2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Learning active manipulation to target shapes with model-free, long-horizon deep reinforcement learning

Matias Sivertsvik,[1] Kirill Sumskiy,[1] and Ekrem Misimi[2]

***Abstract*— We investigate the active manipulation of objects using model-free and long-horizon DRL (Deep Reinforcement Learning) to achieve target shapes. Our proposed approach uses visual observations consisting of segmented images, to mitigate the sim-to-real gap. We address a long-horizon manipulation task requiring a sequence of accurate actions to achieve the target shapes using a robot arm with an RGB-D camera in eye-in-hand configuration, and an elongated, volumetric, elastoplastic object. We find similar objects in food, marine, and manufacturing domains. The aim is to actively manipulate the object into an arbitrary target shape using image observations. We trained a DRL agent using PPO (Proximal Policy Optimization) by running 768 parallel actors in simulation, for a total of 1,2M environment interactions, and tested this on 200 unseen target deformations. In three attempts, 82% of the trials achieved a greater than 90% overlap with the 200 target shapes. By relying on segmentation images as a visual observation space, we successfully transferred the agent to the real world without supplementary training. Our approach does not need any real-world manipulation examples nor fine-tuning in the real world. The robustness of our approach was demonstrated in simulation, and experimentally validated in the real world for specific manipulation tasks, achieving a 94.2% mean zero-shot overlap success rate on previously unseen target shapes.**

I. Introduction

Research into robotic manipulation has primarily focused on rigid objects because their dynamics are easier to model [1]–[3]. However, the assumption of rigidity, i.e., that objects do not deform upon force interaction, does not hold in many real-world applications. For example, many food items are compliant and robots must handle them gently to preserve product integrity [4]. Modelling complexities entail that the robotic manipulation of deformable objects is an open research problem [5]–[7]. Recent work has shown that machine learning techniques offer a valuable complement to modelling when performing robotic manipulation [2], [8], [9]. Moreover, machine learning methods are capable of achieving generalisation between tasks, such as learning to grasp previously unseen objects [9]–[11].

The ability to consider the long-term consequences of actions is foundational for autonomous robotic manipulators capable of generalising to a wide variety of tasks. Reinforcement learning (RL) is a machine learning paradigm for learning sequential decision-making to reach long-term goals. By combining RL with neural networks and image processing techniques from deep learning, it has been possible to train agents to play a variety of video games directly from pixels [12], [13]. DRL has also been applied successfully to vision-based robotic manipulation tasks [8], [11], [14]. One of the key concepts behind our work is that model-free DRL may enable an agent to learn to interact meaningfully with objects directly from visual observations, thus circumventing the challenges raised in modelling the dynamics of deformable objects, and instead learning manipulation tasks in an end-to-end manner [15]–[17].

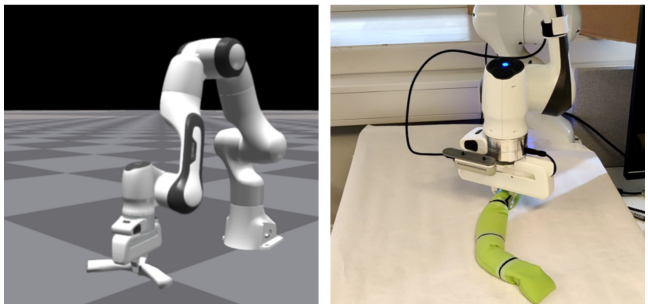


Fig. 1. Franka Panda Emika robot actively deforming the object to target shape. Left - in simulation; right - in the real world. The policy is successfully transferred to the real world, without supplementary training.

Model-free approaches to RL offer the advantage of generality and reuse in multiple scenarios with few modifications since agents can easily be re-trained for new tasks. The drawback is the volume of training data required to learn successful DRL policies [18]. This is especially true in environments with continuous action spaces [19], [20]. Recent work shows that action spaces based on continuous DRL-control of robotic manipulators may fail to converge for tasks with long horizons [19]. Long-horizon tasks, such as manipulation to target shapes in our work, pose challenges because DRL agents learn policies by optimizing rewards over sequences of actions. Such sequences can become quite long in tasks where actions have delayed consequences, making these tasks difficult to solve [21]. Since robotic manipulation is often a multi-step process, the need for alternate action spaces to reduce the overall training and to better apply model-free RL to long-horizon tasks has also been identified in [22]. While some recent progress has been made, studies on the manipulation of deformable objects have largely been limited to the use of one [15] and two-dimensional [17][23] objects, or the use of a suction-cup as a gripper [24] [25]. Despite some recent work in manipulation with parallel-jaw grippers of deformable 3D objects such as a sponge-like elastic object [26], or elastoplastic objects [27], the area of manipulation of 3D objects, in general, and long-horizon task manipulation, in particular, is under-researched.

One study [15] has proposed a pixel-based pick-and-place action for performing rope-straightening and cloth-flattening

[1]NTNU IDI, Norway; [2]SINTEF Ocean, Norway

tasks. The DRL agent uses input images from the camera to learn a placing policy to output pixel locations for placing the grasped object. The robot uses pre-computed motion primitives to grasp and place the object at the corresponding real-world locations of the picking pixel and placing pixel respectively. Training in simulation with domain randomization enabled the agent to achieve a goal intersection coverages of 48% and 84% for the rope-straightening and cloth-flattening tasks, respectively. A similar study based on learned graph dynamics model over a set of keypoints [28], addresses the rope straightening and cloth manipulation task. For the rope straightening, they achieved a total mean success rate of 79.3% in simulation and 62.72% in the real world. Other studies [24], [25] have shown that conditional pick-and-place policy can be learned to perform a wide variety of manipulation tasks, involving deformable objects such as ropes and bag structures. A further study [17] proposed an alternative action space for policy learning for a fabric-folding task, by estimating a pick action and selecting a pre-defined place action of fold angles, achieving an overall accuracy of 78%. The policy was trained on fixed offline data and required one hour of experience gathering in the real world prior to validation.

The manipulation of 3D objects has stricter requirements for the action space. For parallel jaw grippers it is not sufficient simply to output a grasping pixel—we must also consider the gripper orientation. Aforementioned studies [24], [25] circumvent this by using a suction-cup end effector, but these are not interchangeable with parallel jaw grippers. Some recent studies on the manipulation of 3D objects, using parallel jaw gripper, are promising. In [26] is presented an approach based on point clouds for 3D shape-servoing of an elastic object that returns, enabling a robot to manipulate the 3D shape to the desired shape by selecting one manipulating point on the object. Despite its merit, this method does not scale to the elastoplastic objects which do not return and which require manipulation on several points over a long horizon, to achieve the desired shape. In [27] is presented an approach that learns a particle-based dynamics model using graph neural networks to teach a robot which uses a parallel two-finger gripper to shape an 'X' conditioned to several target shapes. While showing promise and although the approach is trained in simulation, it also relies on experience gathering by real-world interactions to learn the policy, and it considers only short-horizon tasks.

Although alternative action spaces may improve sample efficiency and reduce the amount of training data, it is common to supplement the training by using synthetic training data from simulators. Simulators are cost-efficient; they accelerate data acquisition process, mitigate safety linked to the operation of real robots, and reduce overall robot wear and tear [29]. Some studies [30], [10], and [31] have succesfully demonstrated the viability of this approach by successfully training a grasping task in simulation and transferring the policy to the real world. However, they also recognise the need for specific measures to overcome the sim-to-real gap.

Our work addresses the problem of applying model-free reinforcement learning to a long-horizon, robotic manipulation task requiring a sequence of accurate actions to achieve the goal. We formulate a task in which an elongated, elastoplastic 3D object, that does not return, is manipulated to achieve a target shape in the image plane using a robot with a parallel jaw gripper. We find similar elongated, elastoplastic objects in the domains of food, marine and manufacturing industries. Our aim is to obtain generalisability in the range of shapes the developed method is capable of achieving in a single object type, rather than in the diversity of objects it is capable of handling.

We present and demonstrate an approach (Fig.1) suitable for long-horizon learning, requiring multiple, sequential and accurate actions for manipulation to target shapes of deformable objects, using a parallel jaw gripper and eye-in-hand camera. Our approach a) does not need real-world manipulation examples and is trained entirely in simulation; b) the robot learns directly from pixels of segmented images; c) successfully transfers to the real world with only minimal domain randomisation and no supplementary training; d) addresses some issues of generating valid grasp orientations for a discretized action space, using a parallel jaw gripper and eye-in-hand camera; and e) achieves a 94.2% mean zero-shot overlap success rate for previously unseen target shapes.

II. Methodology

A. Task Description

We formulate the task of deforming 3D objects to an arbitrary shape as a goal-conditioned, finite, partially observable Markov decision process (MDP) defined by the tuple ($\mathcal{S}$, $\mathcal{A}$, $\mathcal{G}$, $\mathcal{O}$, P, r, γ). $\mathcal{G} \in \mathcal{S}$ is a set of goal states upon which the reward given to the agent is based. We also define an observation space $\mathcal{O}$ and introduce the notion of an observation o - a partial representation of an underlying state s. Using DRL, the aim is to find a goal-conditioned policy $\pi(a|o, g)$ that maximises the expected sum of rewards of this MDP for any given goal g.

Conceptually, the environment consists of five main components: A robot manipulator, an end effector, an elastoplastic deformable object, an eye-in-hand camera, and a work surface on which the object is placed, implemented both in simulation and real world (Fig. 1). We implemented the simulated environment in NVIDIA Isaac Gym [32], a simulator for RL that utilises the parallelisation abilities of GPUs to simulate large amounts of environments at the same time. Isaac Gym includes the NVIDIA FleX physics engine, which supports the simulation of deformable objects.

The cameras in Isaac Gym can capture standard RGB images, depth images, and a direct pixel-level segmentation image of objects in the scene (Fig. 2). The version of the FleX physics engine in Isaac Gym does not support colouring of objects, nor adding specific textures to the environment, thus limiting the possibilities for visual domain randomisation as implemented in [16], [30]. Therefore, we opted to constrain the observation space to only segmentation images. The simplicity of binary images limits possible differences between observations from the simulation and

the real world. Indeed, due to their similarities in simulation (Fig. 2c) and real world (Fig. 5), we used segmented images to mitigate the sim-to-real gap, while still remaining true to the goal of achieving RL from pixels.

(a) RGB (b) Depth (c) Segmentation

Fig. 2. Images from the camera sensor in Isaac Gym. Note that the segmented image shows only the object without other visual distractions.

A goal-conditioned RL policy, $\pi(o_t, g_t)$, requires a representation of the goal, g_t, as an additional input to the policy. Our proposed observation space consists of a stack of three 96×96 segmentation images of the current observation, the previous observation, and the target deformation (similar to the framestacking in [12]), and a one-hot encoded vector of the previous action. Training episodes were always initialised with a random action, ensuring that there always exists a previous observation. Fig. 3 shows an example of the full observation space.

Fig. 3. Observations for the DRL agent. Left to right: the current observation, the previous observation, and the target deformation. The one-hot encoded vector describes the action taken in the previous timestep.

The viability of an action space is highly contingent on choices regarding the experimental set-up and the selected observation space. Ideally, the selected action space balances the expressiveness needed to perform arbitrary deformations, while also being both learnable for an RL agent and transferrable to the real world. 3D objects put greater demands on the positioning and orientation of the gripper. Therefore, existing pixel-wise action spaces need to be augmented to meet the extra demands imposed by these objects.

Our proposed action space is motivated from [17], [22], [33], where robot motions for manipulation actions are precomputed. Rather than sampling actions over all pixels as in [17] and [33], the proposed action space further discretises the object into four equally-spaced regions and performs one of four deformation actions on the region selected by the agent. The grasp occurs when the gripper is positioned over the centroid of one of the four regions and oriented so that the normal vector of the gripper is approximately aligned with the longitudinal axis of the region. Compared to pixel-level spaces, this simplification was made due to the difficulties in obtaining object orientations from pixels and the fact that misplaced grasp attempts can compromise the object's integrity. A discretisation of the action space by reducing the possible action locations to 4 regions aims to make the environment easier to learn [34]. After approaching one of the four grasping points, the object is grasped, and the region is slightly lifted. Depending on the agent's selected action, the gripper performs a small, translational movement of 3.5 cm in one of four possible directions, creating a small deformation before the object is released.

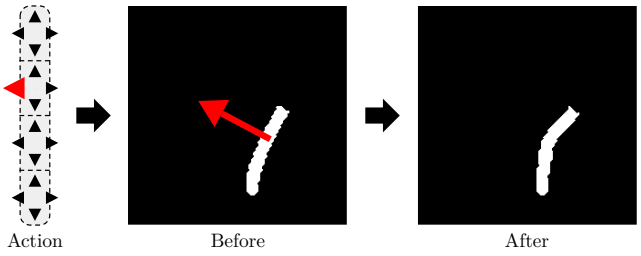


Fig. 4. Left: Schematic representation of the four regions of the body and action's deformation directions. Middle: Segmentation mask before applying the action. Right: Segmentation mask after applying the action.

Fig. 4 shows conceptually how the grasping points are divided along the object and marking the four possible directions of translational movement: two longitudinal actions and two lateral actions which are local to that region. It also shows the resulting deformation in the agent's view (red line added for emphasis) when applying the action marked in red.

We base the success criteria for the deformation task on the degree of pixel overlap between the segmented object in the target image and the current observation - the greater the pixel overlap, the more closely the shapes match. Requiring a full 100% overlap for task completion was found to be too restrictive due to the coarseness inherent to the selected action space and pixel resolution of the observation space. Empirically, we found that padding the target deformation by an extra pixel helped overcome the coarse resolution, and a threshold of >95% overlap was found to produce clearly matching shapes while still being consistently achievable by the agent. Fig. 8 shows examples of successful observations.

We derive the reward function directly from the degree of overlap, with stepsize increases in reward for greater overlaps. The reward at timestep t was defined as

$$r_t = \begin{cases} -0.1, & \text{when overlap = 0\%,} \\ -0.01, & \text{when 0\% < overlap < 50\%,} \\ 0.01 \times \text{overlap}, & \text{when 50\% } \leq \text{ overlap < 75\%,} \\ 0.1 \times \text{overlap}, & \text{when 75\% } \leq \text{ overlap < 95\%,} \\ 1, & \text{if overlap > 95\%.} \end{cases} \tag{1}$$

B. Experimental Setup

We developed two parallel environments conforming to the task description: an RL training environment in Isaac Gym and a matching testing environment in the real world. Both environments include a Franka Emika Panda robot with

a parallel jaw gripper (Fig. 1), and we used a D435 RGB-D camera mounted on the end-effector of the real robot, and internal camera sensors in the simulator. An elastoplastic deformable, cylindrical object of 40 cm length and a diameter of 4 cm, resulting in same width and height of 4 cm, was created in Blender [35] and imported to Isaac Gym. We constructed a real-world 3D-equivalent by filling a sleeve made from an elastic fabric with rice and closing both ends.

C. Training Details

The agent was trained using Proximal Policy Optimisation (PPO) [36]. The observation was first passed through a feature extractor consisting of three convolutional layers shared by both the actor and the critic. The output from the feature extractor was flattened and concatenated with the previous action (one-hot encoded) before being passed through a hidden layer. The output from this layer splits into a value head outputting a single value and a policy head outputting logits for the 16 possible actions. This is based on the hypothesis that the same features extracted from the pixels are useful for both learning a value function and a policy. Additionally, we extended the PPO algorithm with Phasic Policy Gradient (PPG) [37], which divides the training using gradient descent into a policy phase (which functions as regular PPO) and an intermittent auxiliary phase.

During the policy phase, we used the clipped PPO objective with a small entropy bonus. For every timestep t, the loss is defined as

$$L = L_a + L_c - 0.02 * S[\pi_\theta](s_t), \tag{2}$$

where L_a is the actor loss, L_c is the critic loss and $S[\pi_\theta](s_t)$ is the entropy of the policy (see [36] for specifics on the PPO objective). During the auxiliary phase, we update the value head of the network with a joint loss function combining the critic loss L_c with a behavioural cloning loss:

$$L_{\text{joint}} = L_c + \mathbb{E}_t \left[KL[\pi_{\theta_{\text{old}}}(\cdot|s_t), \pi_\theta(\cdot|s_t)] \right]. \tag{3}$$

Training of the agent was performed in 768 environments to take advantage of the parallelisation benefits of Isaac Gym. We trained the agent for 200 training iterations, corresponding to 1 254 400 environment interactions, in total corresponding to ∼48 hours on an NVIDIA GeForce RTX 3090 GPU. The number of environments and training iterations were selected due to the GPU memory and the available hardware constraints and to limit the overall training time. The training runs were repeated for a total of 7 times. Per training iteration, the PPO agent collected 8 state transitions from each of the 768 environments, for a total of 6 272 observations. The parameters for the actor network were updated using batch gradient descent with a mini-batch of 384 transitions on the 6 272 collected observations without sample reuse. Every 16 training iterations, the critic parameters were updated according to the auxiliary PPG loss defined in Eq. 3, using a mini-batch size of 384, where each sample was reused 8 times, according to the range proposed by [37].

We collected a total of 2200 segmentation images as target deformations by letting a random policy act on the object in the environment and then capture the object's current state after an interaction. Of these 2200 images, 2000 were used as training goals for the DRL agent, while 200 were withheld as a test set, unseen by the policy. We started each training episode by sampling a target deformation randomly from the training set for each environment. A random action was performed at the start of every training episode. A training episode lasted until the agent either achieved greater than 95% overlap with the target deformation or performed 60 interactions in the environment. Slight domain randomisation of the dynamics was achieved by adding random perturbations of up to 1 cm to the translational movement.

D. Visual Control

Two main challenges needed to be solved to transfer the agent to the real world, namely how to segment the object and how to divide the object into regions to comply with the action space used in the simulator. To consistently segment the sleeve, we used colour-based segmentation. We divided the sleeve into four regions using a black marker, with the resulting segmentation shown in the left image in Fig. 5. By smoothing and applying a morphological close operation, we were closing the gaps in the segmentation. After cropping and resizing, the processed image resembles those seen by the agent in the simulation (middle image, Fig. 5).

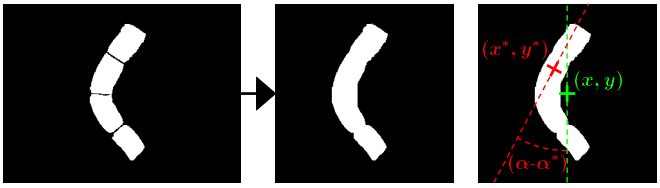


Fig. 5. Left: a segmentation image from the real world before processing. Middle: after processing. Right: Vision control law.

Principal component analysis (PCA) was used to find the principal components of the set of pixels belonging to a given region. The first two principal components define a new orthogonal coordinate system centred in the region mean. The principal components and the centroid for each region are marked with arrows and purple dot (Fig. 6).

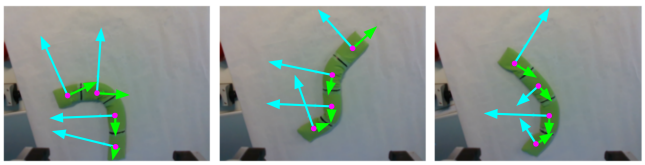


Fig. 6. The outcome of the principal component analysis for three different configurations of the test deformable object.

Our vision control law, illustrated in (Fig. 5), is similar to [10]. We define the current coordinates of the midpoint of the gripper to be (x, y) and the current angle of the gripper around the z-axis of the camera to be α. Correspondingly, we define the position of the first principal component of the region to grasp to be (x^*, y^*) with angle α^*, constituting the desired pose of the gripper to achieve. To perform the visual

control, the lateral translational components t_x and t_y and the rotational component θ around the z-axis are given by:

$$t_x = Z(x - x^*), t_y = Z(y - y^*), \theta = -(\alpha - \alpha^*) \quad (4)$$

where Z is the distance in the z-axis from the camera to the object. Since we have an eye-in-hand configuration, we hold Z constant between interactions, by returning to the same initial position after performing an action. By using a classical position-based visual control [38], the control output $\mathbf{v} = (v_x, v_y, \omega_z)$ has the simple form:

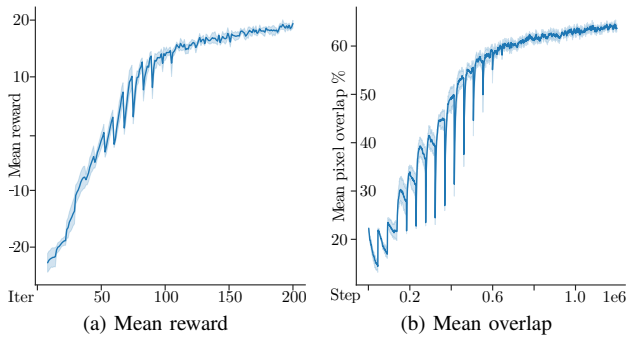
$$\mathbf{v} = -\lambda \begin{pmatrix} \cos\theta t_x + \sin\theta t_y \\ -\sin\theta t_x + \cos\theta t_y \\ \theta \end{pmatrix} \quad (5)$$

where λ is a positive gain. Using $\mathbf{v}$, we can position and perform a top-down grasp of a given region using ViSP [39].

III. Results and Discussion

A. Training Performance

Fig. 7 shows the mean reward and mean pixel overlap obtained by running 768 parallel actors over 200 PPO training iterations (∼1.2 million training steps), averaged over 7 training runs, over slightly more than 48 hours of wall time. The clear upwards trend in achieved rewards as training progresses, shows that the agent is able to learn across all runs. The mean pixel overlap achieved across the parallel actors was a good indicator of the success rate. The periodic drops in mean overlap are due to the resets of the environments every 60 interactions as training episodes finish. In the early phase of training, few of the actors are able to successfully complete the target deformation. This leads to a synchronous wave of resets of all unfinished environments about every 60 timesteps, visible as drops around every 7.5 PPO training iterations (Fig. 7b). Over time, as more actors are able to achieve their target deformations, the resets become more asynchronous, reducing the amplitude of the periodic drops and leading to a stable mean overlap around 65 %. This indicates that the agent successfully learns to solve the tasks throughout training.



(a) Mean reward (b) Mean overlap

Fig. 7. Training performance over all environments in mean reward per training iteration(left) and mean pixel overlap per environment step(right).

Both performance metrics show low variance across the multiple run, visualised by the tight bounds of the shaded regions, representing the 95% confidence interval of the mean performance across 7 runs. Overall, this suggests stability of performance and performance repeatability.

B. Simulation Results

Generalisation of the learned policy to arbitrary shape deformations was tested on an additional set of 200 previously unseen target shapes. The learned policy attempted to achieve each of the 200 test shapes over the same episode length of 60 steps, as during training, for a total of three trials. Table I shows the different overlap levels achieved for each trial, as well as the total when combining all trials. Of 600 attempted deformations, 401 achieved over 95 % overlap and only 40 achieved less than or equal to 80 % overlap. A natural question to ask is why a considerable amount of the good attempts fall below 95%. We observed that the agent seemed to mostly struggle with the examples requiring very long translations, suggesting that the choice of action space may influence the range of deformations the agent is able to achieve. The advantage of our proposed translation movement is that it allows for more accurate positioning once the agent has approximated the target shape. The drawback is that one also has to use the same movement distance when a longer translation might be more efficient. Moving the object across the work surface thus requires more multiple actions, extending the temporal sequence the agent needs to reason over, and increasing the task complexity for the RL agent.

TABLE I

Overlaps achieved on 200 unseen deformations

Overlap levels	Trial 1	Trial 2	Trial 3	Total
$>$ 95 %	140	128	133	401
90-95 %	28	31	32	91
80-90 %	21	28	19	68
$\leq$ 80 %	11	13	16	40

Three examples of trials surpassing the 95% overlap are shown in Fig. 8. By visual comparing a target deformation (left) with the achieved deformation (right) for each trial, one clearly sees the similarities between the images. This points to one of the key strengths of using segmentation images and the proposed reward, in that they make it easy to discern whether the target is achieved.

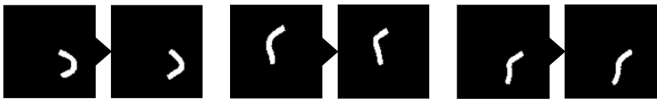


Fig. 8. Three examples of task deformations with $>$ 95% overlap. The target is to the left and the achieved deformation is to the right of each pair.

C. Real-world Results

When transferring the agent to the real world, the agent must cope with a shift in the dynamics of the environment. Additionally, the images captured by the eye-in-hand-mounted camera do not completely match the observations gathered in simulation. For real-world validation we chose five manipulation tasks to previously unseen shapes. Three trials were performed for each of the five tasks. Table II shows the maximally achieved overlaps in each trial, as well the average overlap over all three trials. The trained policy surpassed the 95% overlap threshold in nine out of 15 trials, resulting in a 94.2% total mean zero-shot success rate.

TABLE II

ACHIEVED PIXEL OVERLAPS FOR FIVE TASKS IN THE REAL WORLD.

target	Trial 1 [%]	Trial 2 [%]	Trial 3 [%]	Average [%]
Right kink	**100**	**98.5**	**100**	**99.5**
Right "S"	**95.4**	**98.1**	**96.9**	**96.8**
Left "L"	78.9	**96.9**	93.9	89.9
CCW Rot	85.1	88.8	91.2	88.4
Left "C"	**99.5**	93.6	**97.6**	**96.9**

These results underline the main benefit of the proposed action space, namely that it successfully overcomes the differences in dynamics between the simulation and the real world. This is a key problem [16] when transferring a manipulation policy to the real world. The progression in Fig. 9 shows an example on how the suggested action space is used by the agent to quickly converge on the intended target. Fig. 9 shows the progression of images from the initial position to 99.5% overlap for the *Left "C"*-shape in trial 1. The target image is shown in the green box, while the segmentation corresponding to the final RGB image in the progression is next to it. The agent is sophisticated in how it approaches the task shown in the progression, by first creating a kink in the object by deforming one of the middle regions of the object. The agent then leverages this kink to further deform the object by moving the outer regions until it converges to the desired shape. This suggests that the agent learns manipulation strategies that are applicable both in simulation and in real world, despite the discrepancy between the training and evaluation environments, given the object is more elastic in simulation and more elastoplastic in the real world. Comparatively, our average results from Table II show significantly higher overlaps compared to the results in [15] and [28], who report a mean overlap of 48% and 62.72% respectively, for the case of the rope straightening.

To better take into the account the orientation and shape of the object, we defined an action space that was more likely to successfully grasp the objects by dividing the object into regions. Though this contrasts with earlier work using pixel-wise action spaces, one still retains the core idea of abstracting low-level control of the robot away from the agent. The results suggest that opting for high-level actions, where the details of robot control are abstracted away, is beneficial also for the manipulation tasks in our work, extending the findings from previous work [17], [25], [33].

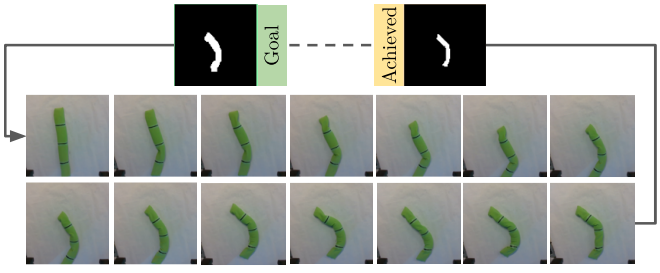


Fig. 9. Progress of an object manipulation from initial to a goal state.

Fig. 10 shows the active manipulation for each of the five tasks. In all cases, the achieved shape closely matches the desired. Despite not fully clearing the *Counterclockwise rotation* task (fourth image from the left), one can see that the agent has achieved the overall rotation required, but the kink in the achieved shape is angled in the opposite direction from the kink in the target. This slight difference results in an achieved overlap of 91.2%. Though the deformation tasks required less translatory movements compared to those in simulation, they still required the agent to generalise to yet another set of previously unseen shape deformations, while coping with the different dynamics of the real world.

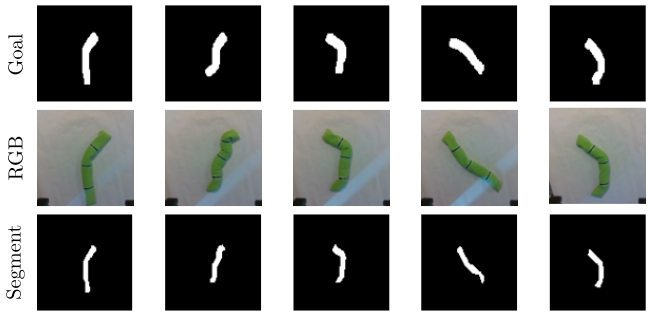


Fig. 10. Five deformation tasks. Top row - target deformations; middle - RGB images of achieved states; bottom - their segmentation masks.

Results show that by using segmentation images as observations, it is possible to transfer an agent from simulation to the real world without added visual domain randomisation or domain adaptation. This contrasts with previous findings [10], [16], [30], where heavy augmentation was required to successfully transfer an RL agent to the real world.

IV. CONCLUSION

In this work, we have presented an approach for active manipulation of elongated, elastopastic objects, suitable for model-free, long-horizon learning, requiring multiple, sequential and accurate actions, over a long horizon, for manipulation to target shapes, using a parallel jaw gripper and eye-in-hand configuration. We propose an approach that: a) does not need real-world manipulation examples, and is trained entirely in simulation; b) the robot learns directly from pixels of segmented images; c) successfully transfers to the real world with only minimal domain randomisation and no supplementary training; d) addresses some issues of generating valid grasp orientations for a discretized action space, using a parallel jaw gripper and eye-in-hand camera; and e) achieves a 94.2% mean zero-shot overlap success rate for previously unseen target shapes. Results suggest that DRL is useful for long-horizon active manipulation tasks to target shapes. A natural extension of this work is to expand the range of objects from the relevant real-world domains, such as the food and the marine domains, for which the approach can be applied, both in terms of shape diversity and elasticity of the objects. In future work, we intend also to augment the action space by generating actions involving non-prehensile actions.

ACKNOWLEDGEMENTS

The work is supported by GentleMAN (299757) and BIFROST (313870) projects (RCN Norway).

REFERENCES

[1] S. F. Gibson and B. Mirtich, "A survey of deformable modeling in computer graphics," Mitsubishi Electric Research Laboratories, Tech. Rep., 1997.

[2] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis - A survey," *IEEE Transactions on robotics*, vol. 30, no. 2, pp. 289–309, 2014.

[3] V. E. Arriola-Rios, P. Guler, F. Ficuciello, D. Kragic, B. Siciliano, and J. L. Wyatt, "Modeling of deformable objects for robotic manipulation: A tutorial and review," *Frontiers in Robotics and AI*, vol. 7, p. 82, 2020.

[4] E. Misimi, A. Olofsson, A. Eilertsen, E. R. Øye, and J. R. Mathiassen, "Robotic handling of compliant food objects by robust learning from demonstration," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*. IEEE, 2018, pp. 6972–6979.

[5] J. Zhu, A. Cherubini, C. Dune, D. Navarro-Alarcon, F. Alambeigi, D. Berenson, F. Ficuciello, K. Harada, J. Kober, X. Li, J. Pan, W. Yuan, and M. Gienger, "Challenges and outlook in robotic manipulation of deformable objects," *IEEE Robotics and Automation Magazine*, pp. 2–12, 2022.

[6] E. Misimi, E. R. Øye, Øystein Sture, and J. R. Mathiassen, "Robust classification approach for segmentation of blood defects in cod fillets based on deep convolutional neural networks and support vector machines and calculation of gripper vectors for robotic processing," *Computers and Electronics in Agriculture*, vol. 139, pp. 138–152, 2017.

[7] E. Bar, J. R. Mathiassen, A. Eilertsen, T. Mugaas, E. Misimi, Å. S. Linnerud, C. Salomonsen, and H. Westavik, "Towards robotic post-trimming of salmon fillets," *Industrial Robot: An International Journal*, vol. 43, no. 4, pp. 421–428, 2016.

[8] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *Journal of Machine Learning Research*, vol. 17, no. 39, pp. 1–40, 2016.

[9] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008.

[10] O.-M. Pedersen, E. Misimi, and F. Chaumette, "Grasping unknown objects by coupling deep reinforcement learning, generative adversarial networks, and visual servoing," in *IEEE International Conference on Robotics and Automation*. Paris, France: IEEE, 2020, pp. 5655–5662.

[11] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, and S. Levine, "Scalable deep reinforcement learning for vision-based robotic manipulation," in *Proceedings of The 2nd Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Billard, A. Dragan, J. Peters, and J. Morimoto, Eds., vol. 87. PMLR, 2018, pp. 651–673.

[12] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller. (2013) Playing atari with deep reinforcement learning. [Online]. Available: http://arxiv.org/abs/1312.5602

[13] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[14] OpenAI, M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba. (2018) Learning dexterous in-hand manipulation. [Online]. Available: https://doi.org/10.48550/arXiv.1808.00177

[15] Y. Wu, W. Yan, T. Kurutach, L. Pinto, and P. Abbeel. (2019) Learning to manipulate deformable objects without demonstrations. [Online]. Available: https://doi.org/10.48550/arXiv.1910.13439

[16] 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.

[17] R. Lee, D. Ward, A. Cosgun, V. Dasagi, P. Corke, and J. Leitner, "Learning arbitrary-goal fabric folding with one hour of real robot experience," in *Proceedings of the 2020 Conference on Robot Learning*, J. Kober, F. Ramos, and C. Tomlin, Eds., vol. 155. PMLR, 2021, pp. 2317–2327.

[18] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. (2015) Continuous control with deep reinforcement learning. [Online]. Available: https://doi.org/10.48550/arXiv.1509.02971

[19] I. Popov, N. Heess, T. Lillicrap, R. Hafner, G. Barth-Maron, M. Vecerik, T. Lampe, Y. Tassa, T. Erez, and M. Riedmiller. (2017) Data-efficient deep reinforcement learning for dexterous manipulation. [Online]. Available: https://doi.org/10.48550/arXiv.1704.03073

[20] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 2016, pp. 1928–1937.

[21] A. C. Li, P. Vaezipoor, R. T. Icarte, and S. A. McIlraith, "Challenges to solving combinatorially hard long-horizon deep rl tasks," 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2206.01812

[22] A. Cherubini, V. Ortenzi, A. Cosgun, R. Lee, and P. Corke, "Model-free vision-based shaping of deformable plastic materials," *The International Journal of Robotics Research*, vol. 39, no. 14, pp. 1739–1759, 2020.

[23] M. Aranda, J. A. C. Ramon, Y. Mezouar, A. Bartoli, and E. Ozgur, "Monocular visual shape tracking and servoing for isometrically deforming objects," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, United States, 2020, pp. 7542–7549.

[24] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, and J. Lee, "Transporter networks: Rearranging the visual world for robotic manipulation," in *Proceedings of the 2020 Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Kober, F. Ramos, and C. Tomlin, Eds., vol. 155. PMLR, 2021, pp. 726–747.

[25] 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," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4568–4575.

[26] B. Thach, B. Y. Cho, A. Kuntz, and T. Hermans, "Learning visual shape control of novel 3d deformable objects from partial-view point clouds," in *2022 IEEE International Conference on Robotics and Automation (ICRA)*, Philadelphia, PA, USA, 2022, pp. 8274–8281.

[27] H. Shi*, H. Xu*, Z. Huang, Y. Li, and J. Wu, "Robocraft: Learning to see, simulate, and shape elasto-plastic objects with graph networks," in *Robotics: Science and Systems (RSS)*, 2022. [Online]. Available: https://www.roboticsproceedings.org/rss18/p008.pdf

[28] X. Ma, D. Hsu, and W.-S. Lee, "Learning latent graph dynamics for visual manipulation of deformable objects," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 8266–8273.

[29] J. Collins, S. Chand, A. Vanderkop, and D. Howard, "A review of physics simulators for robotic applications," *IEEE Access*, vol. 9, pp. 51 416–51 431, 2021.

[30] S. James, A. J. Davison, and E. Johns, "Transferring end-to-end visuomotor control from simulation to real world for a multi-stage task," in *Proceedings of the 1st Annual Conference on Robot Learning*, S. Levine, V. Vanhoucke, and K. Goldberg, Eds., vol. 78. PMLR, 2017, pp. 334–343.

[31] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, *et al.*, "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 4243–4250.

[32] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac gym: High performance gpu-based physics simulation for robot learning," *CoRR*, vol. abs/2108.10470, 2021.

[33] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4238–4245.

[34] Y. Tang and S. Agrawal, "Discretizing continuous action space for on-policy optimization," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 5981–5988, 2020.

[35] Blender. (2018) Blender - a 3d modelling and rendering package. Stichting Blender Foundation, Amsterdam.

[36] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. (2017) Proximal policy optimization algorithms. [Online]. Available: https://doi.org/10.48550/arXiv.1707.06347
[37] K. Cobbe, J. Hilton, O. Klimov, and J. Schulman, "Phasic policy gradient," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 2021, pp. 2020–2027.
[38] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.
[39] É. Marchand, F. Spindler, and F. Chaumette, "Visp for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robotics & Automation Magazine.*, vol. 12, no. 4, pp. 40–52, 2005.