Starting from hanging the cloth in the air in the 1^{st} frame, both methods successfully place the cloth on the surface in the 6^{th} frame. Table 4-1 shows the quantitative experiment results of the cloth placing task. To measure the performance of each method, we select the final contact portion of cloth placing as the metric. From Table 4-1, we can notice that the baseline method has a slightly higher contact portion comparing with the proposed method. No significant difference can be seen from the experiment result of each method. We will perform an experiment with external disturbances in the next section.

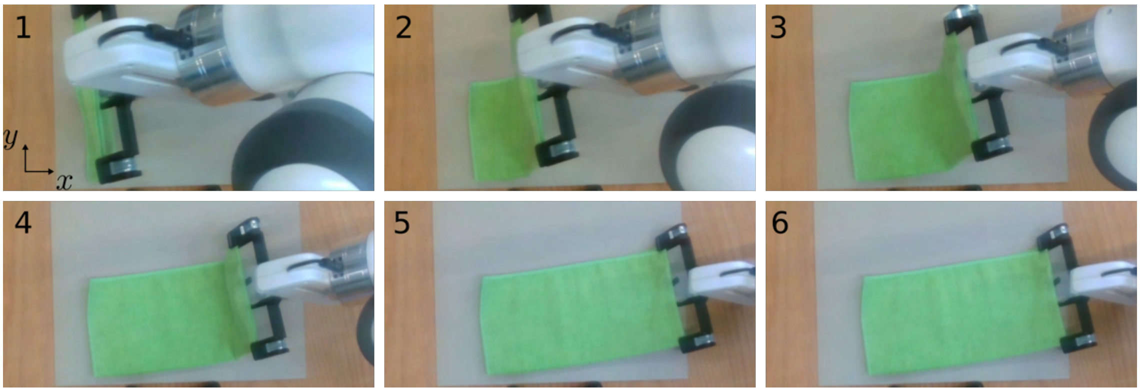
4-3-3 Disturbance rejection

In addition to placing a cloth successfully, the robustness of the trained model is an important characteristic to evaluate. In this section, we present an experiment to test the disturbance rejection performance of the proposed method.

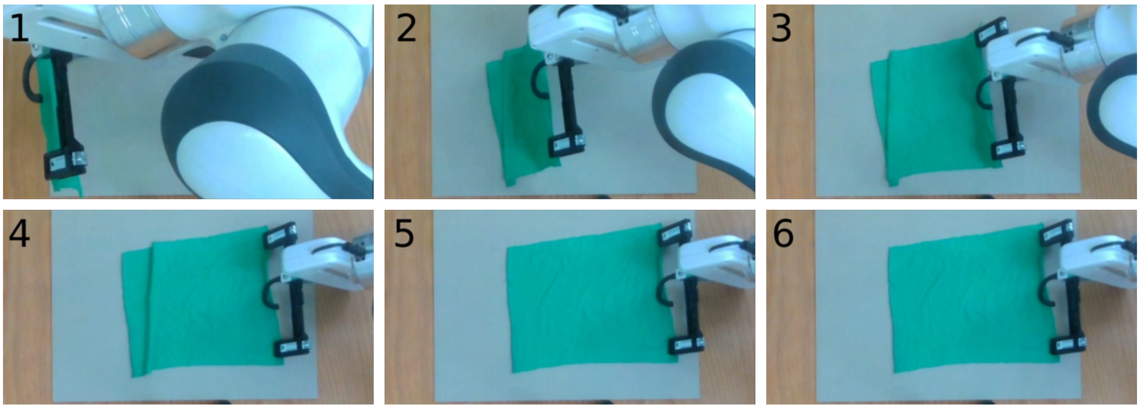
The disturbance rejection experiment has an identical initial condition with the cloth placing task. After the baseline method and the proposed method successfully place the cloth on the surface, we use a stick to disturb the cloth and create wrinkles to see if it reacts to external disturbance.

The disturbance experiment results of the baseline method are shown in Figure 4-9 and the results of the proposed learning method is shown in Figure 4-10. Starting from the 1^{st} frame, the trained model starts cloth placing and complete in the 3^{rd} frame. The cloth contact portion successfully reaches above 90% in each scenario. Since we use an impedance controller as the robot controller, when the GP output displacement is lower than a threshold, the robot stops moving. It is the main reason causing the robot to stop in the 3^{rd} frame for both methods. When the placing movement entirely stops, we use a stick to perturb the cloth and create wrinkles, as shown in the 4^{th} frame in Figure 4-9 and Figure 4-10.

After the external disturbance, from the 5^{th} and the 6^{th} frames of Figure 4-9a and Figure 4-9b we can notice that the baseline method does not remove any wrinkles. In Figure 4-10a and Figure 4-10b, the 5^{th} and the 6^{th} frames show that the proposed learning framework reacts to



(a) Cloth 1

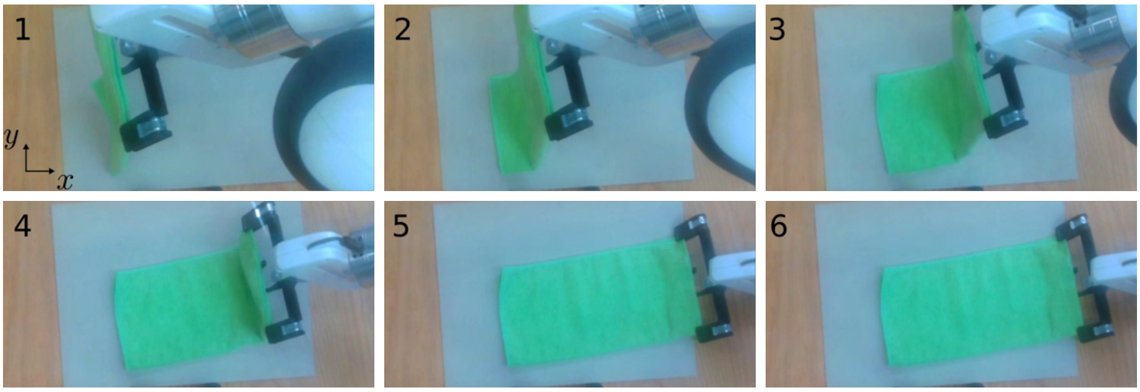


(b) Cloth 2

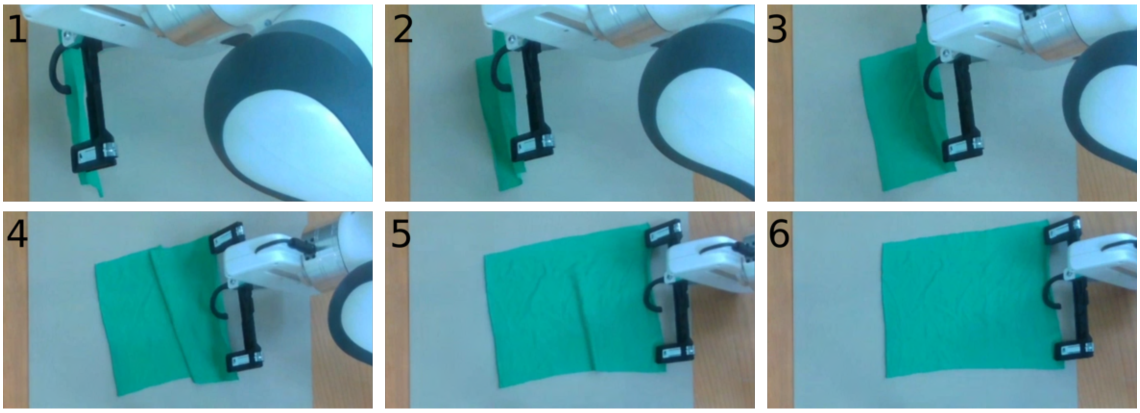
Figure 4-7: Cloth placing experiment results using the baseline method

the disturbance and flattens some wrinkles. In the 7^{th} frame, we disturb the cloth again. The 8^{th} and the 9^{th} show the same behavior. However, not all wrinkles are flattened. A possible reason causing this behavior is that the GP output displacement becomes too small again for the robot to move. Therefore, it is not able to remove all wrinkles.

Table 4-2 shows the quantitative results of the three disturbance rejection experiments. To quantify the performance of each method, we select the contact portion after cloth placing, after the disturbance, and the final contact portion as the metric. From Table 4-2, we observe that the contact portion of the baseline method and the proposed method reach above 95% after cloth placing. It shows again that both methods perform well in cloth placing tasks. After we use a stick to perturb the cloth, we record the contact portion after disturbance and we would like to know if the method reacts to disturbance. Finally, we record the final contact portion. Comparing the contact portion after disturbance and the final contact portion in Table 4-2, the baseline method only has a slight difference which is lower than 1%. It is mainly caused by the small deformation of the cloth after perturbation. The proposed method shows a good performance in flattening the cloth after the disturbance. It is able to increase the contact portion around 5 to 10% after the perturbation. In conclusion, the experiment results show that our proposed learning framework has a significantly better performance in recovering the cloth state than the baseline(position-based) method. Including WORD and contact portion into GPs increases robustness against external disturbance.



(a) Cloth 1

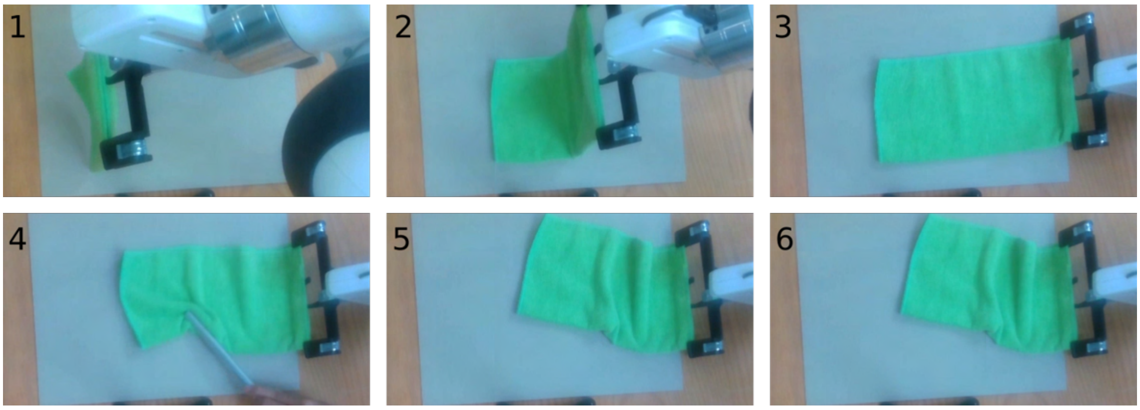


(b) Cloth 2

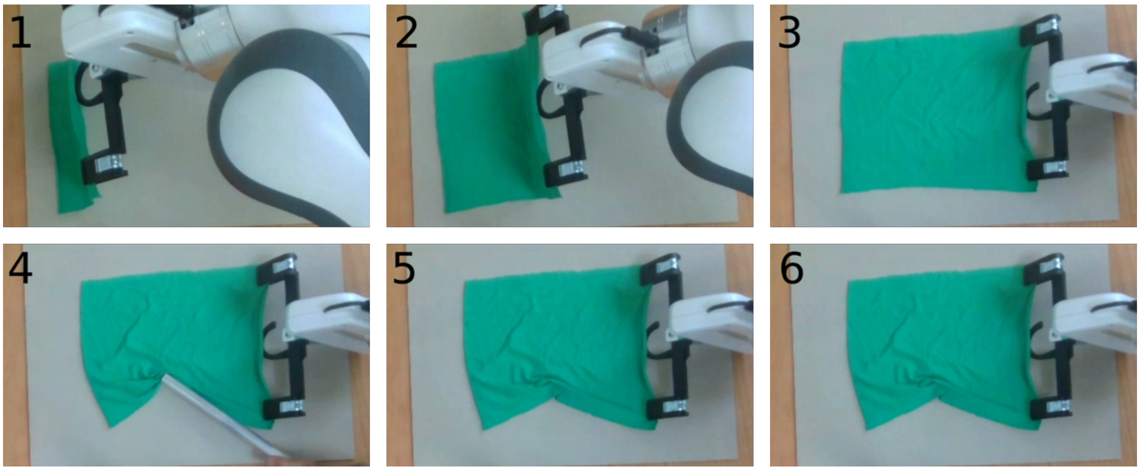
Figure 4-8: Cloth placing experiment results using the proposed method

	Contact portion after cloth placing	Contact portion after disturbance	Final contact portion
Baseline method	95.26%	91.41%	91.25%
	96.33%	89.37%	89.83%
	99.09%	90.04%	90.24%
Proposed method	96.23%	83.90%	92.53%
	98.30%	90.55%	95.32%
	96.60%	78.27%	88.99%

Table 4-2: Performance in disturbance rejection

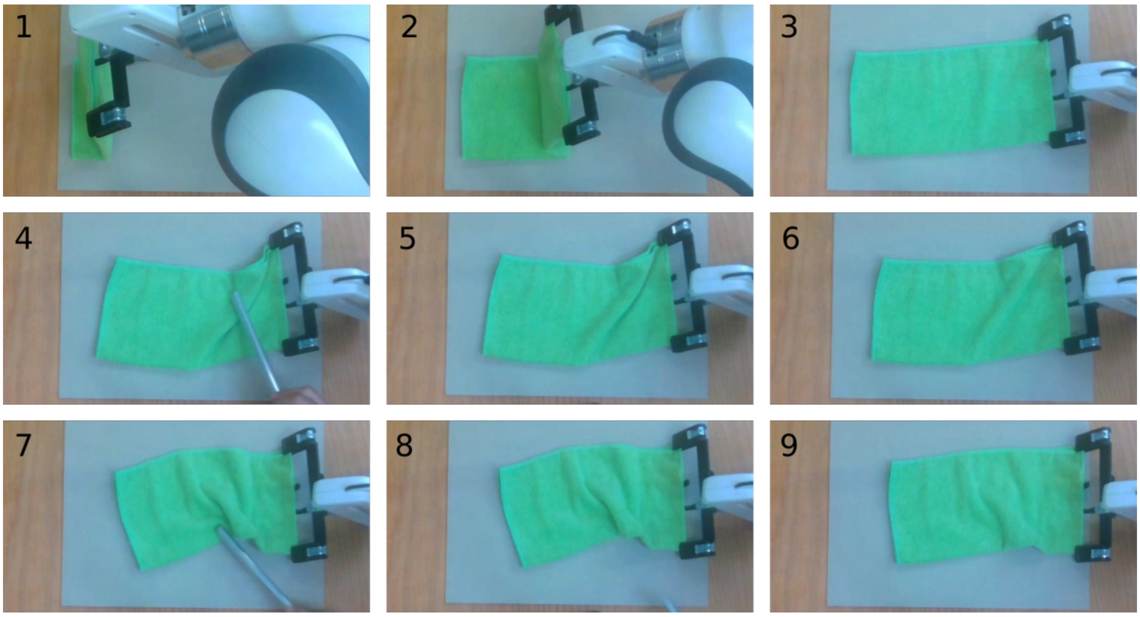


(a) Cloth 1

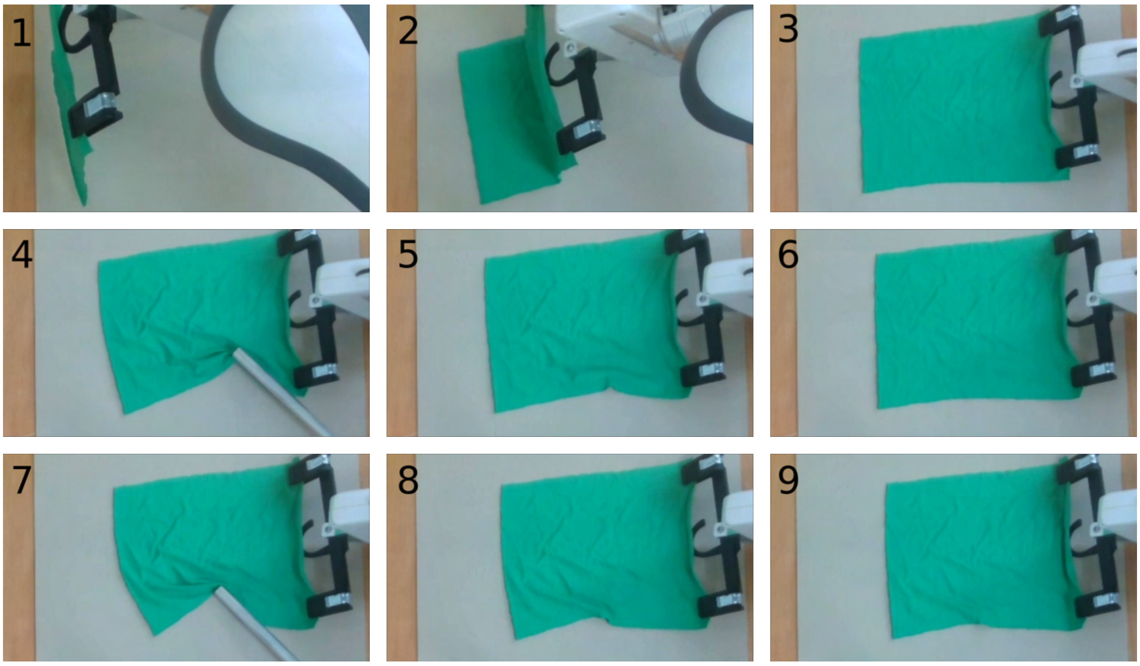


(b) Cloth 2

Figure 4-9: Disturbance rejection experiment results using the baseline method



(a) Cloth 1



(b) Cloth 2

Figure 4-10: Disturbance rejection experiment results using the proposed method

Chapter 5

Discussion

In this chapter, we mention some limitations that we found during the research in section 5-1. The conclusion of this master thesis is presented in section 5-2. Finally, we conclude this thesis by providing possible future research directions in section 5-3.

5-1 Limitations

In this section, we introduce some limitations we found during experiments and provide some possible explanations.

The first limitation we observed is that the proposed learning framework cannot remove all the wrinkles created by external disturbance. In some cases, the robot end-effector has to move toward the $+x$ direction more to remove residual wrinkles. However, due to the decreased GP predicted displacement, the robot is unable to move because of the stiffness of the impedance controller, eventually bringing it to a halt. Although this phenomenon can be alleviated by adjusting the stiffness, it is not a complete solution to this problem.

The second limitation is that our method cannot flatten a cloth when it is not placed properly in the beginning, as shown in Figure 5-1. The 1^{st} frame shows that the reverse side of a small part of the cloth is in contact with the surface then the rest of the cloth is placed above that part in the following frames. It causes the small part of the cloth to be folded beneath other parts of the cloth during the placing process. In the last frame of Figure 5-1, the contact portion stays at 79.39% due to the folded part. Our proposed learning framework only teaches the robot to lower the cloth and place it flat on a surface then drag towards a direction to remove wrinkles. This method is unable to recover from a folded configuration. The main reason that causes the cloth to fold initially is because of improper placing velocity. The improper placing velocity causes the cloth to swing in the air and makes the undesired part of the cloth contact with the surface in the beginning. A possible solution to this problem is to learn appropriate velocities that adapt to different stages of placing. Placing a cloth at a proper speed allows the cloth to be stable during the movement.

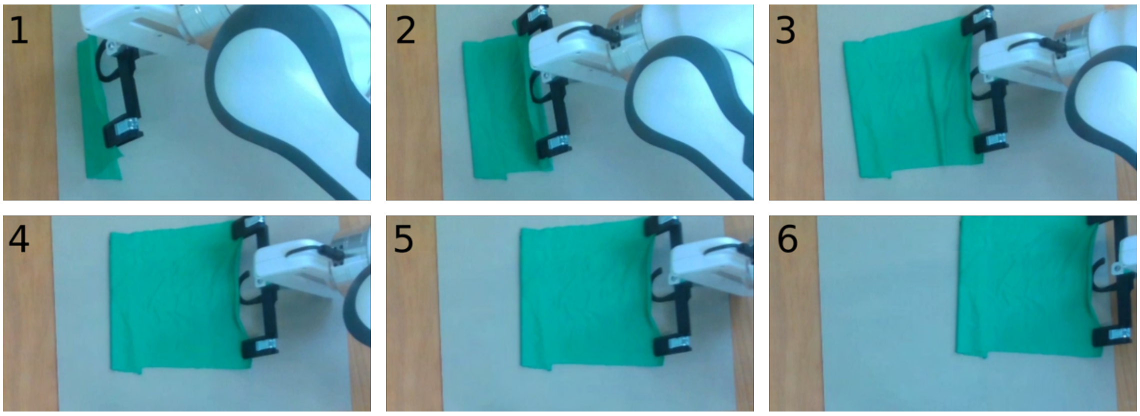


Figure 5-1: Bad initial contact leads to a lower contact portion (79.39%) in the end

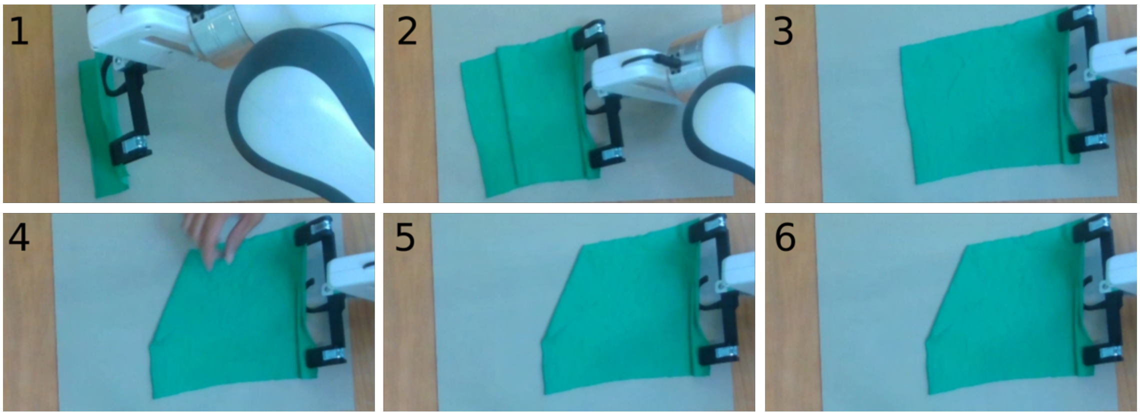


Figure 5-2: The proposed framework cannot flatten the folded corner of the cloth

The third limitation is related to disturbance rejection. In the previous section, we present successful experiment results that the proposed method reacts to the disturbance by a stick and flatten the cloth. In Figure 5-2, the first three frames show a successful cloth placing and the robot stops moving in the 3^{rd} frame. In the 4^{th} frame, we disturb the cloth by folding the upper left corner of it. After making a fold, we notice that the robot moves towards the right and tries to flatten the cloth. Eventually, in the 6^{th} frame the robot stops due to the decreased GP output. To solve this problem, a possible solution is to bring the cloth back in the air and repeat the placing procedure.

5-2 Conclusions

This thesis work presents a practical approach for learning cloth placing and flattening tasks. We present a novel low-dimensional cloth state representation, WORD, and a LfD framework to place a cloth flat on a surface from human demonstration. We show that WORD can efficiently capture the wrinkles on a cloth and compute a stretching direction inspired by human behavior. Validation experiments in SoftGym and on real robots show that WORD performs well in removing wrinkles of crumpled cloth. Thus, we show that WORD is applicable in a real-world scenario. In addition, the proposed learning framework successfully generates a sequence of motions to place a cloth starting from a random configuration in the air. Another contribution of the thesis is showing that including WORD and contact portion into the LfD frameworks improves the robustness against perturbation. With the validation in simulation and real robot experiments, we provide a novel visual descriptor for cloth-like objects and a new learning method for cloth placing tasks that utilizes WORD.

5-3 Future works

In the future, we plan to use a dual-arm robot to validate the WORD descriptor and the cloth flattening framework. Besides, certain important aspects remain to be addressed for the better generalization and robustness of the proposed framework. For example, testing if WORD can perform well under different illumination conditions or performing more experiments with cloths that have different characteristics. Moreover, comparing GP with other motion encoding methods such as DMP or GMM is also a possible future work for this thesis research. In this thesis, we use WORD and contact portion to represent the cloth state specifically for the cloth placing task. This low-dimensional representation might be applicable for other cloth manipulation tasks. Combining with other cloth representation methods could potentially create a more generalized and robust framework that does not only encode cloth placing tasks.

Appendix A

Workshop paper

The paper was accepted by IROS 2021 – the Second Workshop on Robotic Manipulation of Deformable Objects: Challenges in Perception, Planning and Control for Real-World Applications (RoMaDO-RA), and presented at the workshop.

Wrinkle contraction direction: a useful feature for learning robotic fabric manipulation from demonstration

Chia-Yu Tsai, Jihong Zhu, and Jens Kober

Abstract— **Deformable objects manipulation (DOM) is largely considered an open problem in robotics. The complexity stems from the high degrees of freedom and the nonlinear nature of the object configurations. In this paper, we consider placing and flattening tasks for a piece of fabric. We propose a practical framework to place a cloth on a surface based on visual perception and human demonstrations. We present a novel feature, Wrinkle cOntRaction Direction (WORD), which extracts a stretching direction to flatten clothes from image data. Furthermore, we integrate WORD and demonstrations into Gaussian Processes to learn a cloth placing policy. Simulation results are used to validate the performance of WORD and the proposed learning framework.**

I. Introduction

In this paper, we investigate placing and flattening tasks of square cloth-like objects. Given a large enough flat surface, and a (possible multi-arm) robot grasp on a convex vertex of the object hanging in the air [1], we look for a framework to generate sequential motions for the robot to place the cloth on the surface as flat as possible.

By extracting useful features from images, we can infer the cloth configuration efficiently. After knowing the state of the garment, a control method is essential to manipulate the cloth based on the specific goal. Instead of handcrafted design, we propose to obtain such a strategy from human demonstrations. Humans possess the intuition to infer the complex configuration of a deformable object and perform a suitable manipulation strategy. Therefore, learning from demonstration (LfD) is especially handy for deformable object manipulation as it tries to model this complex pattern from data.

The main contribution of this paper is proposing a novel feature, Wrinkle cOntRaction Direction (WORD), which directly maps the wrinkles of the cloth-like objects to an action direction. We integrate WORD into a Gaussian Process framework that yields a robust cloth flattening strategy from human demonstrations.

II. Related work

In this section, we present related works in two categories: cloth state representation and cloth placing and stretching tasks.

A. Cloth state representation

Scale-invariant feature transform (SIFT) features are suitable for cloth classification since they can efficiently capture the distinctive features under different conditions [2] [3]. Histogram of oriented gradients (HOG) features capture the edge of objects but clothes are highly deformable and the shapes are variant [4]. Research shows that after bringing a garment into the desired configuration, combining HOG features with machine learning algorithms can successfully recognize the garment type [5]. Histogram of oriented wrinkles (HOW) features mainly capture the deformation and wrinkles of objects [6] [7]. Based on the HOW features, robots are able to control the deformation of clothes by minimizing the error between the HOW features of the current configuration and the desired configuration. These descriptors successfully compute the low-dimensional features of cloth-like objects from RGB-(D) data, otherwise it is infeasible to represent the infinite degrees of freedom of a garment. In addition to these low-dimensional cloth state representations, there exist other approaches to represent cloth configuration in more detail, such as the deformable model based approach. In this study, we focus on low-dimensional cloth state representations.

B. Cloth placing and stretching task

Endowing robots the ability to place a cloth on a surface properly provides a nice initial state for subsequent tasks, such as cloth folding. Jangir et al. [8] used a deep reinforcement learning approach to solve dynamic cloth placing tasks. With few demonstrations, their approach successfully learned to do diagonal folding, sideways folding, and placing in a simulation environment. However, the proposed approach requires getting the positions and velocities of certain points on the edge of the cloth. Although these states are possible to extract through vision, self-occlusion usually happens in cloth manipulation. Therefore, this approach is not practical to use in a real-world situation.

Balaguer et al. [9] proposed a new learning algorithm that combines imitation and reinforcement learning to perform a towel folding task, which places half of the towel on a surface then folds it in half. A few human demonstrations are captured with the 28 tracker around the edges of the cloth. With an imitation learning algorithm, an initial folding policy is learned. Next, a reinforcement learning algorithm improves the placing policy. Due to imitation learning providing a good starting policy, the reinforcement learning algorithm converges extremely fast. Although the research presented a successful result, there are trackers around the edge of the towel and it requires a motion capture system to acquire data which is not a practical solution for cloth placing and folding tasks in the real world.

All authors are with Cognitive Robotics at the Faculty of 3mE, Delft University of Technology, The Netherlands. `C.Tsai-3@student.tudelft.nl`

III. Proposed framework

The proposed framework in this study is inspired by human behavior. When a person places a cloth flat on a surface, he/she keeps track of the state of the cloth, whether it contacts with the surface or there are any wrinkles. The final goal is to make all the areas in contact with the surface without any wrinkles. Therefore, learning from demonstration is the main method we select to perform the task. Moreover, WORD efficiently provides the cloth state and it is a practical method to use in a real-world scenario. In this section, we present details of WORD and the proposed learning framework.

A. Novel descriptor: WORD

WORD is a 2-dimensional vector indicating a proper stretching direction to remove wrinkles. Compared with HOW features, it significantly reduces the dimension to describe the cloth state. In our problem, the goal of cloth placing is to minimize the wrinkles and maximize the contact area, WORD can provide sufficient information of the cloth state for this specific task.

In order to detect wrinkles on the contact part of the cloth, we use a depth camera to capture the images and depth map. With the depth map, the contact part of the cloth can be detected. As a result, the wrinkles on the contact area can be extracted with HOW features, which are useful to describe the shape variation of highly deformable objects. HOW features are computed by applying 2D Gabor filters [10] on RGB images with multiple orientations and wavelengths then extracting low-frequency and high-frequency components. Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave oriented at an angle. A Gabor kernel can be represented as:

$$
\begin{aligned}
&g(x, y; \lambda, \phi, \sigma, \gamma, \theta) \\
&\quad = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \sin\left(2\pi \frac{x'}{\lambda} + \phi\right), \quad (1)
\end{aligned}
$$

where $x' = x\cos\theta + y\sin\theta, y' = -x\sin\theta + y\cos\theta$, λ represents the wavelength, ψ is the phase offset, σ is the standard deviation of the Gaussian envelope, γ is the spatial aspect ratio, and θ is the orientation of the normal to the parallel stripes of a Gabor function. To compute WORD, the first step is extracting wrinkles in multiple orientations and get the number of pixels containing wrinkles in each direction. Next, we compute a weight for each orientation based on the pixel number of wrinkles. Finally, the summation of weight times the corresponding direction gives the direction of WORD. The algorithm for computing WORD is shown in Algorithm 1. In conclusion, WORD utilizes Gabor filters and a depth map to capture wrinkles on the contact part of a cloth and indicates a stretching direction to remove wrinkles. An RGB-D camera can get essential information for WORD. Therefore, WORD is a practical visual descriptor to use in a real-world scenario without expensive equipment.

Algorithm 1 Computing WORD

Input: image $\boldsymbol{I}$ of size $(w_{\boldsymbol{I}},\ h_{\boldsymbol{I}})$, cloth mask $\boldsymbol{C}$ and depth mask $\boldsymbol{D}$ of size $(w_{\boldsymbol{I}}, h_{\boldsymbol{I}})$, number of orientations n.
Output: WORD vector $\boldsymbol{W}$
for $i = 0, 1, \ldots, n$ **do**
  $\theta_n = i \times \pi/n$
  Get Gabor kernel in direction θ_n
  Apply the Gabor kernel on the image $\boldsymbol{I}$ to get wrinkles in direction θ_n
  Apply the cloth mask and depth mask on the output image to get wrinkles on contact part of the cloth
  Stack the summation of pixel values in an n-dimensional vector $\boldsymbol{L}$
end for
$\boldsymbol{w} = (\boldsymbol{L}/\max(\boldsymbol{L}))$ ▷ $\boldsymbol{w}$: weight vector in each direction
$\boldsymbol{d} = \sum_{i=1}^{n} w_i \boldsymbol{v}_i(\theta_n)$ ▷ $\boldsymbol{v}$: unit vector perpendicular to θ_n direction
$\boldsymbol{W} = \boldsymbol{d}/\mathrm{norm}(\boldsymbol{d})$
return $\boldsymbol{W}$

B. Learning from demonstration using Gaussian processes

A Gaussian process (GP) is a probability distribution over possible functions that fit a set of points [11]. It is defined by its mean function and covariance function. The mean function $m(\mathbf{x})$ and the covariance function (i.e. kernel function) $k(\mathbf{x}, \mathbf{x}')$ of a real process $f(\mathbf{x})$ are defined as:

$$
\begin{aligned}
m(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})] \\
k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}\left[(f(\mathbf{x}) - m(\mathbf{x}))\,(f(\mathbf{x}') - m(\mathbf{x}'))\right].
\end{aligned} \quad (2)
$$

The kernel function $k(\mathbf{x}, \mathbf{x}')$ is a measure of similarity between two input vector $\mathbf{x}$ and $\mathbf{x}'$. A GP regression is denoted as:

$$
f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right). \quad (3)
$$

A Gaussian processes regression is selected to learn from demonstration data since GP predictions interpolate the observations and provide the confidence level of each prediction point. The simulation setup in our study has two independent robot pickers. Each picker holds on to a corner of the cloth. The proposed framework consists of multiple GP models, trained to take the 3D Cartesian positions of both pickers, the placing direction, and the contact portion as input, and the desired picker displacement along each of the three principal directions as output. Therefore, our framework consists of 6 GP models to predict the displacement of each picker in x, y, and z directions, Δx, Δy, and Δz respectively. In this study, the squared exponential kernel was selected within the GP models. The hyper-parameters of the kernel are automatically determined through Expectation Maximization with the L-BFGS method using the GPy package [12].

IV. Simulation results

In order to validate our proposed framework, we utilize a simulation environment, SoftGym [13], to test WORD and

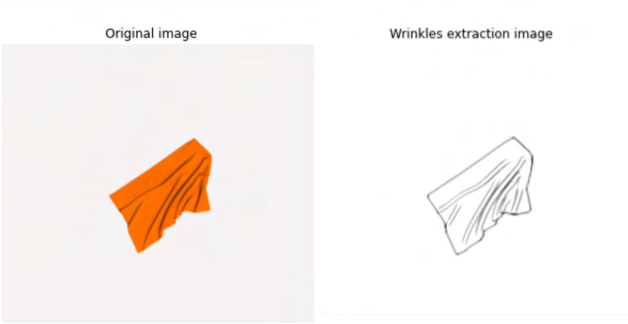


Fig. 1. Original images from SoftGym and extracted wrinkles in 8 directions with parameters: $\sigma = 4,\ \lambda = 8,\ \gamma = 0.1,\ \psi = 10$, kernel size $= 12$

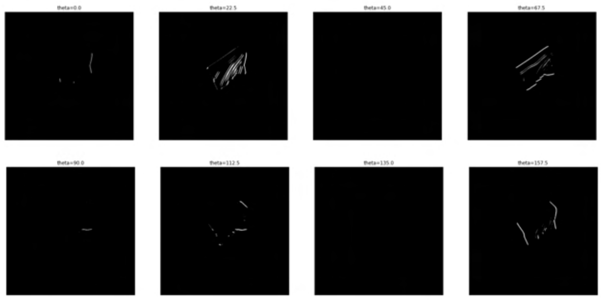


Fig. 2. Detected wrinkles in eight direction, the WORD direction = [0.80 -0.60]

its application on cloth placing tasks. The validation consists of two steps. The first step is to validate the performance of WORD and see if it indicates proper stretching directions to remove wrinkles. The second step is integrating WORD into GPs to perform a cloth placing task.

A. Validate the performance of WORD

We test the performance of WORD in SoftGym, where a crumpled cloth is placed on a surface. Figure 1 shows the original image from SoftGym and the captured wrinkles. Stacking the output images applying 8 Gabor filters in different orientations successfully capture most of the wrinkles on the cloth. The output images in each direction are shown in Figure 2. In this case, our method successfully gets the cloth stretching direction that is perpendicular to the majority of wrinkles based on WORD.

Based on WORD, we design an experiment to show its performance. Figure 3 shows the experiment result. Initially, two independent pickers hold two consecutive corners of the cloth. The cloth is crumpled and placed on the ground. The goal is to use WORD to pull the cloth to the right and flatten it. In the 1^{st} frame of Figure 3, there are several wrinkles and the contact portion is 74.34%. The controller detects the stretching direction and moves the corresponding picker in that direction. After a few movements, it reduces the wrinkles and the contact portion reaches 100.00% in the 10^{th} frame.

B. Cloth placing

The next step is to utilize the proposed learning framework to perform a cloth placing task. An environment with an infinitely large surface is created in the SoftGym. Two white balls represent two robotic grippers picking on two consecutive corners of the cloth. Initially, both grippers pick the cloth properly and hang it in the air.

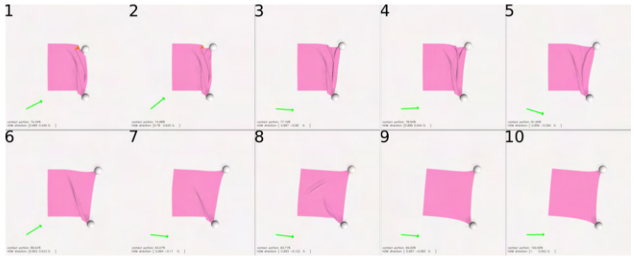


Fig. 3. Wrinkle removal based on WORD (the green arrow indicate the WORD direction for each step)

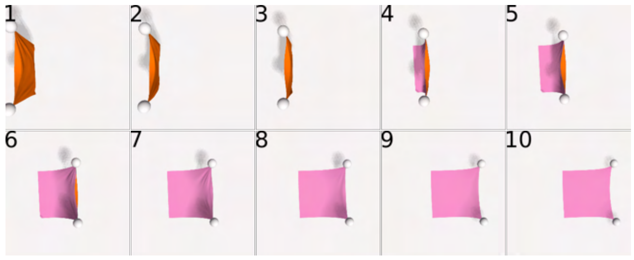


Fig. 4. Experiment results of GP cloth placing

In order to learn from demonstrations to place a cloth flat on the surface, we use a PS4 controller to do demonstrations in SoftGym and record the cloth placing trajectories, WORD, and contact portion. Then we use the proposed GP framework to train the placing policy. Starting from picking a cloth perfectly in the air, the state is randomized by giving each picker a random move for 20 steps. Simulation results in Figure 4 show that the proposed method successfully learns to place the cloth on a surface.

V. Conclusions

In this paper, we present a novel low-dimensional cloth state representation, WORD, and a method to place a cloth flat on a surface from human demonstration. We show that WORD can efficiently capture the wrinkles on a cloth and compute a stretching direction inspired by human behavior. Validation experiments in SoftGym show that WORD performs well in removing wrinkles of crumpled cloth. In addition, the proposed learning framework successfully generates a sequence of motions to place a cloth starting from a random configuration in the air. In the future we plan to add disturbance in simulation to show the robustness of the proposed framework, also to validate it on a real robot setup.

Acknowledgement

This work was supported by the Honda Research Institute Europe GmbH as part of the project: Learning Physical Human-Robot Cooperation Tasks.

REFERENCES

[1] J. Van Den Berg, S. Miller, K. Goldberg, and P. Abbeel, "Gravity-based robotic cloth folding," in *Algorithmic Foundations of Robotics IX*. Springer, 2010, pp. 409–424.

[2] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. Ieee, 1999, pp. 1150–1157.

[3] ——, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[4] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 886–893.

[5] C. Kampouris, I. Mariolis, G. Peleka, E. Skartados, A. Kargakos, D. Triantafyllou, and S. Malassiotis, "Multi-sensorial and explorative recognition of garments and their material properties in unconstrained environment," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1656–1663.

[6] B. Jia, Z. Hu, J. Pan, and D. Manocha, "Manipulating highly deformable materials using a visual feedback dictionary," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 239–246.

[7] B. Jia, Z. Pan, Z. Hu, J. Pan, and D. Manocha, "Cloth manipulation using random-forest-based imitation learning," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2086–2093, 2019.

[8] R. Jangir, G. Alenyà, and C. Torras, "Dynamic cloth manipulation with deep reinforcement learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4630–4636.

[9] B. Balaguer and S. Carpin, "Combining imitation and reinforcement learning to fold deformable planar objects," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 1405–1412.

[10] T. S. Lee, "Image representation using 2d gabor wavelets," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 18, no. 10, pp. 959–971, 1996.

[11] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer school on machine learning*. Springer, 2003, pp. 63–71.

[12] GPy, "GPy: A gaussian process framework in python," http://github.com/SheffieldML/GPy, since 2012.

[13] X. Lin, Y. Wang, J. Olkin, and D. Held, "Softgym: Benchmarking deep reinforcement learning for deformable object manipulation," *arXiv preprint arXiv:2011.07215*, 2020.

Bibliography

[1] D. Sullivan, "6 clothes-folding hacks that save closet and drawer space."

[2] "Fruit and veg growers call for workers to 'feed the nation'," Mar 2020.

[3] M. Cusumano-Towner, A. Singh, S. Miller, J. F. O'Brien, and P. Abbeel, "Bringing clothing into desired configurations with limited perception," in *2011 IEEE International Conference on Robotics and Automation*, pp. 3893–3900, IEEE, 2011.

[4] L. Sun, G. Aragon-Camarasa, S. Rogers, and J. P. Siebert, "Accurate garment surface analysis using an active stereo robot head with application to dual-arm flattening," in *2015 IEEE international conference on robotics and automation (ICRA)*, pp. 185–192, IEEE, 2015.

[5] B. Balaguer and S. Carpin, "Combining imitation and reinforcement learning to fold deformable planar objects," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1405–1412, IEEE, 2011.

[6] J. Stria, D. Prša, V. Hlaváč, L. Wagner, V. Petrík, P. Krsek, and V. Smutnỳ, "Garment perception and its folding using a dual-arm robot," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 61–67, IEEE, 2014.

[7] B. Willimon, I. Walker, and S. Birchfield, "A new approach to clothing classification using mid-level layers," in *2013 IEEE International Conference on Robotics and Automation*, pp. 4271–4278, IEEE, 2013.

[8] Y. Kita, T. Ueshiba, E. S. Neo, and N. Kita, "Clothes state recognition using 3d observed data," in *2009 IEEE International Conference on Robotics and Automation*, pp. 1220–1225, IEEE, 2009.

[9] Y. Kita, F. Kanehiro, T. Ueshiba, and N. Kita, "Clothes handling based on recognition by strategic observation," in *2011 11th IEEE-RAS International Conference on Humanoid Robots*, pp. 53–58, IEEE, 2011.

[10] Y. Li, Y. Wang, M. Case, S.-F. Chang, and P. K. Allen, "Real-time pose estimation of deformable objects using a volumetric approach," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1046–1052, IEEE, 2014.

[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 *2010 IEEE International Conference on Robotics and Automation*, pp. 2308–2315, IEEE, 2010.

[12] A. Ramisa, G. Alenya, F. Moreno-Noguer, and C. Torras, "Using depth and appearance features for informed robot grasping of highly wrinkled clothes," in *2012 IEEE International Conference on Robotics and Automation*, pp. 1703–1708, IEEE, 2012.

[13] K. Yamazaki, "Grasping point selection on an item of crumpled clothing based on relational shape description," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3123–3128, IEEE, 2014.

[14] C. Bersch, B. Pitzer, and S. Kammel, "Bimanual robotic cloth manipulation for laundry folding," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1413–1419, IEEE, 2011.

[15] A. Doumanoglou, A. Kargakos, T.-K. Kim, and S. Malassiotis, "Autonomous active recognition and unfolding of clothes using random decision forests and probabilistic planning," in *2014 IEEE international conference on robotics and automation (ICRA)*, pp. 987–993, IEEE, 2014.

[16] Y. Li, D. Xu, Y. Yue, Y. Wang, S.-F. Chang, E. Grinspun, and P. K. Allen, "Regrasping and unfolding of garments using predictive thin shell modeling," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1382–1388, IEEE, 2015.

[17] R. Jangir, G. Alenyà, and C. Torras, "Dynamic cloth manipulation with deep reinforcement learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4630–4636, IEEE, 2020.

[18] D. Seita, A. Ganapathi, R. Hoque, M. Hwang, E. Cen, A. K. Tanwani, A. Balakrishna, B. Thananjeyan, J. Ichnowski, N. Jamali, *et al.*, "Deep imitation learning of sequential fabric smoothing from an algorithmic supervisor," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9651–9658, IEEE, 2020.

[19] Y. Li, X. Hu, D. Xu, Y. Yue, E. Grinspun, and P. K. Allen, "Multi-sensor surface analysis for robotic ironing," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5670–5676, IEEE, 2016.

[20] S. Miller, J. Van Den Berg, M. Fritz, T. Darrell, K. Goldberg, and P. Abbeel, "A geometric approach to robotic laundry folding," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 249–267, 2012.

[21] Y. Li, Y. Yue, D. Xu, E. Grinspun, and P. K. Allen, "Folding deformable objects using predictive simulation and trajectory optimization," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6000–6006, IEEE, 2015.

[22] T. Tamei, T. Matsubara, A. Rai, and T. Shibata, “Reinforcement learning of clothing assistance with a dual-arm robot,” in *2011 11th IEEE-RAS International Conference on Humanoid Robots*, pp. 733–738, IEEE, 2011.

[23] A. Kapusta, W. Yu, T. Bhattacharjee, C. K. Liu, G. Turk, and C. C. Kemp, “Data-driven haptic perception for robot-assisted dressing,” in *2016 25th IEEE international symposium on robot and human interactive communication (RO-MAN)*, pp. 451–458, IEEE, 2016.

[24] W. Yu, A. Kapusta, J. Tan, C. C. Kemp, G. Turk, and C. K. Liu, “Haptic simulation for robot-assisted dressing,” in *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 6044–6051, IEEE, 2017.

[25] Z. Erickson, H. M. Clever, G. Turk, C. K. Liu, and C. C. Kemp, “Deep haptic model predictive control for robot-assisted dressing,” in *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 1–8, IEEE, 2018.

[26] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2, pp. 1150–1157, Ieee, 1999.

[27] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[28] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, vol. 1, pp. 886–893, IEEE, 2005.

[29] C. Kampouris, I. Mariolis, G. Peleka, E. Skartados, A. Kargakos, D. Triantafyllou, and S. Malassiotis, “Multi-sensorial and explorative recognition of garments and their material properties in unconstrained environment,” in *2016 IEEE international conference on robotics and automation (ICRA)*, pp. 1656–1663, IEEE, 2016.

[30] B. Jia, Z. Hu, J. Pan, and D. Manocha, “Manipulating highly deformable materials using a visual feedback dictionary,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 239–246, IEEE, 2018.

[31] B. Jia, Z. Pan, Z. Hu, J. Pan, and D. Manocha, “Cloth manipulation using random-forest-based imitation learning,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2086–2093, 2019.

[32] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, “Learning and generalization of motor skills by learning from demonstration,” in *2009 IEEE International Conference on Robotics and Automation*, pp. 763–768, IEEE, 2009.

[33] S. Calinon, “A tutorial on task-parameterized movement learning and retrieval,” *Intelligent service robotics*, vol. 9, no. 1, pp. 1–29, 2016.

[34] M. Schneider and W. Ertel, “Robot learning by demonstration with local gaussian process regression,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 255–260, IEEE, 2010.

[35] J. Kober, B. Mohler, and J. Peters, "Learning perceptual coupling for motor primitives," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 834–839, IEEE, 2008.

[36] Z. Ghahramani and M. I. Jordan, "Supervised learning from incomplete data via an em approach," in *Advances in neural information processing systems*, pp. 120–127, 1994.

[37] Y. Huang, L. Rozo, J. Silvério, and D. G. Caldwell, "Kernelized movement primitives," *The International Journal of Robotics Research*, vol. 38, no. 7, pp. 833–852, 2019.

[38] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer school on machine learning*, pp. 63–71, Springer, 2003.

[39] J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, and M. West, "Generative or discriminative? getting the best of both worlds," *Bayesian statistics*, vol. 8, no. 3, pp. 3–24, 2007.

[40] M. Arduengo, A. Colomé, J. Lobo-Prat, L. Sentis, and C. Torras, "Gaussian-process-based robot learning from demonstration," *arXiv preprint arXiv:2002.09979*, 2020.

[41] T. S. Lee, "Image representation using 2d gabor wavelets," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 18, no. 10, pp. 959–971, 1996.

[42] X. Lin, Y. Wang, J. Olkin, and D. Held, "Softgym: Benchmarking deep reinforcement learning for deformable object manipulation," *arXiv preprint arXiv:2011.07215*, 2020.

[43] GPy, "GPy: A gaussian process framework in python." `http://github.com/SheffieldML/GPy`, since 2012.