Semantic State Prediction in Robotic Cloth Manipulation

Georgies Tzelepis[1], Júlia Borràs[1], Eren Erdal Aksoy[2], and Guillem Alenyà[1]

[1] Institut de Robòtica i Informàtica Industrial CSIC-UPC,
Llorens i Artigas 4-6, 08028 Barcelona, Spain
{gtzelepis,jborras, galenya}@iri.upc.edu,
[2] Halmstad University, Center for Applied Intelligent Systems Research, Halmstad, Sweden
eren.aksoy@hh-se

Abstract. State estimation of deformable objects such as textiles is notorious difficult due to its extreme high dimensionality and complexity. Lack of data and benchmarks is another challenge impeding progress in robotic cloth manipulation. In this paper, we make a first attempt to solve the problem of semantic state estimation through RGB-D data only in an end-to-end manner with the help of deep neural networks. Since neural networks require large amounts of labeled data, we introduce a novel Mujoco simulator to generate a large-scale fully annotated robotic textile manipulation dataset including bimanual actions. Finally, we provide a set of baseline deep neural networks and benchmark them on the problem of semantic state prediction on our proposed dataset.

Keywords: Robotics, Simulation, State Estimation, Deformable Objects

1 Introduction

Deformable objects can have diverse poses, which introduces the biggest challenge for robotic perception and manipulation. This is mainly due to the high dimensionality and complexity of the problem, which makes it non-trivial to detect and identify the deformation type of the surface. In robotics, state estimation in a continuous cloth manipulation process is still an essential prerequisite for robot monitoring, learning, and imitation.

The dominant approach for perception tasks in the past decade is the use of deep learning techniques, which most of the time trained in an end-to-end fashion. However, the main focus of such methods has been on rigid objects [1] and their contributions are mainly around the action recognition spectrum rather than the classification. Also, most recent works on the manipulation of garments heavily rely on a limited number of simulated and/or real-world data samples, which are partially or fully restricted to public use [2].

In order to solve the state estimation task in an end-to-end manner with the use of deep learning, it is required to access a large scale dataset under a grasp manipulation framework. Since data annotation is time-consuming and costly, it is preferred to be done with the use of a simulator generating manipulation sequences that are close to real world manipulations.

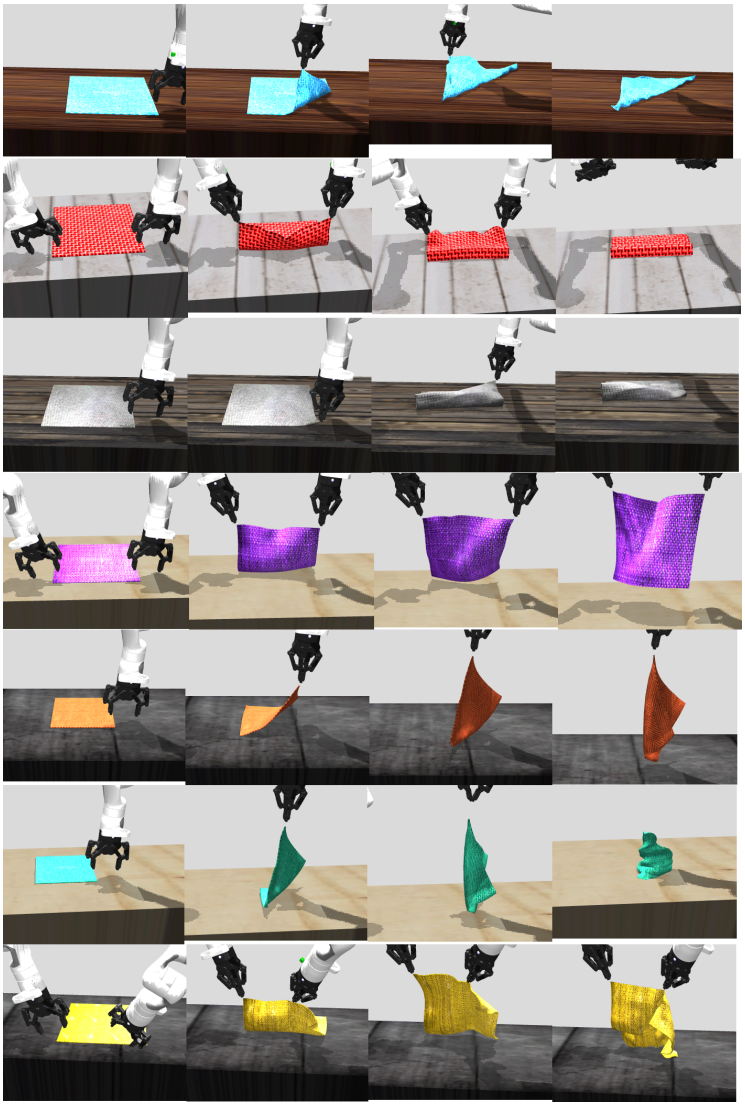

Fig. 1. Seven different manipulations involving nine tasks and ten different states. Each row corresponds to a manipulation while the columns are selected states. The first column shows the initial state in the simulation where the cloth is **flat**. The second column in the manipulations corresponds always to a **semi-lifted** state with either **one** or **two** grippers with the exception of the final row where the **middle grasps** results in the cloth being crumpled. In the third column the first four rows and the last continue having the same state while the other two switched into a **semi-lifted crumpled** state. The last column shows the end states for each manipulation, from top to bottom: **diagonally folded**, the next two are **sideways folded**, **lifted with two grippers**, **lifted with one gripper**, **crumpled** and **middle grasping with two grippers**.

In this work, we aim to investigate and answer whether it is possible to semantically classify states of continuous manipulations. Initially, a framework [3,4] was chosen as the basis to define and generate the semantic states by introducing a novel concept of semantic state representation which defines a unique deformation on the surface. To encode each perceived manipulation, a sequence of such states was then generated. Unlike other approaches which either disregard the presence of gripper [5] or are focused entirely on the manipulation through point grasping with a sole gripper [6], in this framework the grasping state plays an important role in the detection and identification of a semantic state.

Most simulators have been employed to solve various reinforcement learning tasks by focusing around the control and the trajectories of the actions rather than the vision. Several attempts have been made in the past for such tasks in order to simulate the deformable dynamics. However, most of the tasks they perform are with one robotic arm visible, or even if there are two manipulators, they are not visualized together [7].

In order to generate the data, we develop a garment manipulation simulator in MuJoCo [8] with two Kinova arms as manipulators. Unlike previous approaches, we simulate the scene with a camera view covering the garment, the background and both grippers. Furthermore, we generate scenarios which can provide sequences that a real robot can perform in real world manipulation, i.e., the grippers can go out of the camera field of view or occlude a portion of the garment features.

Once the simulated data is generated and annotated, we benchmark different deep neural networks running in an end-to-end manner to solve the state estimation task. To our knowledge, our approach is completely novel and thus it did not make sense to create a neural network to perform this task but rather create baselines which are performed by well-established and known deep learning models for images. However, since the computational cost and resources that are needed to train the networks can be from moderately to extremely high, in order to provide results that are easily accessible and reproducible with limited resources, those baselines will be done with the use of transfer learning and fine-tuning for already pretrained networks on other datasets.

To summarize our work contributes the following:

- We developed a simulator on MuJoCo for garment manipulation with two grippers from the robot's point of view and release the source code in order to encourage the investigation and research of deformable object manipulation.
- We generated an RGB-D dataset and annotated the data according to a grasping framework for deformable object manipulation and release it for public usage.
- We verify that under the current framework, the state estimation problem can be solved in an end-to-end manner with the help of deep neural networks.

2 BACKGROUND AND RELATED WORK

2.1 Grasping Manipulation

To enable learning from human demonstrations or high-level task planning in the context of cloth manipulation, scene state recognition is one of the challenging open issues.

Recent learning-based solutions for cloth manipulation have represented the cloth during a manipulation as the RGBD images of the cloth on the table, and they use Euclidean distance between pixels to identify close or equal states [9,10,11,6,5]. Other similar works use additional information from the robot, like in [12] that uses RGB image plus the robot arm joints and grippers state or [13] that uses the RGB image and robot arm joints. In [14] the state is represented as a 32x32x16 binary voxels, with 1 where there is cloth and 0 in empty voxels. In all these works, the scene state definition does not include information of the interaction with the grasping agent or the environment. In other words, the basic scene state is the cloth when it is not touched by the robot.

In [3] a framework is introduced to describe textile grasps based on the geometry of the prehension agents, including extrinsic geometries from the environment. Later in [4], we extended that framework to define a scene state with very basic cloth configuration semantic labels but including grasping state and environmental contacts. This novel state definition leads to a much thinner task segmentation than previous works, because every re-grasp, contact with the environment or change in cloth configuration triggers a new segment. We believe this is necessary to approach complex tasks where several re-grasps are needed before the cloth is fully released, to obtain simpler action primitives that can be reused in different tasks and contexts, similarly as it was done for rigid objects [15].

Simplified representations of scene states based on the contact interactions between the hands, the object and the environment was used in the past in the context of manipulation of rigid objects [16], and then used for recognition, segmentation [17] and learning manipulation actions to be executed by a robot [18].

2.2 Deformable Simulation and Data

Data-driven methods using deep learning techniques often require a substantial amount of data. Most of the simulators that have been used were developed for the manipulations of rigid objects, mostly due to the difficulties to simulate the deformable's dynamics during the manipulation task.

Recent work on the simulation of deformable objects has been done by developing simulations on PyBullet [19] and with the use of neural networks they were able to re-arrange rigid objects at first [20] with one gripper which was later extended to deformable manipulation [21]. Towel manipulation tasks have been simulated [12] but they have reported issues with the simulator codebase such as instability. Assistive Gym [22], which also uses PyBullet, has simulated collaborative robotics with humans on various tasks which some included deformable objects.

SOFA [23] is another simulation framework which provides more realistic deformations in comparisson to PyBullet and it has been used for dynamic cloth manipulation [6] in reinforcement learning tasks.

More recently SoftGym [7] used Nvidia's FleX simulator to perform manipulation tasks on deformable objects which are modelled by objects in a particle and position based dynamical systems. While this work supported multiple manipulators at each time, unlike MuJoCo it did not have the availability of various simulated grippers

modelled after real world robots. SAPIEN [24] and ThreeDWorld [25] also included deformables in their simulations but did not include tasks for their manipulation.

Table 1. Definition of the semantic states

Label	Grasp type*	Grasp location	Cloth configuration	Description	Image
Flat	Π_e	-	Flat	the initial state after the simulator resets. The cloth has no deformations is in contact with the table, not with the grippers.	
Sideways Folded	Π_e	-	Folded	Folded sideways and it's in contact with the table, not the grippers.	
Diagonally Folded	Π_e	-	Folded	Folded diagonally and in contact with the table, not the grippers.	
Crumpled	Π_e	-	Crumpled	The cloth is deformed enough so that it can't go back to a flat configuration without additional manipulation.	
Flat semi-lifted with one gripper	$PP + \Pi_e$	Corner	Flat	The gripper is grasping a corner the cloth is still in contact with the table. This state can be reversed to its previous state.	
Crumpled semi-lifted with one gripper	$PP + \Pi_e$	Corner	Crumpled	a portion of the cloth is in contact with the table but the cloth cannot reverse to the Flat state without the use of a second manipulator.	
Flat semi-lifted with two grippers	$2PP + \Pi_e$	R&L corners	Flat	as the previous state, but it requires both grippers to be grasping a corner	
Lifted with one gripper	PP	Corner	Crumpled	occurs once the contact between the cloth and the table ceases	
Lifted with two grippers	$2PP$	R&L corners	Flat	the grippers hold the cloth flat and hanging in the air, without contact with the table	
Middle grasping with two grippers	$2PP + \Pi_e$	Two corners on an edge	Crumpled	is the state where one gripper grasps one corner while the other grasps a point on the adjacent edge of the garment and the cloth is in contact with the table.	

Grasp type notation from [3]: PP grasp stands for a pinch grasp, $2PP$ for a double pinch with both grippers and Π_e is the extrinsic contact with a plane (the table).

A few attempts have also been made to generate a dataset for manipulation tasks but in a rather limited spectrum of actions and all of them did not include the grippers in their input or generate entire continuous sequences of manipulations. State estimation was performed by generating solely depth synthetic data of hanging garments [2] from multi-view points and generating a large scale dataset which was used to train a neural network. Another approach uses RGBD images to generate a mesh of a 3D deformable object by the minimization of an energy function [26]. Unfortunately both those data are unavailable to the public and/or limited in their use-cases.

3 Simulation of Garment Manipulation and Data Annotation

MoJoCo is the simulator of our choice and the focus has been on point grasp manipulation actions which are described in [3]. Two Kinova arms were chosen over floating grippers due to their more limited trajectories which are positional dependant and their collision reaction with rigid objects such as the table. The deformable object that is simulated is a towel.

3.1 Simulation

The cloth of the simulator is generated by a mesh of vertices which are in their own turn 3D objects. The size of the cloth depends on the number of vertices and the length of the edges. The downsize with this approach is edges' lack of mass which allows interactions between the gripper and the cloth possible only through the vertices.

A MoJoCo python interface was used along with openAI gym [27] in order to generate different simulated environments. The parameters which are needed to be specified are the edges' **length** and **stiffness**, the **size** and the **mass** of the vertices and finally the garment's and the gripper's initial position. Those parameters are randomized under some constrains which ensure the feasibility of the manipulation, i.e. the garment's initial position is within the gripper's reach. Since our main objective is to study the dynamic deformations of the object during the manipulation, we bypass the need to model a physical grasp and instead we substitute it with a fake one implemented as a binary point grasp to manipulate the textile.

Six different **environments** were generated to manipulate the garment in nine different **tasks**. The environment choice is depended solely on the initial grasp and the whether ending position of the garment's manipulation. For a better clarification how the tasks are grouped under specific environments:

One Hand Folding Sideways: Folding manipulation performed solely by one gripper. The goal state is achieved by transporting the corner point of the cloth of our choice close to a proximity of the goal location which in this case is the opposite vertex. In order to avoid deformations which would result for the cloth to crumple, speed constrains are also used (see Fig. 1, third row).

One Hand Folding Diagonally: This environment is implemented to simulate diagonal folding of the cloth with one gripper in a manner to the previous task. The only difference is that the goal location is the vertex which is located diagonally across the grasping corner (see Fig. 1, first row).

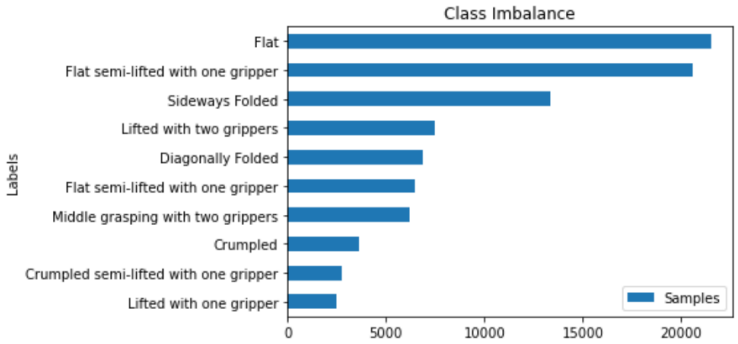

Fig. 2. The class distribution of the generated dataset. The class imbalance is resulted since several manipulations share the same states

One Hand Lifting: This environment involves three tasks. The first task is always to **lift** the cloth by grasping one corner. The goal is achieved by lifting it higher than the diagonal distance of the cloth (Fig. 1, fifth row). Constrains are applied on the size of the cloth since its diagonal needs to be smaller than the maximum height the kinova's gripper can reach. Following the lifting of the cloth the next two tasks can be performed. First is to simple **lower** (Fig. 1, sixth row) the garment enough so a portion of it is in contact with the table. The final task is to **drop** (Fig. 1, fourth and sixth row) the garment which is done simply by releasing the gripper.

Two Hand Folding Sideways: Folding manipulation performed by two grippers. Each gripper manipulates the cloth by grasping the corner of the cloth that is the closest to it and transporting it to the opposite vertex (Fig. 1, second row)

Two Hand Lifting: This environment involves two tasks. The first task is always to **lift** the garment be grasping the corner of the cloth that is the closest to each gripper and transport them into a height that exceeds the length of the cloth (Fig. 1, fourth row).

Two Hand Middle Grasp: Uses two grippers and is focused on grasping one corner with one gripper while the other is grasping some point in the middle of the adjusted edge that is closer to the position of the camera. After grasping it the goal is to lift this edge while part of the cloth remains in contact with table (Fig. 1, last row).

Each task runs for a limited number of simulated steps which can vary from task to task before it resets to the initial state. At the start of each episode, the base of the manipulators is initialized to a default position while the cloth and the grippers are allocated randomly under some constrains. There are no deformations on the cloth.

3.2 Data Generation and Labeling

Over the course of each simulation, RGB-D images and the mesh of the cloth are generated at each simulation step. The simulator automatically segments the captured data in order to distinguish the cloth, the manipulator and the background. It also returns automatically annotated states each of which is defined by following the grasping-centered framework introduced in [3,4]. To increase the scale of the dataset, the simulator automatically generates various manipulation samples of the same type by altering the

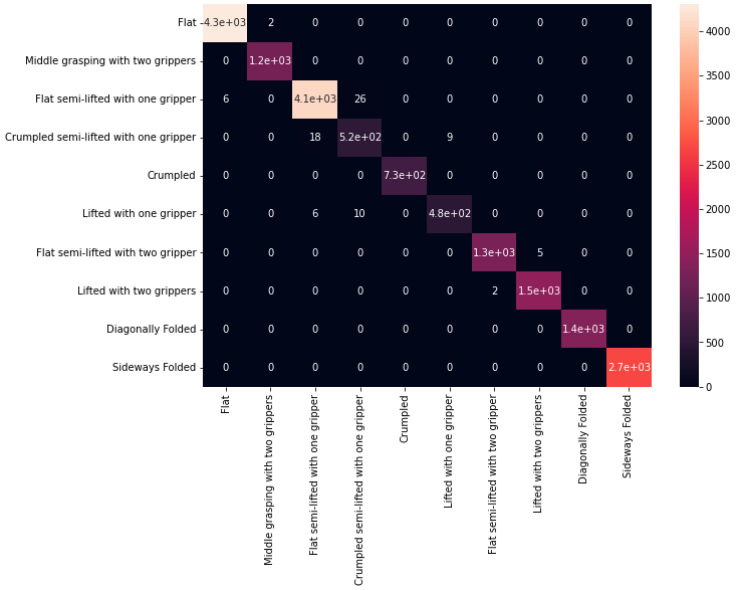

Fig. 3. Confusion matrix for the ResNet50 predictions. It can be seen that the states which are the most confusing are between the the **crumpled semi-lifted with one gripper** and **flat semi-lifted with one gripper** states.

trajectory, initial position, speed of the manipulation and the size of the garment. In order to generate a higher variety in the RGB-D data, initially domain randomization [28] was applied on the simulation, however the coloring distortion was too high and thus the results were poor even for only 3 states. Eventually, different textures where introduced to contribute into a different scenery. Five textures of different colors were used for the table (**wood, brown, white, and black**) and nine for the cloth (**black, blue, brown, green, orange, purple, red, white, and yellow**).

Another set of parameters which also contributes to the data generation is the camera position. The coordinates offset from the object to look at and the elevation, azimuth and distance are also important during data collection. In order to gather RGB-D data with more variety, a slight randomization is introduced as well.

By performing the nine continuous manipulation tasks, we were able to generate and label ten states listed in Table 1

3.3 Deep Learning

To verify our approach we opted for the use deep neural networks for our state estimation problem. Further to show that our approach can be replicated easily without the use of lots of resources we verify our results with the use of transfer learning. However to make sure that our comparisons are meaningful without bias we decided to use networks under some criteria: (1) they are trained on the same dataset, (2) they have the

same input resolution for the images and finally (3) we only substituted the final layer to fit our the number of states that are we want to estimate and train on this layer.

The networks of our choice were ResNet-50 [29], ResNeXt-50 [30] and EfficientNet [31]. We discarded the choice of using Visual Transformers [32] despite being the state of the art in classification we didn't have meaningful results since it probably needs to be retrained on our dataset and thus refrain from our initial approach of transfer learning and fine-tuning.

4 EXPERIMENTS

As a dataset we generated one of 91380 samples and we did the training validation split of 80 to 20 as seen in Table 3 while making sure we didn't over-sampled from one set over the other on a class. It is also noticeable that there is a class imbalance, see Fig. 2. However this is to be expected several manipulations share the same states and all of them start from the initial state where the cloth is flat.

To simulate a more realistic approach from a robotic point of view, we added some random permutation on the camera location of 10 degrees vertically and horizontally as well as a slight noise on the camera's distance from the cloth, on which the camera was locked on.

Finally we experimented with several optimizer and schedulers and found out that by using stochastic gradient descent with warm restarts we managed to boost our results

Table 2. Accuracy table

States	ResNet50	ResNeXt50	EfficientNet-b0
Flat	0.9995	1.0000	1.0000
Middle grasping with two grippers	1.0000	1.0000	1.0000
Flat semi-lifted with one gripper	0.9922	0.9942	0.9920
Crumpled semi-lifted with one gripper	0.9504	0.9430	0.9596
Crumpled	1.0000	1.0000	1.0000
Lifted with one gripper	0.9679	0.9639	0.9759
Flat semi-lifted with two grippers	0.9961	0.9954	0.9876
Lifted with two grippers	0.9987	0.9933	1.0000
Diagonally Folded	1.0000	1.0000	1.0000
Sideways Folded	1.0000	1.0000	1.0000
Overall Accuracy	99.54%	99.51%	99.54%

State estimation accuracy per class and overall for each network.

significantly. We used transfer learning and fine-tuning on pre-trained networks on Imagenet which are available through the PyTorch framework for only 20 epochs, since we didn't intend to train any of the feature layers. All 3 networks managed to achieve over 99% accuracy on the validation set (see Table 2).

It can be observed from the confusion matrix in Fig. 3 that the most conflict cases occur between the **flat semi-lifted with one gripper**, **crumpled semi-lifted with one gripper** and **lifted with one gripper**. This is not surprising since the deformation while they change from one state to another are very difficult to be perceived. More specifically the change from **flat semi-lifted with one gripper** to **crumpled semi-lifted with one gripper** occurs when the distance between the adjacent corners of the the grasping point becomes smaller than when the cloth was **flat** (Fig. 4). We hypothesize that the wrong predictions between the **crumpled semi-lifted with one gripper** and **lifted with one gripper** is the result of the camera angle which can make rather difficult without the use of depth whether the cloth has any contact with the table or not (Fig. 4).

To explain this surprising high accuracy we need to examine the current limitations of the simulator and the dataset. First, as it can be observed in Table 2 the network had almost perfect predictions for states which are easy to perceive like the initial state where the garment is **flat**, the **flat semi-lifted with one gripper** and the **sideways folded**. However due to the class imbalance of our dataset those three states are almost half of the dataset samples and thus contributing to the over all accuracy considerably more than the **crumpled semi-lifted with one gripper** and **lifted with one gripper** which have also the most conflict cases (see Fig.2 and Table 2). Second, the grippers in the current simulation are clearly visible and it is easy to observe when they are open or closed (Fig. 1), thus, making very clear the difference between **flat** and the rest of the states that involve a closed gripper. Third, we discarded the simulation of dynamic grasping actions, i.e. the initial grasp of the cloth after the initial state, and thus, the dataset does not include intermediate state changes which are hard to estimate correctly. Fourth, some short transitions are not represented in our formulation (and thus not annotated), i.e. at the end of a folding manipulation, when the gripper releases the garment from a certain height, the state is automatically labelled as **sideways folded** even though the corners are still on the air and falling. Finally, it is also has to be taken into account that the garments we generated at this stage are rather simple since they are rectangular and noise free.

5 CONCLUSIONS

In this paper we presented a novel approach to solve the semantic state estimation problem for robotic cloth manipulation in an end-to-end manner. To achieve it we adopted a grasping manipulation framework and was used as a baseline for automatic data labelling which were generated by our MuJoCo simulator while performing complex manipulation actions on a textile uni-manually and bi-manually. Then we fed those data in pretrained neural networks on Imagnet and fine-tuned them.

The results we provided showed that indeed semantic state estimation is possible to be investigated by data driven methods such as deep learning in combination with a semantic classification that simplifies and groups the high number of complex states

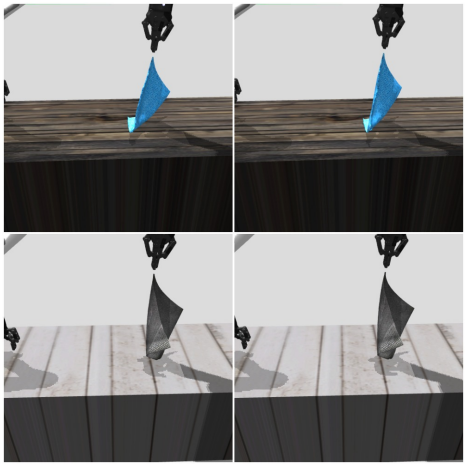

Fig. 4. Conflicted cases due to highly similarity between two states. The states on the left column are **Flat semi-lifted with one gripper** and are predicted correctly while the minimal changes in the next states that are occur in comparison to the right column images where not enough to change the network's prediction. The actual states on the right column are **Crumpled semi-lifted with one gripper** and **Lifted with one gripper**.

that a garment can be. We managed to achieve over 95% accuracy in all three models we have used for 10 different states with only the use of RGB data. The results are very promising and induce to believe that the method can be valid also for real images. The challenge in this case will be to obtain enough data to train the network that needs to be obtained either by manual labelling or other means.

In the future we aim to improve the simulator by introducing a higher variety of deformable shapes and attempt to solve the problem of state estimation with smaller datasets and investigate what techniques can be used when the amount of data that are

Table 3. Data samples per class

Classes	Dataset	Train	Val
Flat	21541	17233	4308
Flat semi-lifted with one gripper	20606	16485	4121
Sideways Folded	13387	10710	2677
Lifted with two grippers	7484	5987	1497
Diagonally Folded	6851	5481	1370
Flat semi-lifted with one gripper	6454	5163	1291
Middle grasping with two grippers	6214	4971	1243
Crumpled	3636	2909	727
Crumpled semi-lifted with one gripper	2717	2173	544
Lifted with one gripper	2490	1992	498

Data samples per class for the whole dataset and an 80-20 training-validation split that used for our experiments.

available are limited, like in the real world. Furthermore, we will generate a new dataset which will include more grasps and more complex manipulations, improve our annotation framework for more ambiguous states and generate data from multiple angles with limited class imbalance. We believe and hope that our simulator, our data and most importantly the method we used to approach the problem will be adopted by the cloth manipulation community and trigger further contributions in deep learning and data generation for textile investigation.

Acknowledgements

This work has been partially funded by MCIN/ AEI /10.13039/ 501100011033, Spain, under the project CHLOE-GRAPH (PID2020- 119244GB-I00); by the European Union's Horizon 2020 under ERC Advanced Grant CLOTHILDE (no. 741930); and by MCIN/ AEI /10.13039/501100011033, Spain, and by the "European Union NextGenerationEU/ PRTR, Spain, under the project COHERENT (PCI2020-120718-2);

References

1. A. Byravan and D. Fox, "Se3-nets: Learning rigid body motion using deep neural networks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 173–180.
2. I. Mariolis, G. Peleka, A. Kargakos, and S. Malassiotis, "Pose and category recognition of highly deformable objects using deep learning," in *2015 International conference on advanced robotics (ICAR)*. IEEE, 2015, pp. 655–662.
3. J. Borràs, G. Alenyà, and C. Torras, "A grasping-centered analysis for cloth manipulation," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 924–936, 2020.
4. I. Garcia-Camacho, J. Borràs, and G. Alenyà, "Knowledge representation to enable high-level planning in cloth manipulation tasks," in *ICAPS Workshop on Knowledge Engineering for Planning and Scheduling*, 2022.
5. M. Lippi, P. Poklukar, M. C. Welle, A. Varava, H. Yin, A. Marino, and D. Kragic, "Latent space roadmap for visual action planning of deformable and rigid object manipulation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5619–5626.
6. 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.
7. X. Lin, Y. Wang, J. Olkin, and D. Held, "Softgym: Benchmarking deep reinforcement learning for deformable object manipulation," *arXiv preprint arXiv:2011.07215*, 2020.
8. E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.
9. R. Hoque, D. Seita, A. Balakrishna, A. Ganapathi, A. K. Tanwani, N. Jamali, K. Yamane, S. Iba, and K. Goldberg, "Visuospatial foresight for multi-step, multi-task fabric manipulation," *arXiv preprint arXiv:2003.09044*, 2020.
10. D. Seita, N. Jamali, 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," *arXiv preprint arXiv:1809.09810*, 2018.

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. 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.
13. P.-C. Yang, K. Sasaki, K. Suzuki, K. Kase, S. Sugano, and T. Ogata, "Repeatable folding task by humanoid robot worker using deep learning," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 397–403, 2016.
14. D. Tanaka, S. Arnold, and K. Yamazaki, "Emd net: An encode–manipulate–decode network for cloth manipulation," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1771–1778, 2018.
15. R. Zoliner, M. Pardowitz, S. Knoop, and R. Dillmann, "Towards cognitive robots: Building hierarchical task representations of manipulations from human demonstration," in *Proceedings of the 2005 IEEE International Conference On Robotics and Automation*. IEEE, 2005, pp. 1535–1540.
16. F. Wörgötter, E. E. Aksoy, N. Krüger, J. Piater, A. Ude, and M. Tamosiunaite, "A simple ontology of manipulation actions based on hand-object relations," *IEEE Transactions on Autonomous Mental Development*, vol. 5, no. 2, pp. 117–134, 2013.
17. E. E. Aksoy, A. Abramov, J. Dörr, K. Ning, B. Dellen, and F. Wörgötter, "Learning the semantics of object–action relations by observation," *The International Journal of Robotics Research*, vol. 30, no. 10, pp. 1229–1249, 2011.
18. M. J. Aein, E. E. Aksoy, and F. Wörgötter, "Library of actions: Implementing a generic robot execution framework by using manipulation action semantics," *The International Journal of Robotics Research*, vol. 38, no. 8, pp. 910–934, 2019.
19. E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," 2016.
20. A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani *et al.*, "Transporter networks: Rearranging the visual world for robotic manipulation," *arXiv preprint arXiv:2010.14406*, 2020.
21. D. Seita, P. Florence, J. Tompson, E. Coumans, V. Sindhwani, K. Goldberg, and A. Zeng, "Learning to rearrange deformable cables, fabrics, and bags with goal-conditioned transporter networks," *arXiv preprint arXiv:2012.03385*, 2020.
22. Z. Erickson, V. Gangaram, A. Kapusta, C. K. Liu, and C. C. Kemp, "Assistive gym: A physics simulation framework for assistive robotics," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 10 169–10 176.
23. F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik *et al.*, "Sofa: A multi-model framework for interactive physical simulation," in *Soft tissue biomechanical modeling for computer assisted surgery*. Springer, 2012, pp. 283–321.
24. F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang *et al.*, "Sapien: A simulated part-based interactive environment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 097–11 107.
25. C. Gan, J. Schwartz, S. Alter, M. Schrimpf, J. Traer, J. De Freitas, J. Kubilius, A. Bhandwaldar, N. Haber, M. Sano *et al.*, "Threedworld: A platform for interactive multi-modal physical simulation," *arXiv preprint arXiv:2007.04954*, 2020.
26. B. Willimon, S. Hickson, I. Walker, and S. Birchfield, "An energy minimization approach to 3d non-rigid deformable surface estimation using rgbd data," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 2711–2717.
27. G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

28. J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.
29. 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.
30. S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.
31. M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6105–6114.
32. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.