Citation for published version (APA):
Pecyna, L., Dong, S., & Luo, S. (2022). Visual-Tactile Multimodality for Following Deformable Linear Objects Using Reinforcement Learning. In *The 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2022)*

Visual-Tactile Multimodality for Following Deformable Linear Objects Using Reinforcement Learning

Leszek Pecyna[1], Siyuan Dong[2] and Shan Luo[3,*]

Abstract— **Manipulation of deformable objects is a challenging task for a robot. It would be problematic to use a single sensory input to track the behaviour of such objects: vision can be subjected to occlusions, whereas tactile inputs cannot capture the global information that is useful for the task. In this paper, we study the problem of using vision and tactile inputs together to complete the task of following deformable linear objects, for the first time. We create a Reinforcement Learning agent using different sensing modalities and investigate how its behaviour can be boosted using visual-tactile fusion, compared to using a single sensing modality. To this end, we developed a benchmark in simulation for manipulating the deformable linear objects using multimodal sensing inputs. The policy of the agent uses distilled information, e.g., the pose of the object in both visual and tactile perspectives, instead of the raw sensing signals, so that it can be directly transferred to real environments. In this way, we disentangle the perception system and the learned control policy. Our extensive experiments show that the use of both vision and tactile inputs, together with proprioception, allows the agent to complete the task in up to 92% of cases, compared to 77% when only one of the signals is given. Our results can provide valuable insights for the future design of tactile sensors and for deformable objects manipulation. Code and videos can be found at: `https://github.com/lpecyna/SoftSlidingGym`.**

I. Introduction

Humans and animals explore and interact with their environment through a variety of senses of different modalities. In some cases, we are able to observe the integration of different modalities when one signal affects the perception of other sensory inputs. An example of such multimodal influence is the McGurk effect [1], in which humans' perception of particular sounds is affected by visual cues. Touch and vision are especially used by humans during object identification and manipulation. This can be seen in neuropsychological studies on fMRI data, which shows that both visual and haptic signals are processed in a cross-modal fashion during some of these tasks [2], [3].

In contrast, most of the artificial systems are based on a single modality when performing their tasks and often different types of algorithms are developed to approach particular modalities. As robots are operating in more complex and dynamic environments, it can be expected that the usage of variety of sensing modalities will play a more important role for them [4].

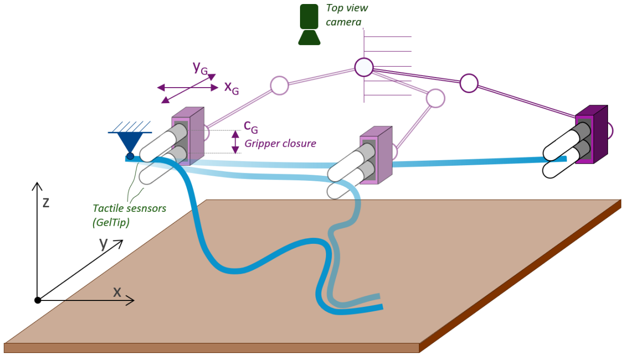

Fig. 1. Manipulation goal: the gripper starts from the fixed end of the rope/cable (on the left) and follows it, up to its tail end (on the right).

One of the situations when the environment becomes more complicated, and multimodal perception might help in better comprehension of it, is manipulation of flexible objects. As such, contour following of Deformable Linear Objects (DLOs) is a common task performed by humans, e.g., cable following. We perform this by grasping a cable between the thumb and the forefinger and slide the fingers to the target position [5], e.g., when untangling the cables or when following the cable to find its plug-end.

Cable following can be a challenging task for artificial systems as the cable shape is changing dynamically while the gripper is sliding. Moreover, different cables/ropes can be characterised by different stiffness and friction, and their starting shape might be complex and undetermined (kinks, intersections, etc.). Due to these challenges, most of the research concerning DLO manipulation uses some additional constraints, e.g., an object is placed on a table [6], [7].

Training a Reinforcement Learning (RL) agent in a simulated environment, in many aspects, is a desirable approach as the environment can be explored through an extensive number of episodes without possibility of damaging a robot. However, simulation of sliding and realistic grasping is a challenging task itself. For this reason, most of prior works on flexible object manipulation utilise a firm grasp, i.e., the section or a point of the object is fixed to the gripper and cannot move in relation to it [8], [9]. In our work, we cannot use such an approach as the gripper is sliding along the object, instead we aim to modulate the grasping force while the hand is moving.

This work was supported by the EPSRC project "ViTac: Visual-Tactile Synergy for Handling Flexible Materials" (EP/T033517/1).

[1] Leszek Pecyna is with the Department of Computer Science, University of Liverpool, Liverpool L69 3BX, UK; and with the Department of Engineering, King's College London, London WC2R 2LS, UK. Email: `leszek.pecyna@kcl.ac.uk`

[2] Siyuan Dong is with the Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle, WA, USA. Email: `siyuandong.bme@gmail.com`

[3] Shan Luo is with the Department of Engineering, King's College London, London WC2R 2LS, UK. Email: `shan.luo@kcl.ac.uk`

∗Corresponding author.

In this paper, we create an RL agent for a cable/rope following task in a simulated environment and investigate how its behaviour can be boosted using visual-tactile fusion, compared to using a single sensing modality. As shown in Fig. 1, we have both visual and tactile perspectives of the state of the deformable linear object in the gripper. The robot agent's goal is to pick up the object at its fixed end and follows it up to its tail end. We chose this task, as sliding along DLOs is not well explored, especially when it comes to simulation, and it is a good candidate for research concerning visual-tactile synergy as both of the signals can provide useful information to complete the task.

To the best of our knowledge, this is the first study to use both vision and tactile inputs for the task of cable following. We used distilled information as the observations of the agent, e.g., the object pose in both visual and tactile perspectives, instead of the raw visual images or tactile data. In this way, the trained agent can be directly transferred to real environments, and the learned control policy can be disentangled from the perception system. Through our extensive experiments, we find that when both vision and tactile inputs, together with proprioception, are used, the agent can complete the task (reach the end of the cable and hold it) in up to 92% of cases, compared to the best result of 77% with a single sensory input used (for vision); and when two signals were used – 89% (for vision and proprioception).

II. Related Work

A. Visual-Tactile Multimodality

Vision and touch are two main important senses used for object manipulation. They have been widely used in robotics but, in most of the cases, with only one sensory input used [10], [4]. In the resent years, there have been several studies aiming to combine both of these inputs. Many of them concentrate on sensing, rather than on manipulation, like feature sharing or feature extraction [11], [12]. When it comes to object manipulation itself, even when both of the senses are utilised, in many cases, one input type supports another one, and is used in a specific sub-task, e.g., tactile sensing can help to verify the contact with the object (several examples of that are provided in [10]). Hence, they are not used simultaneously together.

There are a few works to extract features from the sensors and use them together in the control scheme. In [13], multiple sensory inputs are integrated in a grasping stability task (of mugs and bottles). In [14], in-hand object location is estimated from joint sensors, and [15] covers object pose estimation. The work presented herein follows the idea where the extracted features are used simultaneously together.

B. DLO Following

There have been several studies concerning contour following of rigid objects that utilise vision [16] or tactile sensing [17], [18]. From the point of view of flexible object following that utilises tactile sensors, there are two works [19] and [5]. The first [19] proposes a reinforcement learning approach to close a deformable ziplock bag using BioTac sensors. The robot grasps and follows the edge of the bag using a constant grasping force. The authors used Contextual Multi-Armed Bandits (C-MAB) RL algorithm to train the robot to close the bag in discrete time steps with a maximum velocity of 0.5 cm/s (trapezoidal velocity profiles). The second [5], which is the most relevant to our work from the point of view of the task, presents a control framework that uses a real-time tactile feedback from a GelSight sensor [20], [21] to accomplish the task of cable following. To achieve that, the authors designed a parallel gripper with a servo motor actuator. In their study, only the tactile signal was used to modulate gripping force. The RL was not used in [5], instead two controllers were used: PD – for cable grip control, and LQR – for cable pose control. Compared to the decoupled controllers in [5], in this work we control both the gripping force and the end-effector pose simultaneously using the RL policy.

III. Problem Statement

The goal of the presented DLO following task is to grasp the cable/rope at the beginning end with the gripper, and follow it – using an appropriate grasping force – to its tail end. The task should finish by holding the object close to its finishing end. The beginning of the rope is firmly held by the second gripper (it is attached to a point in space in the simulator), as illustrated in Fig. 1.

In many aspects, this task is similar to the one presented in [5]. There are, however, many differences: we performed the training in a simulator (we are planning to test our model on a real platform in the next stages of our work); we use data from both vision and tactile sensors (in [5] only tactile signal was used); our model uses an RL algorithm (compared to decoupled PD and LQR controllers in [5]). Also, we do not perform re-grasping procedure[1]. Instead, we finish a training episode when the DLO falls from the gripper. We assume no plug at the end of the cable (which allowed cable-end recognition by the tactile sensor in [5]). Hence, in our case, we expect the vision to play a principal role in the object-end identification. As our task is conducted in the simulator, the parameters we chose can make the object properties correspond to those of a cable or a rope, which can be much softer than the cable used in [5]. As in [5], we consider a planar motion; in this way the extracted angles from a tactile sensor and a top-view camera provide sufficient information about DLO configuration in the plane of motion.

Our model of DLO manipulation is defined as finite-horizon, discounted Markov decision process (MDP) represented by a tuple of ($\mathcal{S}$, $\mathcal{A}$, p, $\mathcal{R}$). The state space $\mathcal{S}$ and action space $\mathcal{A}$ are assumed to be continuous. State transition probability p, represents the probability density of the next state $s_{t+a} \in \mathcal{S}$ given the current state $s_t \in \mathcal{S}$ and action $a_t \in \mathcal{A}$. $\mathcal{R}$ is an immediate reward emitted by the environment on each transition. The details about the agent's actions, state space and the definition of a reward function implementation are provided in the next section.

[1] This was not part of the controller task in [5], instead it was used when the gripper recognises it is losing the cable or reaches workspace limits.

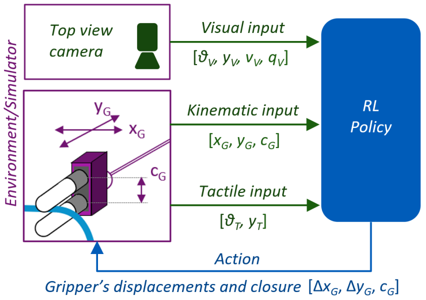

Fig. 2. DLO following task with the RL policy.

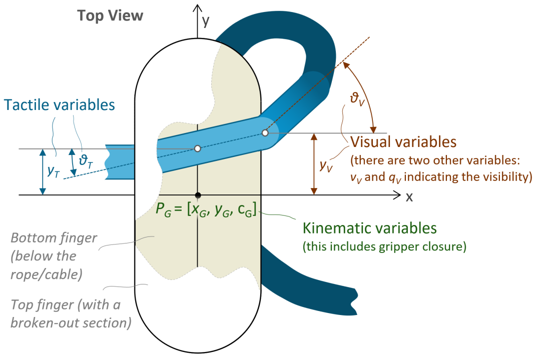
Top View
y
x
Tactile variables
y_T
ϑ_T
ϑ_V
Visual variables
(there are two other variables: v_V and q_V indicating the visibility)
y_V
$P_G = [x_G, y_G, c_G]$
Kinematic variables
(this includes gripper closure)
Bottom finger (below the rope/cable)
Top finger (with a broken-out section)

Fig. 3. Illustration of state variables (observations) available for the model.

IV. Methodology

In this section we present our methodology for implementation and usage of an RL agent to solve the task of DLO following. We first describe the model, its architecture, what observations it uses and what actions the agent can make. Next, we present the reward function that is set to promote the behaviour of reaching the object-end and staying there. Finally, we explain how our model performance is evaluated.

A. Agent Description

In our study we use Soft Actor-Critic (SAC) [22] that provides state-of-the-art performance in continuous control tasks (like robotic manipulation). SAC combines sample efficient off-policy method with ability to operate in a continuous action and state spaces.

1) Model Architecture: The model is composed of an actor network and a critic which is made of two Q-value networks (to combat the problem of overestimation of Q-values).

Both Q-value networks and the policy network are MLPs with two hidden layers of 1,024 neurons with ReLU activation function. The actor takes as an input the state and outputs the mean and covariance for the Gaussian distribution that represents the policy [22]. From that the action is sampled. The Q-value network input is made of actions together with observation space and produces single values (Q-value). The model's general scheme of interaction with the environment can be seen in Fig. 2.

2) Observations: In general, observations we use in our model can be divided in three categories as shown in Fig. 3.

- Kinematic (proprioceptive): it provides the information about the position of the gripper in the space (x and y coordinates) and its closure state (variable from 0 to 1, where 0 corresponds to the situation where gripper is fully open and 1 where it is fully closed). It is an array of 3 components: $O_G = [x_G, y_G, c_G]$.
- Tactile: Described more in Section V-B, it is composed of the angle and the position of the DLO in relation to the gripper: $O_T = [\vartheta_T, y_T]$.
- Visual: Described more in Section V-C, the visual input is composed of 4 components: information of if the cable is visible on the right side of the gripper, how confident the angle is, the angle, and the y position of of the cable in relation to the gripper: $O_V = [v_V, q_V, \vartheta_V, y_V]$.

3) Actions: As the gripper is supposed to move freely in the x-y space, our action array is composed of target displacements in these directions. Apart from that, the agent is able to modify the closing force of the grip hence the array of action is composed of: $A = [\Delta x_g, \Delta y_g, c_G]$. As the time step is constant in the simulation (set to 0.01 s) these Δx and Δy values directly correspond to velocities of the gripper. The simulator allows to set how many time steps are executed after each action step, in our case this was set to 8. We limit these values (by scaling the output of the agent) to keep the speed of the gripper in a feasible range for the real robot and also to assure better RL training. After some preliminary experiments, we chose maximal values of Δx and Δy to be 0.0025, which corresponds to the velocity of 0.25 m/s.

4) Reward: Our reward can be represented by partial rewards and defined as:

$$R_t = \begin{cases} R_{move} + R_{end}, & \text{if } n_h \neq 0 \\ P_{fall}, & \text{if } n_h = 0 \end{cases}$$

where R_{move} is a reward for moving towards the end of the DLO; R_{end} is the reward for being close to the end of the rope; P_{fall} is the penalty for dropping it; n_h is a number of particles being held by the gripper.

The partial rewards are defined as follows:

$$R_{move} = \alpha_{move}(d_t - d_{t-1})$$

where α_{move} is the weight of that reward, d_t is the distance (in meters) at the current time step t. In the simulation, it is calculated as:

$$d_t = \frac{p_i L_c}{n_c}$$

where p_i is the average value of indexes of particles being held by the gripper (the starting particle is indexed as 1 and the end particle is indexed as n_c that is the number of particles of the rope), L_c is the length of the rope.

R_{end} is given only when the gripper is less than 20 particles from the end of the object and it is increasing linearly when fingers approach the end[2].

$$R_{end} = \begin{cases} \alpha_{end}(p_i + 20 - n_c), & \text{if } p_i > n_c - 20 \\ 0, & \text{otherwise} \end{cases}$$

[2] Quadratic function was also tested.

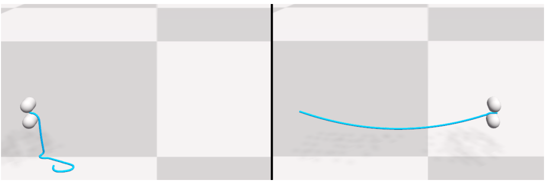

Fig. 4. Frames from the simulated environment. On the left – the beginning of the task where the cable falls freely; on the right – the gripper finished the task and holds the cable's end.

P_{fall} is a constant value. Together with α_{move} and α_{end}, it was chosen through a hyperparameter search[3]. These values were set to $P_{fall} = -0.5$, $\alpha_{move} = 10$, and $\alpha_{end} = \frac{1}{20}$.

5) Evaluation Metrics: We evaluate the performance of our model using several metrics. One of them is to classify each of the completed episodes in one of the categories:

- *Hold the end* – the gripper follows the DLO till its end, stays there and holds the object. This is the goal behaviour. We defined being at the end as the situation when the gripper holds any of the last 10 particles.
- *Stop before* – the gripper did not reach close to the end but it did not drop the object.
- *Reach end but drop* – the gripper reached the end of the DLO (last 10 particles) but failed to keep the object.
- *Drop before* – the gripper dropped the DLO earlier, without reaching its end.

Apart from this classification, two more metrics are used:

- *Time spent at the end* – we check how long (i.e., how many time steps) the gripper spends at the end of the DLO (at any of 10 last particles).
- *How far it goes* – we check how close to the end of the DLO gripper reached (we measure this distance from the end of the object as the length of the rope is randomised). The distance is measured in particles.

V. Experiment Setup

We simplified our observations to scalars (e.g., angles and positions). This allows relatively simple assessment of what piece of information is useful, ensuring that the observations are not influenced by the process of information extraction from the real images. This approach could potentially help us to avoid domain shift in the future Sim-to-Real transfer. It also makes this method more general, allowing its application with different types of sensors.

A. Simulation

We use Nvidia Flex – a particle based simulation technique [23], [24], wrapped in SoftGym [9] - which is a set of benchmarks. SoftGym provides simulated environments and agents, and it uses PyFlex [25] that provides Python interface for Nvidia Flex, and Gym [26], a toolkit for developing and comparing reinforcement learning algorithms.

We constructed a new task and environment in SoftGym based on "rope flatten" task. We assumed the usage of a camera and a gripper with tactile sensors, based on that we amend the environment. We use two capsule-shape objects to simulate GelTip sensors [27]. We adjusted how the gripper can move and introduced a new way of gripping rope's (or a cable's) particles (Section V-D). The screenshots from the simulator, with an example of the rope configuration at the beginning and at the end of the task, can be seen in Fig. 4.

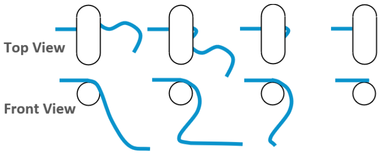

Fig. 5. Possible occlusions in rope visibility caused by the gripper (capsule in the simulation).

B. Tactile sensors

The sensor provides us with an angle ϑ_T of the cable with the gripper's x axis as presented in Fig. 3, and with the position of the cable in the y direction (along the finger). In the case of the GelSight like sensors, these angles and positions can be obtained using algorithms described in [20]. We use particles' positions (between gripper's fingers) in the simulator to calculate these values (we fit the line to the centres of these particles and use its angle and its intersection with y axis). In the case of the real sensor, the precision depends on the normal force. This is well illustrated in [5] (for the GelSight sensor), where the controller is adapting the gripping force taking into account tactile quality.

In some of the experiments presented herein we added a random noise – proportional to the gripper closure – to the measured angle and position.

$$\vartheta_T = \vartheta_{T,nom} + (1 - c_G)\vartheta_{noise},$$

$$y_T = y_{T,nom} + (1 - c_G)y_{noise},$$

where ϑ_{noise} and y_{noise} are sampled from the normal distributions with different standard deviations (depending on assumed sensitivity); $\vartheta_{T,nom}$ and $y_{T,nom}$ are the nominal values. These are calculated based on the positions of the DLO particles between the grippers fingers.

There are two reasons why we imitate the finger-like sensors in our simulations. First, we are planning to use GelTip sensors in the future experiments with the robot. Second, due to the nature of the simulator – a cable/rope is made of particles connected with springs in the Nvidia Flex environment – rectangular-shape fingers cause unwanted behaviour of the rope which is difficult to overcome, e.g., the corner of the sensor gets between particles when the gripper is pulling the cable. Hence, more rounded shape is more appropriate for this simulation task.

C. Camera

We use a top-view camera and in this way we simplify the task to a planar problem (see Section III). Similarly as with tactile data, we extract and use the position and angle of the rope from the camera image. We use the position

[3] For simplification, we performed hyperparameter search for all modalities together – for each parameter we checked the model behaviour and performance with different sensing modalities.

and angle of the rope on the right from the gripper (in the direction of motion) as presented in Fig. 3 – ϑ_V and y_V variables. The camera's top-view can be subjected to gripper and cable occlusions, and not always the position and angle of the cable are visible in the image as illustrated in Fig. 5 (top row). Hence, we also include the information if the cable is visible from the right – v_V, and how far continuously it goes to the right – the confidence about the angle – q_V.

In the simulation, the algorithm analyses the positions of DLO particles. After finding the first particle at the edge of the finger we follow the rope and record when the particles change the direction – when their x component decreases more than the assumed value (0.25 particle radius). We use the particles between the finger edge and the direction change (we check maximum 10 particles) to fit the line and obtain the angle. The confidence value corresponds to the normalised distance along these selected particles. A similar method can be applied for real image input where the DLO is detected using topology extraction and vectorisation algorithm (e.g., [28]). In that case, we would analyse the segments of the rope to detect the change of direction.

D. Gripper

The gripper can move in x and y directions and close or open the grip, similarly as in [5]. We implemented the process of gripping with variant gripping force in the simulated environment. This task is not trivial and in most simulations it is implemented by attaching the object being held to the gripper without taking into account possibility of sliding or factors such as friction. This was the case of the original environments implemented in SoftGym [9]. As the rope is in fact simulated as particles connected by springs, simple decreasing of the gap between two capsules (that we used to represent the gripper's fingers) was causing unnatural behaviours – the gripper was getting caught between the particles holding them firmly.

Instead of doing that, our intention was to modify the friction between the DLO and the fingers. This, on the other hand, was not straight forward because the friction parameters are global in the simulator. To this end, we modify the value of the inverse mass of the rope's particles that were between the fingers, and amended their positions according to the grippers' movement. This is similar to the approach in the original environments of the SoftGym, where the inverse mass was set to 0 and the position was set to follow the gripper – causing that the particle was fully attached to it.

Assuming that the closing action is scaled between 0 and 1 and that the gripper changes the friction (or rather the inverse mass of held particles) when that value is above 0.5 but the full closure appear at 0.9, the inverse mass w_p can be expressed as:

$$w_p = max(2.25 w_{p,nom} - 2.5 w_{p,nom} max(c_G, 0.5), 0)$$

where $w_{p,nom}$ is the nominal value of the inverses mass of DLO particles used in the simulator.

E. Randomisation

To allow our agent to operate in different environments we randomised variety of parameters in the simulation. This randomisation makes the agent to better generalise the task and could allow Sim-to-Real transfer even when the real environment, e.g., cable parameters, are quite different from these in the simulation.

We randomised:

- Length of the cable (uniform, from 30 to 60 particles);
- DLO starting position – we pick the rope in a random place and place it in a random location, we repeat that 4 times before each episode;
- DLO stretch stiffness (uniform, from 0.8 to 1.4);
- Bending stiffness (uniform, from 0.8 to 2.4);
- Friction coefficient (uniform, from 0.04 to 0.3);

The ranges of these parameters were chosen empirically in the simulator. Changing some of them in a bigger range would require to decrease the simulation step, which extends the training time. In these ranges the simulation was stable and at the same time we could observe different interactions between the gripper and the DLO.

F. Training the agent

In the training, we used a batch size of 128, learning rates for actor and critic of 0.001, 1,000 initial steps (with no agent updates), horizon length for each episode of 150 steps, and the maximum number of training steps was set to 50,000. The reward discount was set to $\gamma = 0.99$.

VI. Results

To test the proposed methods and investigate how different sensory inputs contribute to the rope following task, we conducted multiple experiments. First, we compared the behavior of the agent and its performance when only one of the signals is provided. Next, we conducted similar experiments using two out of three inputs. Following that we performed the ablation studies where we trained the agent with all three inputs but we tested it excluding one of the signal. At the end, we check the agent's performance when different sensitivity of the tactile sensor is used – different randomisation when the gripper is open.

Although the task performed throughout our experiments is similar to the one in [5], some differences make the result comparison implausible. Studies presented in [5] centred around the velocity and performance of the gripper (and its controller) and on the gripper construction itself. In our research we focus on comparing different input modalities and on investigation of RL agent's behaviour.

All the results are collected in such a way that every 200 training steps we evaluate agent performance using 10 random environments (i.e., with a random DLO properties and configurations). We repeat the whole training 10 times for different random seeds (this way we train 10 different independent agents). The results presented in the paper are the average values from these 10 independent training sessions together with 95% confidence margin. The curves

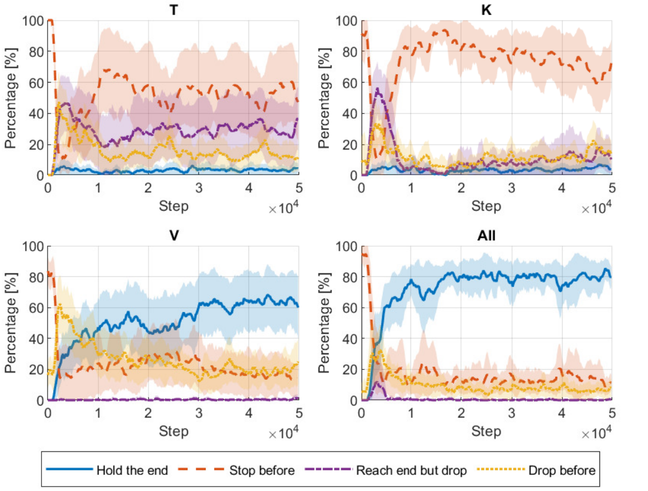

Fig. 6. Episode outcome. Each subplot corresponds to a different sensory input: tactile, kinematic, vision, or all. The curves were collected from 10 independent training sessions, shading represents 95% confidence range.

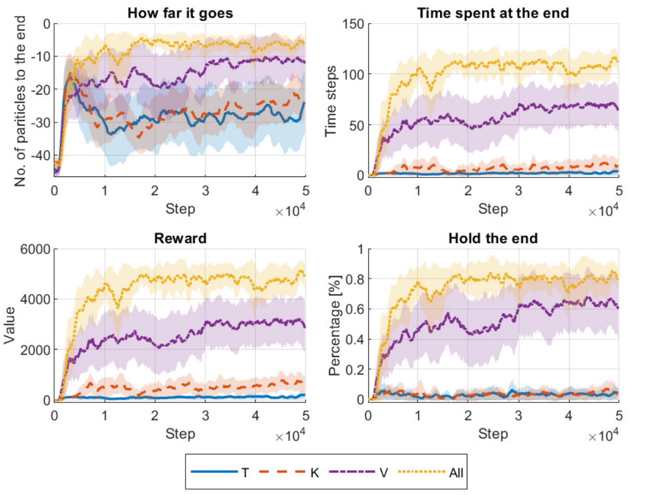

Fig. 7. Agent's performance evaluation when one type of the inputs is used. Each subplot corresponds to a different metric. The curves were collected from 10 independent sessions, shading represents 95% confidence range.

were additionally smoothed using a window size of 5 (i.e., the average of 5 following results).

A. Training Performance When a Single Sensory Input is Provided

We first analyse the results obtained when only one of the inputs was used (tactile, visual or kinematic). These are compared with the results when all three signals are provided together. Fig. 6 shows the outcome of the RL agent behaviours; how often the task was finalised in a most desired way: *Hold the end*; or how frequently other 3 outcomes were observed (described more in Section. IV-A.5). We can see that the behaviour of the agent changes significantly depending on the input. Only vision allows the model to stop and hold the DLO at the appropriate moment (*Hold the end*). This was expected, as both *T* and *K* signals do not hold information that allows to identify the end of the object.

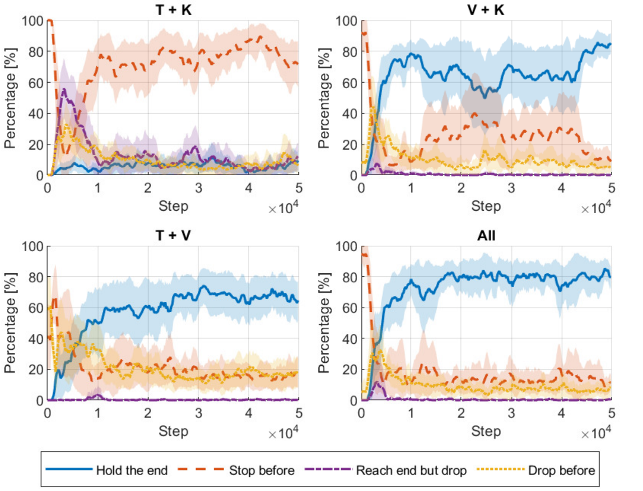

Fig. 8. Episode outcome. Each subplot corresponds to a different combination of sensory inputs. The curves were collected from 10 independent training sessions, shading represents 95% confidence range.

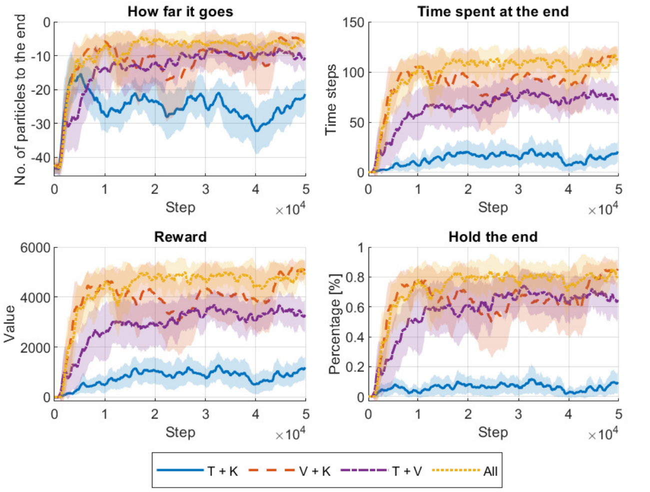

Fig. 9. Agent's performance evaluation when two types of inputs are used. Each subplot corresponds to a different metric. The curves were collected from 10 independent sessions, shading represents 95% confidence range.

This is, however, improved when all the signals are used together, in that case, the agent is able to obtain much better performance faster. As the kinematic signal does not provide any information about the DLO itself the agent prefers to not follow along the object and stops prematurely. Simulations with a tactile input show similar behaviour but the agent relatively often tends to follow the object till the end and drops it after that.

The results from particular modalities are compared more directly in the Fig. 7. Each curve represents a different type of inputs and each subplot shows a different type of metrics. In the figure, we also included episode collective reward and the most desired outcome: *Hold the end*. The best mean results for each modality are 12%, 11%, 77% and 92% (*Hold the end*), respectively for *T*, *K*, *V* and *All*[4].

[4] As we mentioned before, the results in figures are smoothed for better readability, therefore listed results might not be directly visible.

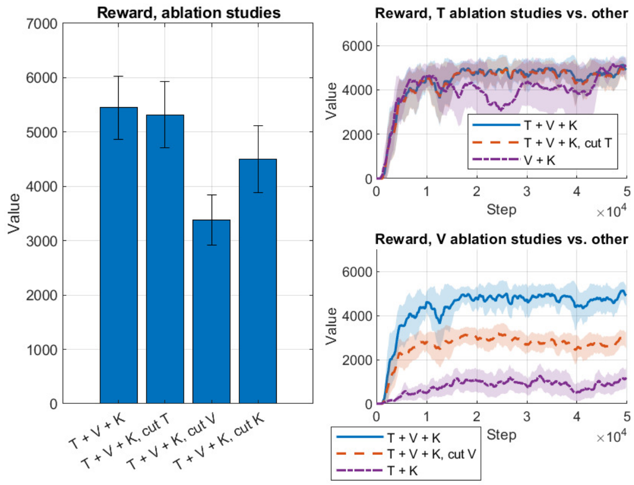

Fig. 10. Performance of the agent when trained with all of the sensory inputs but some of the inputs were not provided in the testing phase. On the left subplot – episode reward in a particular case; on the right – comparing ablation studies (when tested without T or V) with all signal case and with the case when the model was trained from the beginning without one of the inputs. Error bars and shading correspond to 95% confidence range.

These results allow us to make a clear conclusion that the agent with a visual input outperforms the agents trained with other signal types. However, it is also clear that when other inputs are included this performance is improved. We can also see that both kinematic and tactile inputs help in sliding along DLO (*How far it goes* subplot)[5].

B. Training Performance When Two Sensory Inputs Are Provided

Figures 8 and 9 are created in the same manner as the figures in the earlier subsection. As expected, we can observe better performance of the training when two sensory inputs are used. When we compare the V subplot from Fig. 6 with subplots $V + K$ and $T + V$ presented here, we can see that each of the inputs (T or K) provides some improvement.

Again, we can see that the visual signal plays a crucial role and only when it is included in the input the agent is more often successfully performing *Hold the end* behaviour. The best mean results for each paired-modalities are 16%, 89%, 77% and 92% (*Hold the end*), respectively for $T + K$, $V + K$, $T + V$ and *All*.

In the case of *How far it goes* metric, any combination that contains visual input ($V + K$ or $T + V$) allows to achieve relatively high performance, similar to the one with *All* signals.

C. Ablation studies

We trained the agent using all of the inputs ($T + V + K$), but in this experiment when the model was tested we ablate one of the signals to investigate its effect on the agent's performance. The performance of the agent is presented in the left subplot in Fig. 10, where we show the mean value of collective episode reward. We can see that removal of V signal caused the biggest drop in performance. Ablation of T input, on the other hand, was insignificant and the results are almost the same as with that signal. This implies that the tactile input is in some way complementary.

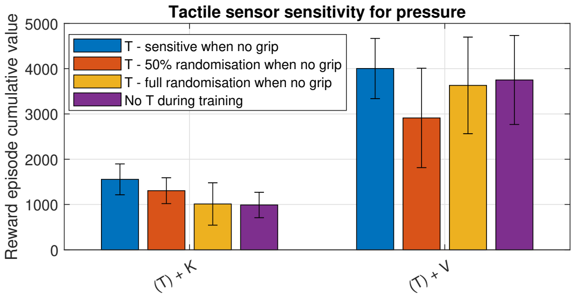

Fig. 11. Performance of the agent (episode reward) when different sensitivity of the tactile sensor was assumed (sensitivity vary depending on the grasping force). Error bars correspond to 95% confidence range.

To investigate the influence of the tactile signal, we present our ablation studies on a training development plot (Fig. 10, top right). The curve with removed signal is obtained in a way that the the agent is trained with all inputs but every 200 steps is tested with a tactile-free signal[6]. We can see that removal of T input has practically no impact on the reward. These results were compared with the session where tactile input was not used at all during the training. The results are interesting as we can see that the model which has access to additional tactile information in the training phase can learn faster and obtain better performance than the model trained without tactile signal T (both agents are tested without T input). Possible explanation is that the DLO angle information (which is more certain in the case of tactile input) is useful in the training process to interpret and take advantage of the position information. Similar observation is even more evident when the visual signal is removed (Fig. 10, bottom right). The agent that was trained with all signals while tested without visual input performs much better compared to the model that was trained without vision.

D. Tactile Sensitivity Study

As described in Section V-B, we took into account the impact of tactile sensor sensitivity. We included the random noise in the tactile input that was equal to zero when the gripper was using maximum gripping force and was increasing with gripper opening. Fig. 11 shows the results for different sensitivity and when the tactile input is not included at all. *Full randomisation when no grip* corresponds to the situation where the standard deviation of aforementioned randomisation was set such high that angle should be practically not useful to estimate the real orientation of the rope (σ was set to 0.5 rad) and position should be very unreliable (σ was set to 0.002 m). In that case, we can see that the agent was not able to learn to use that input (during the 50,000 steps

[5] This edge following is fully possible with a tactile signal but it is not preferred by the agent due to the penalty for cable dropping which happens when it reaches the end.

[6] We use the same agent for tactile-free and all signals tests, hence, the curve has a very similar characteristic. The difference in performance is more visible when checking other metrics.

training) – the reward was as good as in the case of lack of tactile input. However, we can observe improvement of the agent performance when partial randomisation was used if the K input was provided together with T. This shows that such less sensitive input can be used by the agent but only with the information about the gripper's closure, which allows to evaluate the reliability of the tactile signal.

VII. Conclusions and Future Works

In this paper, we investigated the use of both vision and tactile inputs in completing a task of following deformable linear objects. We introduced a benchmark in simulation and studied how an RL agent's behaviour can be boosted using visual-tactile fusion, compared to using single sensing inputs. We also conducted ablation studies where the agent, trained with a larger amount of signals, was tested with fewer inputs. Our results show the importance of multimodality and each sensing modality plays a different role in completing the task. We find that vision played a crucial role in finishing the task and finding the end of the cable. Without vision (and due to the nature of our reward function), the agent prefers to finish the movement prematurely. We also see the importance of kinematic input which allows the agent to know where it is. Tactile input in some aspect was redundant with visual input, however, as we showed it was important for the agent to go further along the cable. The importance of the tactile signal is more significant when vision is not available, which is common due to obstacles in real-life situations.

The results presented in this paper provide useful insights for future designs of tactile sensors and for deformable objects manipulation. The presented approach can provide guidance in the process of simulating tasks where sliding or touching of flexible materials is required. One of future works will be to adapt the trained agent on a real platform. Thanks to the usage of the distilled information, such transfer of knowledge should be much less affected by a domain shift. The research also has potential to be extended to more complex tasks, where we manipulate other objects like a cloth or train the agent to achieve a different goal, e.g., wrapping a cable around a pin.

References

[1] H. McGurk and J. MacDonald, "Hearing lips and seeing voices," *Nature*, vol. 264, no. 5588, pp. 746–748, 1976.

[2] T. W. James, G. K. Humphrey, J. S. Gati, P. Servos, R. S. Menon, and M. A. Goodale, "Haptic study of three-dimensional objects activates extrastriate visual areas," *Neuropsychologia*, vol. 40, no. 10, pp. 1706–1714, 2002.

[3] R. Blake, K. V. Sobel, and T. W. James, "Neural synergy between kinetic vision and touch," *Psychological science*, vol. 15, no. 6, pp. 397–402, 2004.

[4] S. Luo, N. F. Lepora, U. Martinez-Hernandez, J. Bimbo, and H. Liu, "Vitac: Integrating vision and touch for multimodal and cross-modal perception," *Frontiers in Robotics and AI*, vol. 8, 2021.

[5] Y. She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, and E. Adelson, "Cable manipulation with a tactile-reactive gripper," *The Int. Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1385–1401, 2021.

[6] M. Yan, Y. Zhu, N. Jin, and J. Bohg, "Self-supervised learning of state estimation for manipulating deformable linear objects," *IEEE robotics and automation letters*, vol. 5, no. 2, pp. 2372–2379, 2020.

[7] J. Zhu, B. Navarro, P. Fraisse, A. Crosnier, and A. Cherubini, "Dual-arm robotic manipulation of flexible cables," in *2018 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. IEEE, 2018, pp. 479–484.

[8] 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 Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2015, pp. 1382–1388.

[9] X. Lin, Y. Wang, J. Olkin, and D. Held, "Softgym: Benchmarking deep reinforcement learning for deformable object manipulation," in *Conf. on Robot Learning*, 2021, pp. 432–448.

[10] S. Luo, J. Bimbo, R. Dahiya, and H. Liu, "Robotic tactile perception of object properties: A review," *Mechatronics*, vol. 48, pp. 54–67, 2017.

[11] W. Yuan, S. Wang, S. Dong, and E. Adelson, "Connecting look and feel: Associating the visual and tactile properties of physical materials," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 5580–5588.

[12] S. Luo, W. Yuan, E. Adelson, A. G. Cohn, and R. Fuentes, "ViTac: Feature sharing between vision and tactile sensing for cloth texture recognition," in *2018 IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2018, pp. 2722–2727.

[13] Y. Bekiroglu, D. Song, L. Wang, and D. Kragic, "A probabilistic framework for task-oriented grasp stability assessment," in *2013 IEEE Int. Conf. on Robotics and Automation*, 2013, pp. 3040–3047.

[14] P. Hebert, N. Hudson, J. Ma, and J. Burdick, "Fusion of stereo vision, force-torque, and joint sensors for estimation of in-hand object location," in *IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 5935–5941.

[15] J. Bimbo, P. Kormushev, K. Althoefer, and H. Liu, "Global estimation of an object's pose using tactile sensing," *Advanced Robotics*, vol. 29, no. 5, pp. 363–374, 2015.

[16] F. Lange, P. Wunsch, and G. Hirzinger, "Predictive vision based control of high speed industrial robot paths," in *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 3, 1998, pp. 2646–2651.

[17] C. Lu, J. Wang, and S. Luo, "Surface following using deep reinforcement learning and a GelSight tactile sensor," *arXiv preprint arXiv:1912.00745*, 2019.

[18] B. Ward-Cherrier, N. Pestell, L. Cramphorn, B. Winstone, M. E. Giannaccini, J. Rossiter, and N. F. Lepora, "The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies," *Soft robotics*, vol. 5, no. 2, pp. 216–227, 2018.

[19] R. B. Hellman, C. Tekin, M. van der Schaar, and V. J. Santos, "Functional contour-following via haptic perception and reinforcement learning," *IEEE transactions on haptics*, vol. 11, no. 1, pp. 61–72, 2017.

[20] W. Yuan, S. Dong, and E. H. Adelson, "Gelsight: High-resolution robot tactile sensors for estimating geometry and force," *Sensors*, vol. 17, no. 12, p. 2762, 2017.

[21] D. F. Gomes, P. Paoletti, and S. Luo, "Generation of GelSight tactile images for Sim2Real learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 4177–4184, 2021.

[22] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Int. Conf. on Machine Learning*, 2018, pp. 1861–1870.

[23] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff, "Position based dynamics," *Journal of Visual Communication and Image Representation*, vol. 18, no. 2, pp. 109–118, 2007.

[24] M. Macklin, M. Müller, N. Chentanez, and T.-Y. Kim, "Unified particle physics for real-time applications," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, pp. 1–12, 2014.

[25] Y. Li, J. Wu, R. Tedrake, J. B. Tenenbaum, and A. Torralba, "Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids," *arXiv preprint arXiv:1810.01566*, 2018.

[26] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," 2016.

[27] D. F. Gomes, Z. Lin, and S. Luo, "GelTip: A finger-shaped optical tactile sensor for robotic manipulation," in *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2020, pp. 9903–9909.

[28] Z. Zhang, X. Liu, C. Li, H. Wu, and Z. Wen, "Vectorizing line drawings of arbitrary thickness via boundary-based topology reconstruction," in *Computer Graphics Forum*, vol. 41, no. 2, 2022, pp. 433–445.