A Data-Centric Approach For Dual-Arm Robotic Garment Flattening

Li Duan and Gerardo Aragon-Camarasa

Abstract—Due to the high dimensionality of object states, a garment flattening pipeline requires recognising the configurations of garments for a robot to produce/select manipulation plans to flatten garments. In this paper, we propose a data-centric approach to identify known configurations of garments based on a known configuration network (KCNet) trained on depth images that capture the known configurations of garments and prior knowledge of garment shapes. The known configurations of garments are the configurations of garments when a robot hangs garments in the middle of the air. We found that it is possible to achieve 92% accuracy if we let the robot recognise the common hanging configurations (the known configurations) of garments. We also demonstrate an effective robot garment flattening pipeline with our proposed approach on a dual-arm Baxter robot. The robot achieved an average operating time of 221.6 seconds and successfully manipulated garments of five different shapes.

Index Terms—Computer Vision for Automation, Deep Learning for Visual Perception, Visual Learning, AI-Enabled Robotics

I. INTRODUCTION

ROBOTIC deformable-object manipulation remains a significant challenge in robotic research. Deformable object manipulation covers many areas of daily life: robot-assisted garment wearing for disability [1], robotic laundry [2], automated manufacturing, and robotic cooking [3]. Mastering deformable object manipulation for robots is key to a more accessible future for the disability community and advanced manufacturing.

Deformable objects have two characteristics: the dimensionality of their object states (configurations) is high, changes in their object states (configurations) are instant, and irregular deformation patterns when a robot manipulates deformable objects. When deformable objects such as garments are flattened, folded, or sponges are gripped, their deformation patterns are irregular, making it challenging to find manipulating points. Researchers traditionally consider two ways to handle the challenge: model-centric and data-centric approaches. Model-centric approaches [4], [5], [6] focus on finding and defining specific models for objects, and manipulation plans are derived and updated by monitoring changes in configurations of those models. Data-centric [7], [8], [9] approaches are divided into two categories. First, deformation patterns are learned from large-scale datasets, which are used for identifying grasping/manipulating points on garments. Those grasping/manipulating points are then used for flattening or operating other manipulations on garments. Second, robots are trained to learn skills for manipulating objects via reinforcement and imitation learning.

There are several problems with model-centric approaches. Firstly, defining or finding black-box models that represent deformable objects is challenging. Deformable objects tend to have an infinite number of states, which means single or multiple black-box models cannot represent all possible object states. Model-centric approaches are usually validated in objects with simple deformations such as sponges [10], towels [11] or fruits [4]. Complex, deformable objects such as garments are rarely employed because black-box models for representing object states would require an almost infinite database representing all possible object states of those deformable objects. Secondly, models cannot be generalised, which means changing objects means updating models. Different objects have different characteristics (e.g. materials and textures), which means that black-box models are specific and object-restricted. A new model is needed for a new object, while knowledge or parameters may be invalidated, causing robotic manipulation tasks constrained to specific objects.

Similarly, limitations exist for data-centric approaches. Robots are usually trained in simulated environments for reinforcement and imitation learning approaches due to limitations in real environments. Robots may be rewarded with actions that are not feasible in real environments due to the range of constraints of orientations and positions for robotic grippers [12]. Robots can take 100-200 epochs to be trained, which would take a significant time if trained in real environments [13]. While reward policies trained in simulated environments can potentially be invalid in real environments due to the differences in external factors such as illumination conditions and the imperfections of simulated objects [14].

In this paper, we propose a data-centric approach that aims to find known configurations of garments and a robot that flattens garments with pre-designed manipulation plans. Known configurations of garments are the configurations of hanging garments after a robot has grasped them from a given, random grasping point. A known configuration network (KCNet) has been trained in this paper to recognise the known configurations of garments with five different shapes. Compared with model-centric approaches, our approach does not require defining specific models for garments. We do not train a robot to learn manipulation skills with reinforcement or imitation learning. Instead, we propose that a robot select pre-designed manipulation plans based on the recognised known configurations of garments. Traditional approaches such as
A. Model-centric approaches

Each model has four components: a landmark generator, a contour generator, a legal input (a set of parameters that ensure a model is reasonably defined) and a transformation generator. The simplified 2D representations are input to an algorithm which outputs the corresponding manipulation plans to fold garments. Their approach successfully enables a robot to manipulate garments of various shapes. However, fitting 2D polygon representations with garments need human intervention. Yan et al. \[11\] introduced a model-based towel and rope flattening approach based on a contrastive estimation. A contrastive forward model (CFM) is proposed, which decides whether a predicted future object state belongs to the current object state with a specific action. In their experiments, images of towels and ropes with crumpled starting and goal configurations are input into the CFM, which outputs a sequence of object transformations between starting and goal configurations and corresponding actions. Their model achieved the best performance among other forward models. Still, they only validated their approach with simply-structured objects (towels and ropes), and the manipulations are time-consuming (about 15 minutes) for a towel flattening task.

To find proper forces and manipulation plans for picking up deformable objects (fruits), Lin et al. \[4\] proposed a “liftability test”, which is an algorithm to decide whether an object can be picked up with a given force by monitoring and analysing the deformations of the object. The objects were simulated by a finite element method (FEM) used for deformation monitoring and analysis. In their experiments, objects were simulated and modelled by FEM. FEM uses simulated meshes to model those objects and a simply-structured object (such as fruits) may take hundreds of meshes to be modelled, making FEM simulations computationally costly. If FEM simulations were to be adopted for complex structured objects (such as garments), more meshes would be needed to model the object and would require significant computational resources and delay real-time monitoring of its deformations.

Yu et al. \[6\] introduced a model-based simulation approach to classifying the outcomes of robot-assisted dressing tasks. They trained hidden Markov models (HHMs) with simulated haptic data generated from an Nvidia PhysX simulator to classify the outcomes, tested the trained models in a real
environment, and achieved high accuracy in classifying dressing tasks. One of the critical steps in their experiments is to optimise the Nvidia PhysX simulator with real-world haptic data such that the simulator can output simulated haptic data to train the HHMs. However, optimising the simulator during the training period takes a significant time (16.8 hours), decreasing the efficiency of their approach.

B. Data-centric approaches: reinforcement and imitation learning

McConachie et al. [14] proposed a multiarmed bandit approach for deformable object manipulation. Instead of defining specific models for each type of deformable object, their approach features selecting models for manipulation that match deformable objects from the multi-armed bandit approach. They balanced explorations of models that fit objects and exploitations of high-utility models. Their approach addressed the limitations of models only for specific object types, and they validated their approach in simulated environments. Balaguer et al. [7] introduced combining imitation and reinforcement learning to fold towels, featuring a “momentum fold” strategy where a robot swings towels to learn towel dynamics. They demonstrated that using the imitation technique in reinforcement learning can lead to faster algorithm convergence than reinforcement-only approaches. However, the initial configurations of the towels in their experiments are already flattened. Folding those towels is easier than folding towels from crumpled initial configurations, which are the main focus of this paper.

C. Data-centric approaches: finding grasping points

Triantafyllou et al. [17] proposed to find grasping points of garments from their hanging configurations by finding common features in garments. They created a ‘dictionary’ of features of grasped garments: junctions, edges and folds. By matching features in the ‘dictionary’ with the features from grasped garments, they used pre-designed manipulation plans to unfold garments. They advanced state-of-art by not considering models for garments or robotic skills for manipulating garments with reinforcement-imitation learning. However, their approach would fail to manipulate unseen garments without a large-scale database to train a deep neural network to recognise hanging garments. Maitin-Shepard et al. [19] proposed detecting the grasping points of towels by observing the grasped configurations of garments by dropping and re-grasping them until the grasped configurations are recognised. In their experiments, a robot unfolds them. Their approach required rotations and re-grasps, which incurred in extra manipulation operations. In Li et al. [21], the authors advanced their previous approach in [16] by proposing a feature extraction approach to match reconstructed 3D mesh models with the 3D mesh models in the database. However, rotations and re-grasps are still needed for matching.

In [22], we have proposed recognising grasped configurations of garments similar to [17], [19], [16], [21] and using pre-design manipulation plans to flatten garments. Our previous approach features a deep neural network called a known configuration network (KCNet) trained on grasped configurations (called known configurations), which enables recognising known configurations of unseen garments. Compared with previous work [17], [19], [16], [21], our approach is more robust to unseen garments; however, due to training on garments of various shapes, the previous KCNet could not distinguish between shirts and sweaters due to their shape similarity. Our approach advances our previous approach by introducing prior knowledge on garments shapes, which helps identify the known configurations of garments.

III. MATERIALS AND METHODOLOGY

A. Flattening pipeline

The flattening task in this paper is defined as the robot sliding the garment over the edge of a table after the garment has been unfolded in the air based on prior knowledge of its shape and known configuration. Therefore, the proposed flattening pipeline consists of three core stages: shape prediction, known configuration prediction and manipulation stages as shown in Fig. 3.
The flattening starts with a robot grasping a garment from a table to a point above the table (approx. 1m above the table). During the motion trajectory, the robot predicts the garment’s shape by continuously perceiving it as it is being translated to estimate its shape. For this, we employed a garment similarity network (GarNet) method to predict garment shapes. Compared with traditional approaches of using single images to predict garment shapes, GarNet enables a robot to continuously perceive the video frames of garments being grasped from a table to a point above the table.

Specifically, the robot grasps a garment from a table to a point above the table (approx. 1m above the table). A sequence of depth images (as captured while the robot moves the garment from the table) are input into GarNet, which maps each image into a Garment Similarity Map (GSM) defined as a Garment Similarity Point (GSP). On the GSM, each garment shape learned by GarNet is clustered together, and the GSP from the input depth image is mapped into one of these shape clusters. After the robot perceives at least 20 images, if 80% of GSPs from the input images belong to the same shape, the shape cluster class is used as the predicted shape for the grasped garment. Further details and experiments can be found in [23]. We use a GarNet trained on depth images based on the experiment results from [23].

C. Robotic garment flattening with known configuration network (KCNet)

After garment shapes are successfully predicted by GarNet in section III-B, the robot will recognise the known configurations of garments and match pre-designed manipulation plans with the recognised known configurations to flatten garments. Figure 2 shows the pipeline of the proposed robotic garment flattening pipeline.

We define a known configuration as the configuration of a garment in a hanging state after being grasped from a different configuration. That is, the known configurations of garments highly depend on grasping points because of gravity,
regardless of their starting configurations, which means that the irregular and complex starting configurations of garments are converted into stable, constant known configurations.

For this, we have adopted KCNet (described in [22]) such that it can recognise the known configurations of garments based not only on images of their known configurations but also on prior knowledge of garment shapes, which GarNet predicted (Section III-B).

KCNet consists of a ResNet18 convolutional network and three linear networks. A KCNet consists of a ResNet18 convolutional network [24] and three linear networks. An image of an Known Configuration is an input to the KCNet to get its extracted features. The garment shape, as predicted by GarNet is encoded to a one-hot vector and then input into a linear network to get its latent representation. The feature extracted from the image and the latent representation from the one-hot vector of the predicted shape is concatenated, which is then input into a linear network to output the known configuration of the garment. Similar to [22], our KCNet is trained with a Negative Log-Likelihood Loss (NLLLoss).

D. Pre-designed manipulation plans

After the known configurations of garments are successfully recognised from KCNet, the robot matches pre-designed manipulation plans with the recognised known configurations. A pre-designed manipulation plan is a sequence of end-effector commands for the robot to flatten a garment. Each gripper command contains 16 parameters, including XYZ coordinates of the right and left grippers, orientations (defined as a quaternion) of right and left grippers, a choice of grippers (left or right) and gripper status (open or close). On average, 18 gripper robotic actions are needed to flatten a garment.

Similar to [16], a manipulation plan follows a three-step manipulation rule: find the first grasping point, the second grasping point, and stretch and flatten garments. A known configuration has a corresponding pre-designed manipulation plan. We have ten pre-designed manipulation plans for each shape of garments, totalling 50 manipulation plans as we have five shapes in our experiments.

E. Fine-tuning Garment Grasping

We pre-defined grasping points in our pre-designed manipulation plans, consisting of the first and second grasping points. However, garments deform irregularly, so those pre-designed grasping points must be fine-tuned for stable grasps during a flattening operation. Therefore, we proposed to locate grasping points by finding the closest point between the robot’s gripper and the garment in a point cloud.

Firstly, our Xtion camera is placed in front of the robot for recognising known configurations of garments. To bring the camera and the robot into alignment, we use Khan’s et al. [25] hand-eye calibration approach. After the robot and camera coordinate systems are aligned, we capture point clouds of garments. From the captured point cloud, the points belonging to garments have a specific range of values because they have similar distances to the camera. Therefore, points belonging to garments can be easily identified and segmented from the points belonging to other objects (for example, the Baxter robot). Then, we find points that have the smallest distances with pre-designed grasping points and define them as fine-tuned grasping points used for grasping garments. We can ensure that grasping points are on garments and stable grasps can be generated on-the-fly in this way.

In our experiments, we do not need to identify local features or landmarks of garments as used in previous research ([15], [16], [17]); we only need to do visual servoing when known configurations are recognised. Therefore, our data-centric approach does not require constructing models for garments and updating manipulation plans for unseen garments. Figure 3 shows an example of pre-designed grasping points (coloured in yellow) and fine-tuned grasping points (coloured in green) for the first and second grasping points (described in section III-D) of a sweater. We can observe that the pre-designed grasping points are near to the garment but the fine-tuned grasping points can ensure stable grasps compared with pre-designed grasping points.

F. Robotic Setup

A Baxter dual-arm robot is employed for our experiments, controlled by MoveIt software. A four-legged table is placed in front of the Baxter robot to place garments in front of the robot. Two Xtion stereo cameras are set up facing the Baxter robot at two locations to enable wide- and narrow-field image capturing. Both cameras capture RGBD images (i) for predicting garment shapes and (ii) for recognising garment known configurations and fine-tuned pre-designed points (Section III-E). For predicting garment shapes, the garment similarity network (Section III-B) needs to observe how garments deform during grasping. Therefore, the camera must capture garment deformations as the robot it up; hence, this camera is placed at a larger distance from the Baxter robot. Conversely, for recognising the known configurations of garments (Section III-C), the camera needs to capture the details of garments and is, thus, placed close to the Baxter robot. We implement our manipulation plans using the robot operating system (ROS, Kinect version) and the Ubuntu 16.04 environment, installed on a computer with a Core i7 CPU and
an Nvidia 1080 Ti GPU. An openni2 camera-controlling system controls the cameras. The sampling rate is 30 Hz for both cameras.

We use five shapes of garments with four different instances for each shape, totalling 20 garments tested. The shapes considered in this paper are jeans, shirts, sweaters, towels and t-shirts. Garments have different colours, textures and materials. Jeans are made of denim, shirts are made of polymers, and sweaters, t-shirts and towels are made of cotton. Experiments are conducted in a robot laboratory, where lighting conditions are set the same during each experiment. The laboratory is enclosed, which means that the external environment is stable and does not affect the experiments.

IV. Experiment Setup

A. Database

We used the database that we constructed in [22]. For each grasping point (equivalent to each known configuration), we captured 100 RGB and depth images. For each garment instance, the robot randomly grasps the garment and lifts it above the table ten times, totalling 19,269 images in our database. Note that 731 images had image artefacts or noise. Thus, we removed them from our database. Each image has a resolution of $256 \times 256$.

B. KCNet training experiments

As described in [III], KCNet consists of ResNet18 and three linear networks, where the features of predicted shapes and the images of known configurations are concatenated and used for recognising known configurations.

We implement a leave-one-out cross-validation method in the experiments to validate our approach. Our KCNet is trained on a small database. Thus, implementing the leave-one-out cross-validation method ensures that our KCNet has a robust performance in recognising known configurations of garments. The database is divided into training and testing groups. The images from three of four garment instances for each shape serve as a training group, while the images of the remaining instance serve as the testing group. We ensure that the garments in testing groups never appear in training groups, which means testing garments are unseen by GarNet and KCNet. Therefore, four experiments have been conducted, where testing and training groups differ each time. We averaged the recognition accuracies across the four experiments, and the averaged value is used as the accuracy of the KCNet. We compare KCNet combined with GarNet in this paper with the KCNet in [22]. We conduct an ablation study on recognition accuracy on depth, RGB, and RGBD images to investigate image formats’ impacts on recognising known configurations. We will use the KCNet combined with GarNet with the highest accuracy for our robotic garment flattening pipeline. We used an Adam optimiser to train KCNet, which is regulated by a learning-rate scheduler with a step size of eight epochs and a decay factor of $10^{-1}$. The initial learning rate for the optimiser is $10^{-3}$, the number of training epochs is 30 and the batch size is 64.

C. Robotic Garment Flattening Pipeline

We performed 20 garment flattening operations to validate our robotic garment flattening pipeline experimentally. For this, we recorded starting configuration states, final configuration states, times for each manipulation step, and success rates (the percentage of successful tests in all tests). A starting/final configuration state is computed as:

$$\text{Starting State} = \frac{S_{\text{start}}}{S_{\text{goal}}} \times 100\% \quad (1)$$

$$\text{Final State} = \frac{S_{\text{ending}}}{S_{\text{goal}}} \times 100\% \quad (2)$$

$S_{\text{start}}$ refers to the area in pixels of the starting configurations of garments, and $S_{\text{ending}}$, the area in pixels of the final configurations of garments. $S_{\text{goal}}$ represents the area in pixels of the goal configurations of garments. A goal configuration is a garment configuration when the garment is fully flattened (referenced in [26]). An example of goal configurations is shown in Figure [4]. To capture the starting, final and goal configuration, we positioned a camera at a fixed distance above the table, and captured images at each state and cropped them using the table’s corners and edges as guidelines.

V. Experiment Results

A. KCNet training and ablation study results

Tables [I] and [II] show the results for unseen garment instances of KCNets trained on different image types. We can observe that the KCNet trained on depth images (92% for the KCNet with prior knowledge of garment’s shapes and 89% for the KCNet without prior knowledge of garment’s shapes) has the best performance compared to RGB (78% with prior knowledge of garment’s shapes and 73% without) and RGBD (78.5% with prior knowledge of garment’s shapes and 81.5% without). Depth images capture structural and spatial information about garments, compared to RGB images which capture texture information. Structural and spatial characteristics are similar between garments of similar shapes, while textures are easily affected by lighting conditions and the garment’s colours that vary across different garments. Also, compared with KCNets without prior knowledge of garment’s shapes [22], KCNets with prior knowledge of garment’s shapes demonstrate better performance on all types of images, specifically on recognising known configurations of shirts and sweaters. That is, shirts and sweaters share similar structures, thus, identifying differences between their known configurations based only on their depth images is challenging. The latter is overcome by having prior knowledge of the garment’s shape.
B. Robotic Garment Flattening Results

Table [III] shows the starting configuration states, running times for each step, final configuration states and success rates. On average, the Baxter robot requires 29 seconds to predict the garment shapes with GarNet. For garment flattening, Baxter needs, on average, 79.8 seconds to grasp the first grasping point, 60.2 seconds to grasp the second grasping point, and 52.6 seconds to stretch and flatten a garment. Compared to the state-of-art [11], where the robot needs approximately 15 minutes to fold a piece of garments, our approach has a faster manipulation operation. Traditional approaches update robotic manipulation plans in real-time, which requires costly computations. Also, updating robotic manipulation plans may result in robots’ unnecessary actions, taking additional time.

On average, the starting configuration states are 27.80% (ref. Section [IV-C]), while it increases to 80.1% for final configuration states. The results show that the robot successfully flattened garments and reached a high average final configuration state (80.1%) with our approach. We found that jeans, shirts, sweaters and t-shirts have better final configuration states than towels. When Baxter finds the second grasping point for flattening the towels, it sometimes grasps the incorrect corners because the towels are symmetrical, resulting in final configuration states lower than other garments.

Table [IV] shows a comparison between our approach with state-of-art. Sun et al. [27] was one of the first to demonstrate dual-arm flattening of towels and achieved a dual-arm success rate of 46.3%; their approach required approx. 20 minutes to flatten a squared towel with one or two folds. Li et al. [16] and Martin et al. [28] achieved high manipulation success rates; however, they required re-grasping garments to ensure that the robot recognised the pose (equivalent to the known configuration in this paper). Their approaches, therefore, required longer manipulation time. Li et al. [16] achieved a lower pose estimation accuracy than our approach because they trained their network with simulated data. Martin et al. [28], Yan et al. [11] and Sun et al. [27] only tested towels in their experiments, while we used five different shapes of garments. Overall, our experiments achieved the fastest manipulation and second highest known configuration recognition accuracy but have a lower success rate than other approaches.

The average success rate for our flattening approach is 66.7%. We can observe that jeans and towels have the lowest success rates (57.1%). Jeans are heavier than other garments, so they sometimes slip from the robotic grippers and result in failures. As we mentioned, towels are sometimes grasped from wrong grasping points because they are symmetric, resulting in failure cases. In future research, we plan to use grippers suited for manipulating garments (e.g. [27]) for grasping and flattening jeans and better manipulation plans for grasping and flattening towels. Figure [I] shows three successful garment flattening operations of three shapes (a shirt, a jean and a sweater). A video demonstration can be found at [https://youtu.be/j7yEbJcAgDM].

VI. CONCLUSION

In this paper, we have proposed an effective robotic garment flattening pipeline where a Baxter robot predicts garment shapes, recognises their hanging state (known configurations of garments) based on their shapes and depth images, and uses pre-designed manipulation plans to flatten them. We found that our approach achieved an accuracy of 92% for recognising known configurations, and the robot takes, on average, 29 seconds to predict the garment shapes and 192.6 seconds on average to manipulate and flatten garments.

In this paper, we only use garments from the five shapes because manipulation plans are designed based on these five shapes. Garments with an unknown shape (for example, shorts) can not be manipulated, limiting our approach to specific shapes of garments, and this is an open problem in deformable object manipulation. We plan to implement a reinforcement learning approach [29] to improve the success rate of garment flattening, where the robot learns basic skills which can be composed to manipulate any garment shape. Another limitation of our flattening pipeline is that we use two cameras to
TABLE III
ROBOTIC GARMENT FLATTENING RESULTS

| SHAPE | Starting State | Shape Prediction | Grasping Point 1 | Grasping Point 2 | Flattening | Total Time | Final State | Success Rate |
|-------|----------------|------------------|------------------|------------------|-----------|------------|-------------|--------------|
| Jeans | 40.51%         | 26.0s            | 55.5s            | 50.0s            | 72.5s     | 204.0s     | 87.75%      | 57.1%        |
| Shirt | 27.04%         | 30.0s            | 71.0s            | 47.5s            | 42.0s     | 190.5s     | 96.30%      | 66.7%        |
| Sweater | 25.98%      | 28.8s            | 86.0s            | 62.0s            | 66.0s     | 242.8s     | 76.65%      | 80.0%        |
| Towel | 23.85%         | 30.0s            | 82.5s            | 71.0s            | 41.0s     | 224.5s     | 65.44%      | 57.1%        |
| T-shirt | 21.61%       | 30.0s            | 104.0s           | 70.5s            | 41.5s     | 246.0s     | 74.06%      | 80.0%        |
| Average | 27.30%         | 29.0s            | 79.8s            | 60.2s            | 52.6s     | 227.6s     | 80.10%      | 66.7%        |

TABLE IV
COMPARISON WITH STATE-OF-ART (N/A MEANS THAT THE VALUE IS NOT REPORTED IN THE PAPER.)

| Method  | Estimation Accuracy | Running Time | Success Rate |
|---------|---------------------|--------------|--------------|
| Sun [27] | N/A                 | 1.200        | 46.3%        |
| Li [16]  | 83%                 | N/A          | 80.0%        |
| Yan [11] | N/A                 | 900s         | N/A          |
| Maitin [28] | 96%             | 1478 s       | 81%          |
| Ours     | 92%                 | 221.6s       | 66.7%        |

recognise the garment’s shapes and known configurations. In future work, we plan to use only one camera for both shape and known configuration predictions and investigate active vision approaches to control the camera’s viewpoint.

REFERENCES

[1] Y. Gao, H. J. Chang, and Y. Demiris, “Iterative path optimisation for personalised dressing assistance using vision and force information,” in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016, pp. 4398–4403.
[2] B. Willimon, I. Walker, and S. Birchfield, “A new approach to clothing classification using mid-layer levels,” in 2013 IEEE International Conference on Robotics and Automation, 2013, pp. 4271–4278.
[3] S. Choi, G. Zhang, T. Fuhlbrigge, T. Watson, and R. Tallian, “Applications and requirements of industrial robots in meat processing,” in 2013 IEEE International Conference on Automation Science and Engineering (CASE), 2013, pp. 1107–1112.
[4] H. Liu, F. Guo, F. Wang, and Y.-B. Jia, “Picking up a soft 3d object by “feeling” the grip,” The International Journal of Robotics Research, vol. 34, no. 11, pp. 1361–1384, 2015.
[5] L. Zaidi, J. A. Corrales, B. C. Bouzgarrou, Y. Mezouar, and L. Sabourin, “Model-based strategy for grasping 3d deformable objects using a multi-fingered robotic hand,” Robotics and Autonomous Systems, vol. 95, pp. 196–206, 2017.
[6] 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), IEEE, 2017, pp. 6044–6051.
[7] 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.
[8] E. Pignat and S. Calinon, “Learning adaptive dressing assistance from human demonstration,” Robotics and Autonomous Systems, vol. 93, pp. 61–75, 2017.
[9] Y. Tsurumine, Y. Cui, E. Uchibe, and T. Matsubara, “Deep reinforcement learning with smooth policy update: Application to robotic cloth manipulation,” Robotics and Autonomous Systems, vol. 112, pp. 72–83, 2019.
[10] B. Tawbe and A.-M. Cretu, “Acquisition and neural network prediction of 3d deformable object shape using a kinect and a force-torque sensor,” Sensors, vol. 17, no. 5, p. 1083, 2017.
[11] W. Yan, A. Vangipuram, P. Abbeel, and L. Pinto, “Learning predictive representations for deformable objects using contrastive estimation,” arXiv preprint arXiv:2003.05436, 2020.
[12] L. Twardon and H. Ritter, “Learning to put on a knit cap in a head-centric policy space,” IEEE Robotics and Automation Letters, vol. 3, no. 2, pp. 764–771, 2018.
[13] J. Matas, S. James, and A. J. Davison, “Sim-to-real reinforcement learning for deformable object manipulation,” in Conference on Robot Learning. PMLR, 2018, pp. 734–743.
[14] D. McConachie and D. Berenson, “Estimating model utility for deformable object manipulation using multiarmed bandit methods,” IEEE Transactions on Automation Science and Engineering, vol. 15, no. 3, pp. 967–979, 2018.
[15] J. Qian, T. Weng, L. Zhang, B. Okorn, and D. Held, “Cloth region segmentation for robust grasp selection,” in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020, pp. 9553–9560.
[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). IEEE, 2015, pp. 1382–1388.
[17] D. Triantafyllou and N. Aspragathos, “Garment type agnostic robotic unfolding of garments from random configuration,” in International Conference on Robotics in Alpe-Adria Danube Region. Springer, 2022, pp. 487–495.
[18] 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. 246–267, 2012.
[19] 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. IEEE, 2010, pp. 2308–2315.
[20] D. Seita, N. Jumali, M. Laskey, A. K. Tanwani, R. Berenstein, P. Baskaran, S. Iba, J. Canny, and K. Goldberg, “Deep transfer learning of pick points on fabric for robot bed-making,” in The International Symposium of Robotics Research. Springer, 2019, pp. 275–290.
[21] Y. Li, Y. Wang, Y. Yue, D. Xu, M. Case, S.-F. Chang, E. Grinspun, and P. K. Allen, “Model-driven feedforward prediction for manipulation of deformable objects,” IEEE Transactions on Automation Science and Engineering, vol. 15, no. 4, pp. 1621–1638, 2018.
[22] L. Duan and G. Aragon-Camarasa, “Recognising known configurations of garments for dual-arm robotic flattening,” 2022. [Online]. Available: https://arxiv.org/abs/2205.00225
[23] L. Duan and G. Aragon-Camarasa, “A continuous robot vision approach for predicting shapes and visually perceived weights of garments,” IEEE Robotics and Automation Letters, vol. 7, no. 3, pp. 7950–7957, 2022.
[24] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[25] A. Khan, G. Aragon-Camarasa, L. Sun, and J. P. Siebert, “On the calibration of active binocular and rgbd vision systems for dual-arm robots,” in 2016 IEEE International Conference on Robotics and Biomimetics (ROBO), 2016, pp. 1960–1965.
[26] Y. Wu, W. Yan, T. Kurutach, L. Pinto, and P. Abbeel, “Learning to manipulate deformable objects without demonstrations,” arXiv preprint arXiv:1910.13439, 2019.
[27] 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). IEEE, 2015, pp. 185–192.
[28] 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, 2010, pp. 2308–2315.
[29] A. Pore and G. Aragon-Camarasa, “On simple reactive neural networks for behaviour-based reinforcement learning,” in 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020, pp. 7477–7483.