Workshop on Knowledge Engineering for Planning and Scheduling at KEPS 2022

Knowledge Representation to Enable High-level Planning in Cloth Manipulation Tasks

Irene Garcia-Camacho, Júlia Borràs and Guillem Alenyà

Institut de Robòtica i Informàtica Industrial, CSIC-UPC
08028, Barcelona, Spain
igarcia@iri.upc.edu, jborras@iri.upc.edu, galenya@iri.upc.edu

Abstract

Cloth manipulation is very relevant for domestic robotic tasks, but it presents many challenges due to the complexity of representing, recognizing and predicting the behaviour of cloth under manipulation. In this work, we propose a generic, compact and simplified representation of the states of cloth manipulation that allows for representing tasks as sequences of states and transitions semantically. We also define a Cloth Manipulation Graph that encodes all the strategies to accomplish a task. Our novel representation is used to encode two different cloth manipulation tasks, learned from an experiment with human subjects manipulating clothes with video data. We show how our simplified representation allows to obtain a map of meaningful steps that can serve to describe cloth manipulation tasks as domain models in PDDL, enabling high-level planning. Finally, we discuss on the existing skills that could enable the sensory motor grounding and the low-level execution of the plan.

Introduction

The manipulation of highly-deformable objects is becoming an important area of robotic manipulation research, with very interesting potential applications in industrial, domestic or health care scenarios. Despite its importance, core capabilities such as grasping, placing, or handing to a person still remain as a hard and unsolved problem when dealing with textiles, as opposed to rigid objects. This is because a rigid object state can be defined by 6 pose parameters, while the deformation space of a fabric is infinite dimensional. This huge dimensional jump makes usual manipulation solutions not applicable to deal with textiles. In particular, the complexity of defining and recognizing scene states dealing with clothes makes any reasoning symbolic representation of scenes very difficult to ground, hindering the training of AI systems and task planners.

Although learning techniques can benefit from simulation, the transfer to reality has only been successful for simple skills (Matas, James, and Davison 2018; Yan et al. 2020; Hoque et al. 2020; Tanaka, Arnold, and Yamazaki 2018), because simulated cloth differs highly from real behaviour. There have been some works learning from real data using either video and sensory-motor data from a robot performing the manipulation in teleoperation (Yang et al. 2016) or from demonstrated robot actions connecting different images of the scene (Lippi et al. 2020). However, they show clear limitations when it comes to generalizing to other tasks (Yang et al. 2016) or when the scene contains cloth with self-occlusions (Lippi et al. 2020). It is even less common to learn cloth manipulation tasks from human demonstrations. However, learning from humans would be important to obtain a diversity of strategies to accomplish a task, and with different parameters related to safety, fast accomplishment of the objective or number of steps needed to accomplish a task, inducing a measure of task complexity. Learning through human demonstration follows a pipeline similar to Fig. 1. Large amounts of data could be obtained from human demonstrations in the form of video data and motion data of the hands (Verleysen, Biondina, and Wyffels 2020), but learning from this kind of data is challenging due to the difficulty of annotating data and recognizing cloth states from images.

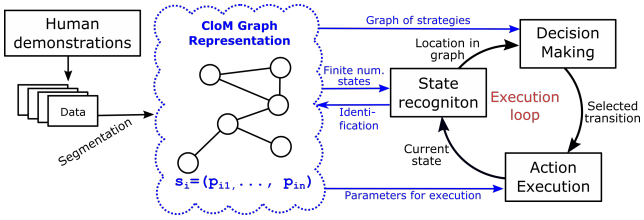


Figure 1: Generic pipeline for learning from human demonstration for manipulation tasks. A good task representation learned from the segmentation of the data can be used for decision making. State representations have to be defined to ease state recognition but also to enable action execution.

Another challenge for cloth manipulation is to find general solutions (Sanchez et al. 2018). Most recent attempts to find general approaches consist of end-to-end learning approaches (Tanaka, Arnold, and Yamazaki 2018; Yan et al. 2020; Hoque et al. 2020; Lippi et al. 2020) that are still limited to relatively simple tasks with limited self-occlusions, that is, with fabrics laying flat or semi-flat on a table. We believe that the key to general solutions is to define a domain of semantic scene states (cloud box in Fig. 1) with carefully chosen parameters to facilitate state recognition and localization of manipulation relevant features, that is, that can be grounded with real-sensor data.

The literature of cloth manipulation has dealt since the

very beginning on how to represent cloth and simplify it to a tractable manner to plan actions (Miller et al. 2012; Doumanoglou et al. 2016). In the Related work Section we will review how the literature has dealt with the problem of scene representation in the context of cloth manipulation. Approaches range from very simple 1-dimensional representations (Petrík et al. 2016) to mesh representations of the cloth (Bersch, Pitzer, and Kammel 2011), including to end-to-end learning approaches using the whole scene state image as state definition (Yan et al. 2020). However, none of the representations used has defined a full domain model to enable PDDL planning.

The first contribution of this work is to propose a novel idea to define a semantic scene state in cloth manipulation tasks. The novelty lies in including information on how the cloth is grasped (Borràs, Alenyà, and Torras 2020), where it is grasped from, what are the environmental contacts and the possible transitions between them. The second contribution is the Cloth Manipulation (CloM) Graph, a graph that can be built using the previous representation to encode all the possible states and transitions of a given manipulation task seen from video demonstrations, enabling to capture the diversity of strategies. We show the feasibility of our approach by extracting the graph for two textile manipulation tasks, one to fold a napkin in 3 folds and the other to unfold and put a tablecloth, following a recent benchmark (Garcia-Camacho et al. 2020). We performed an experiment with 8 subjects that wear a gripper to reduce the number of possible grasp types and dexterity ability. Finally, we show how the CLoM Graph can be easly converted into PDDL and feed into a solver to plan a cloth manipulation task.

The proposed scene representation and the CloM Graph is also motivated to potentially provide explainability to the decision-making processes, in line with the trustworthy AI from the EU guidelines. As opposed to opaque end-to-end deep learning methods (Yang et al. 2016; Tanaka, Arnold, and Yamazaki 2018), latent space variables (Lippi et al. 2020) that are difficult to interpret, or learned latent dynamic models from large amounts of random samples (Yan et al. 2020) that produce plans that are difficult to explain to a human, our CloM Graph provides a framework that is designed to provide both semantic explanations by construction, high-level planning as well as low-level building blocks to plan a task and execute it.

Related work

Task planning understood as a decision-making module that evaluates different strategies and chooses the optimal plan has been quite unexplored in cloth manipulation. Seminal literature on cloth manipulation was more focused on motion planning given a task plan (Cusumano-Towner et al. 2011; Doumanoglou et al. 2016). For cloths already flat on a table, simplified planar polygonal representations were used in (Miller et al. 2011; Doumanoglou et al. 2016; Li et al. 2015) or even simpler 1-dimensional ones in (Petrík et al. 2016) for rectangular clothes. For grasping hanging clothes, contours were used in (Triantafyllou et al. 2016).

Recently, more general literature has focused on deep learning approaches where the scene is represented as RGB-D images and the system learns the mapping between an image with an action and a resulting image, where the action is modelled as the pick-up point pixel coordinates and a direction of displacement (Hoque et al. 2020; Seita et al. 2019; Yan et al. 2020; Jangir, Alenya, and Torras 2020). In (Matas, James, and Davison 2018) they apply reinforcement learning, where the state is represented by an RGB image plus the robot arm joints and grippers state. All these works are trained in simulation but achieve acceptable sim-to-real results. In (Yang et al. 2016) they use a similar approach by feeding directly the RGB image and robot arm joints to a neuronal network that is trained with teleoperated real robot data.

A few works do task planning using similar approaches. In (Tanaka, Arnold, and Yamazaki 2018) they use deep learning to obtain mappings between image states and sequences of simple actions. The method is general but only achieves very simple plans due to the large amounts of data needed, that are in simulation but augmented with large amounts of real robot data. The work in (Lippi et al. 2020) is, up to our knowledge, the only that considered the importance of building a graph of scene states to enable task planning. They build a graph in latent space where each node is a set of RGB images related by just perturbations that is linked to another node if it can be obtained through the application of a simple action, modelled as pick-up point and release point in pixel coordinates. The system is trained by demonstrating the linking actions with a real robot.

All these works assume the basic scene state is the cloth when is not touched by the robot. Instead, in our approach, every re-grasp, contact with the environment or change in cloth configuration triggers a new segment in the graph. 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 (Zoliner et al. 2005). To the best of our knowledge, no work has been able to learn from videos of human demonstrations.

In (Jia et al. 2019) they do imitation learning in robot-human collaboration tasks. They assume the scene is the RGB-D image and the N coordinates of the points where the cloth is grasped, and define the action as the destination location of the grasped points. In this case, no re-grasp or release is considered.

High-level planning has been tackled in the context of robot-assisted dressing (Canal et al. 2018; Kapusta et al. 2019), but without addressing the cloth representation issue and minimizing the part of cloth manipulation by assuming pre-grasped garments.

In our previous work (Borràs, Alenyà, and Torras 2020) we introduced a framework to describe textile grasps based on the geometry of the prehension agents, including extrinsic geometries from the environment. In this paper, we use that notation to identify the grasp but we use additional information to define the scene state.

(a) (b) (c) (d)

Figure 2: The geometries of prehension are points (P), lines (L) and planes (Π). (a) Double pinch grasp. (b) A double pinch with the additional extrinsic contact of the table, denoted with an "e" subscript. (c) A double line-plane grasp (d) A combination of grasps of the hands against the table.

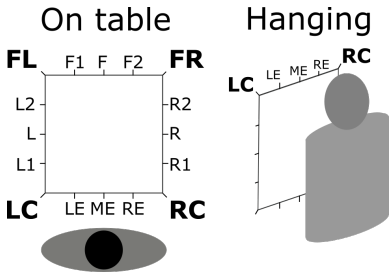


Figure 3: Location of grasp points with respect to subject. Any interior point is labeled I.

A generic state-and-transition definition

To recognize and understand a manipulation action, it is necessary to interpret the states of a scene at each time-step. This is a difficult problem and our approach is to define a simplified representation of a scene in a way that can be recognized by a robot and that allows executing the next action.

We propose to define a *state* as a tuple

$$\mathbf{S} =< GT, GL, CC > \tag{1}$$

where

- GT is the grasp type,
- GL are the location of the grasp with respect to the cloth, and
- CC is the cloth configuration.

Then, we define a *manipulation primitive* as the triple

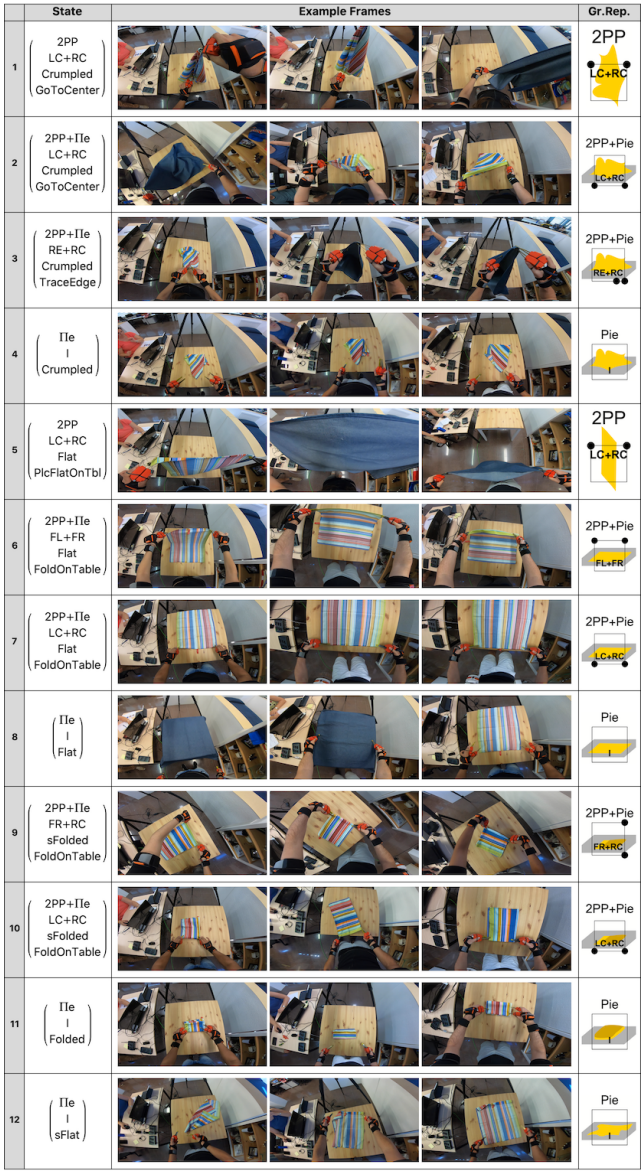
$$< \mathbf{S}_o, \mathbf{S}_d, M > \tag{2}$$

where

- $\mathbf{S}_o$ and $\mathbf{S}_d$ are the origin and destination states, and
- M is a semantic label of the action primitive the subject is performing.

The definition of the grasp type GT is based on the cloth grasp framework and taxonomy introduced in our previous work (Borràs, Alenyà, and Torras 2020). In this framework, each grasp is defined by the geometries of the two virtual fingers that apply opposing forces. A partial glimpse of the grasp framework is provided in Fig. 2. A very important feature is that our grasp framework considers elements in the environment as extrinsic contact geometries and, therefore, it explicitly models environmental contact interactions. Thus, all cloth states realize a grasp, as when there is no contact with the subject, the cloth lays on a table, corresponding to a non-prehensile Π_e grasp.

Table 1: Example frames by scene state



All frames, including frames from additional states, can be found in the paper website [1].

Regarding the grasp location GL, we have defined a set of labels to describe the approximate locations of the grasping points on a given rectangular cloth, shown in Fig. 3, corresponding to coordinates in a 2D cloth reference. Note that a similar notation could be used for other shaped garments along the silhouette. Locations are encoded with respect to the subject grasping hands, i.e., left corner (LC) refers to the corner closest to the subject's left hand, and right corner (RC) likewise, up to rotations of 45º. The two farthest corners are labelled far left (FL) and far right (FR). When the cloth is hanging, the right and left corners are the top ones (closer to subject hands). This means that for certain state transitions we may get a swap of labels for the same points. For instance, when placing a cloth flat on a table, and

then folding it without releasing it, the labelling swaps from (LC+RC) to (FL+FR) after the table contact has been added. See the next section for more details and examples. This notation is used regardless of the cloth configuration. Therefore, when the cloth is folded, each corner contains several layers of fabric. If only the top layer is grasped, it is logged with the subscript RC_1. If no subscript is used, it is assumed that the subject is grasping all the layers. Although this label represents a coarse grid on the border of the cloth, the associated manipulations that a robot may do depending on these locations does not require more precision, as only the concept is important for the decision process. For the execution phase, additional information could be added regarding the location of the robot grippers, cloth corners and edges.

Regarding the configuration of the cloth, CC, it is well known that the configuration state of a textile is infinite-dimensional. That, together with the high number of self-occlusions that occur when manipulating clothes, makes cloth state estimation a difficult problem. The high complexity of its full solution has been bypassed in the past by just looking for task-oriented features, such as adequate and accessible grasping points, e.g., shirt collars for hanging (Ramisa et al. 2016) or towel corners for folding (Maitin-Shepard et al. 2010). Increasingly, it becomes clearer that we need simplified representations, specially regarding deformable objects, as stated in specialized surveys on the topic (Smith et al. 2012; Yin, Varava, and Kragic 2021). We have defined only 5 categories of simplified cloth configurations:

$$\{Crumpled, Flat, Folded, Semi-Folded, Semi-Flat\}$$

This is a very short list of states, but in combination with the grasping information and the interaction with the environment, we found it reduced the variability enough inside one same state. This can be seen in Table 1, where we show examples of frames corresponding to segments identified in our experiments. For instance, the crumpled state, that appears in rows 1-4, can have many configurations. However, whether it is in contact with the table or not, or grasped by corners or not reduces the possible configurations to very similar shapes inside each state. This is not true for the case where it is not grasped, like in row 4. In this case, there is the possibility of enriching each category with different descriptors such as (Ramisa et al. 2013) to measure the amount of deformation or the number of visible edges and corners using methods such as (Qian et al. 2020) or (Liu et al. 2016). This is beyond the scope of this work, and we assume here that at least a corner is visible.

The other state that may seem ambiguous is the semi-folded, rows 9 and 10 in Table 1, as we are not considering how many folds have been done. Indeed, we can see in the table how cloths with different number of folds appear under the same state. However, we propose to only identify the final state of fold (row 11), as all partial folds afford the same kind of action, that of continuing folding until you are done. The semi-flat state (row 12) is important as it can trigger a flattening action, but it has been purposely ignored in the data for simplicity, as will be explained in the next section.

Finally, regarding the motion semantic label M, we define a set of labels related to the action the subject is performing from that initial state until the following one, like for instance, "Place flat on table", "Fold on table" or "Trace edge". Semantic labels are useful for high-level planning and scene understanding, and can be linked to low-level parameters like motion primitives or other trajectory representations. They can also be seen in Table 1 because for the data we have collected, all states where the cloth is grasped trigger the same action although they may finish in different states, as can be seen in the graph representations.

Figure 4: Experimental setup. (Top) The subject wears a motion tracking suit, a GoPro camera mounted on the head and we also take a Kinect screenshot of the final result, although the latter is not used in this paper. (Bottom) Wearable point-point gripper used in the experiments.

The proposed state and transition definition induces a segmentation of manipulation tasks at each change of scene state. The state changes at each re-grasp, which in our grasp framework, this includes changes in contacts with the environment. In addition, there is also a change of state when the grasp locations vary (like in a "Trace Edge") or when the cloth configurations changes (like in "GoToCenter", also known as Unfold in the air).

Experimental setup and data collection

We tested a total of 8 subjects wearing a GoPro camera fixed at their forehead (Fig 4). Additionally, subjects wore a motion data suit (XSens), but we don't use these data for the current paper. The experiment included several cloth manipulation tasks, but for the scope of this paper, we focus on the task of folding a napkin with 3 folds on the table and the unfolding to put a tablecloth. We asked the subjects to wear a simple gripper, shown at the bottom of Fig 4, to reduce their manipulation dexterity to one closer to that of the robot. Subjects were allowed to train with the grippers, executing the tasks three to four times before starting the recordings.

When it comes to cloth manipulation, human experiments provide us with a lot of useful information regarding the va-

riety of strategies to accomplish a task, that is not observed in robot cloth manipulation demonstrations, as analyzed in (Borràs, Alenyà, and Torras 2020). Therefore, learning state sequences from humans will provide us with a much richer graph regarding alternative strategies, and we will be able to learn new manipulation approaches for robots. However, there is a trade-off between obtaining a great diversity of strategies and sparsity on the obtained data derived from particular ways subjects perform one same task. This is specially true when it comes to cloth manipulation that almost every subject has its own tricks to fold their clothes. For this reason, we instructed the subjects to perform a very specific task (fold on the table, not in the air, and in 3 folds, and unfold the tablecloth to directly place it on the table). Despite these indications, we obtained a lot of variability, sometimes even between the trials of one same subject. However, some strategies have been used consistently by most of the subjects.

From the data collected, we have manually labelled the videos at each change of state, associating a motion semantic label to each transition depending on the action that was done, following the proposed representation. We purposely ignored any manipulation that corrected a mistake, or that relocated the cloth on the table, just to simplify the data. Examples of the labels and their corresponding graphic state representations can be seen in Table 1. The labels include timestamps at each change of state, providing the segmentation of the data and the sequence of states.

Cloth manipulations graph

Thanks to the proposed representation, and extracting the sequences of state and transitions of the labelled video data, we can generate a graph where each node is a scene state, and the edges represent the transition action.

To generate the graph, for each trial we defined an edge for each state change, and we represented it symbolically using the formulation introduced in Section "A generic state-and-transition definition", where each initial and destination states are the initial and end node of the graph edge, and the motion semantic value is the edge label. We then identify common nodes and common edges, defining the graph with all the distinct vertices and edges that have appeared, counting their multiplicity.

To simplify the data, we have removed some left and right distinctions. For instance, a single corner grasped is the same irrespective of whether it is the left or right corner, grasped with the left or right hand. We also assume two grasped points on the same cloth edge are the same regardless if they are on the right or left side. All these simplifications are described in the additional material [1].

Using all the data collected, we obtain a graph with 32 nodes and 65 edges, but many of them appear a single time in our data. If we require each edge to appear at least 2 times in the data, the graph is reduced to 18 nodes with 27 edges. The reduced graph is shown in Fig. 5. The complete graph can't be included in the paper for space reasons, but you can find it on the provided website. The CloM Graph of the task of unfolding and putting the tablecloth can also be found in the website, in this case, the simplified one has 12 states and 15 transitions, while the full graph has 17 states and 32 transitions, meaning that this task is much less complex than the previous one. As the two tasks are inverse one of the other, only one transition is common in both graphs, the one of "Place flat on table" from the central state (2PP, RL+LC, Flat) to the (2PP+Π_e, RL+LC ,Flat) that appears 21 times for the tablecloth task and 20 for the folding task.

We performed a total of 24 trials, meaning the maximum times one primitive can appear repeated in the data is 24. Despite the diversity of strategies displayed by the subjects there are some transitions that consistently appear. We plotted in red the transitions that appear in at least half of the total capacity (12 times) and, in orange, the ones that appear 6 times or more. We can see that the weakest flow in the graph is in the transition from Fig. 5-a to Fig. 5-b. That is because there is a great variety of manipulations to find the two corners, that can be appreciated in the full graph. Once the corners are grasped, the primitives to unfold in the air become less sparse (Fig. 5-c). The bottom state at the column (a), the (PP, RC, Crumpled) state, is reached by several edges with a multiplicity 1 that don't appear, but can be seen in the full graph.

Planning

As it has been presented, the CloM Graph allows to easily establish the sequence of states and actions necessary to execute a cloth manipulation task. In this section, we show that the definition of the CloM Graph is adequate and convenient for generating a planning domain that can, for example, potentially be used in decision making to solve a task with a robot manipulator. Given an instance of a CloM Graph, we can translate the presented representation into a classical planning problem, synthesizing it in the STRIPS language, which is a subset of the Planning Domain Definition Language (PDDL).

A classical planning problem is defined by a 4-tuple $P =< F, A, I, G >$, where F is the set of fluents with binary valuation, A is the set of actions, and I and G are the initial and goal states of the problem. On the one hand, fluents F are propositional variables that serve to describe the states. On the other hand, actions A are defined by a set of preconditions and a set of positive and negative effects (both described by a set of fluents), meaning that an action is only applicable when the preconditions hold in that state. As an important implementation detail, we observe that in our representation the states (i.e. nodes of the graph) are defined by the tuple **S** in Equation 1, being able to consider each parameter of which is composed as a propositional variable f, where $f \in F$. However, as we rely on binary fluents when we ground GT, GL and CC to a particular value the rest of fluents are negated.

Furthermore, our actions (i.e. transitions in the graph) are the *manipulation primitives* composed of the tuple $< S_o, S_d, M >$, in which S_o include the preconditions of the action, S_d the effects of the action, and M is the semantic label of the action. Finally, I and G represent the graph states

[1] http://www.iri.upc.edu/groups/perception/#PlanningCloMGraph

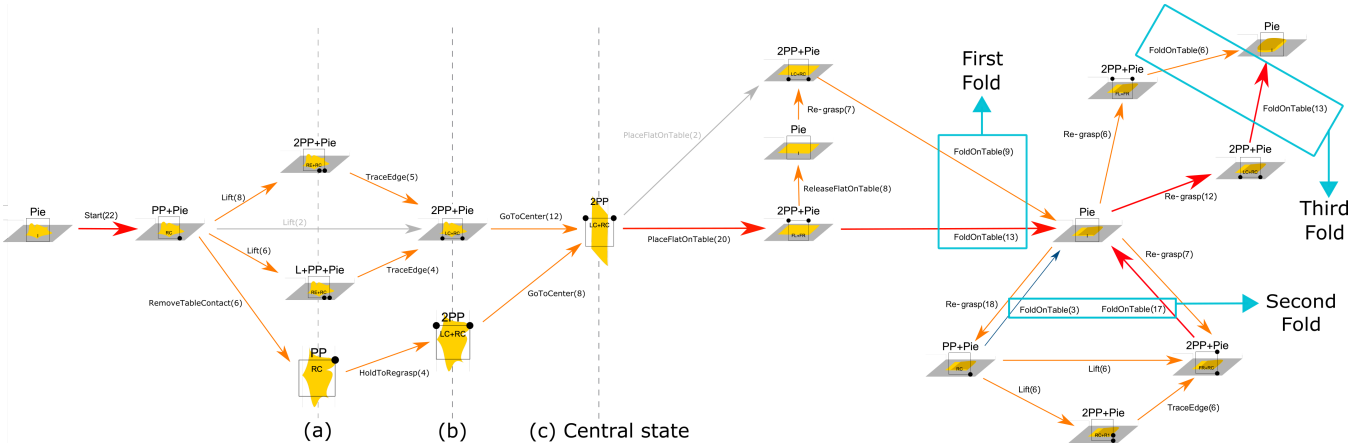


Figure 5: Reduced CloM graph obtained by requiring each edge to be observed at least 3 times in the data. We can clearly see the different phases of the task, from the crumpled on the table phase on the left, to the central hanging part of the manipulation (central state), and then the semi-folded states on the table during the first, second and third folds, located to the right. The label of each edge consist of the semantic name of the primitive and the number of times it appears in the data (in parenthesis).

used to define the problem, whose solution presents a sequence of actions, whose execution induces a state sequence to arrive from state I to state G. Having this definition, any cloth manipulation task can be described as a deterministic problem P using the representation presented in Section "A generic state-and-transition definition". To validate this approach we translated into STRIPS the CloM Graphs of the cloth manipulation tasks of folding a napkin in 3 folds and spreading a tablecloth. The PDDL domain and the problem files for both tasks, as well as the code to generate the LISP-like files of any cloth manipulation task from the CloM Graph can be found on the paper's website [1].

To instantiate the propositional variables F that are necessary in the domain, it is necessary to identify the different grasp types (GT), grasp locations (GL) and cloth configurations (CC) that appear during the execution of the task. In the example of folding a napkin, we have a total of 6 different Grasp Types, 7 combinations of Grasp Locations and 4 Cloth Configurations:

- PP, 2PP, Π_e, PP+Pie, 2PP+Π_e, L+PP+Π_e
- I, RC, LC+RC, FL+FR, FR+F2, FR+RC, RE+RC
- Crumpled, Flat, sFolded, Folded

In addition, actions are created by instantiating as preconditions the specific GT, GL and CC of the origin state of the action and specifying as effects the fluents that had changed its value (e.g. if an action only changes the CC from Flat to sFolded, then $eff = sFolded \wedge \neg Flat$). The domain will be composed of a total number of actions equal to the number of transition in the CloM Graph.

By construction, the result of the domain given a problem whose initial state I is the first state of the graph (Pie, -, Crumpled) and the goal G is the last one (Pie, -, Folded) will provide a graph structure as the one in Fig. 5, being able to directly obtain the plan solution given any two states of the graph.

Now that we have the domain model, a solver (Fast-Forward in our example) can be used to solve a particular problem given both the initial I and goal states G. We expect the solution of the solver to match the corresponding fragment of the graph. As an example, we consider the fragment of the task of folding a napkin shown in Fig. 6. As we can see in the figure, there exist four different paths, highlighted in colors, to arrive from the initial state S1 to the goal state S7. In classical planning, it is also usual to define some cost functions that serve to express a measure of plan quality that the planner should try to optimize, but obtaining the optimal path is not in the scope of this work, but to present a way to obtain the possible actions and state sequences of a task. Therefore, all the existing outcomes of the given problem are explored by manually increasing the cost of some actions, so the planner is forced to produce successively the different solutions. The resulting plans for the PDDL problem of the fragment in Fig. 6 are presented in Fig. 7, with their corresponding state sequences. Observe the completeness of the approach, as the solver was able to produce all the four expected plans, which correspond to the paths on the graph.

We argue that the costs should be computed related to the particular robotic setup, capabilities, and tasks. In the context of cloth manipulation, costs could be defined for example to minimize the total number of actions to execute or the running time. In our example, the green path in Fig. 6 would be the optimal option. Also, cost functions can be defined to select best paths according to the restrictions of the robotic embodiment (e.g. use tactile sensors to perform the edge tracing action, then the red and yellow paths would be the best options), or of the hardware (e.g. gripper that allows to perform linear grasps).

Discussion

The proposed state representation simplifies the perceptual information that needs to be acquired. Observe that the com-

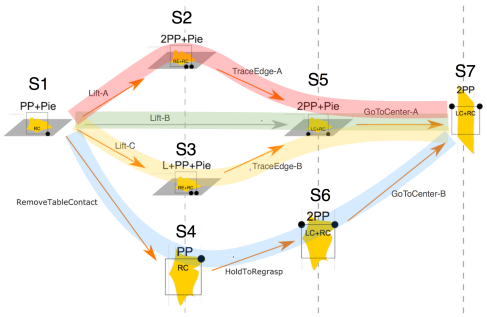


Figure 6: Fragment from the CloM Graph of folding a napkin. PDDL problem defined with initial state I=S1 to goal state G=S7.

```
ff: found legal plan as follows
step    0: LIFT-B
        1: GOTOCENTER-A

S1→S5→S7
```

```
ff: found legal plan as follows
step    0: REMOVETABLECONTACT
        1: HOLDTOREGRASP
        2: GOTOCENTER-B

S1→S4→S6→S7
```

```
ff: found legal plan as follows
step    0: LIFT-C
        1: TRACEEDGE-B
        2: GOTOCENTER-A

S1→S3→S5→S7
```

```
ff: found legal plan as follows
step    0: LIFT-A
        1: TRACEEDGE-A
        2: GOTOCENTER-A

S1→S2→S5→S7
```

Figure 7: Plan outcomes for the given problem S1 to S7.

putation of the graph requires only a high-level segmentation of the states. We have recently demonstrated that these assumptions are realistic and that the required perceptions are feasible (Tzelepis et al. 2022). However, note that the effective execution of every transition requires much more detailed perception, in the form of cloth part recognition and pose estimation (Ramisa et al. 2016) and grasping point detection (Corona et al. 2018; Qian et al. 2020). We believe that is with the integration of this methods, together with the definition of planning domains, that we can enable high-level planning and decision making of complex cloth manipulation tasks.

There is an open question remaining regarding whether additional information would be necessary to enrich the representation of cloth configuration in the most complex cases. For example, in the case of a crumpled cloth on a table, the presence of visible corners or not, changes the grasping action necessary to start the task (e.g. grasp directly the corner or try to reveal a hidden corner by grasping the cloth by any point and release it). This can be done by including descriptors such as (Ramisa et al. 2013) or by using methods to detect cloth features (Qian et al. 2020) and including the number of visible corners and edges in the category of cloth configuration.

To build the CLoM graph the proposed granularity for segmentation is much thinner than other works like (Lippi et al. 2020), where only states with the cloth on the table are considered. This is done with the motivation of obtaining simple motion primitives to facilitate re-usability. Also, we believe that this segmentation is also relevant for benchmark purposes, to represent the complexity of a task and identify different evaluation segments.

Another motivation behind our approach is to enable explainable reasoning at the manipulation level as well as learning a dynamic movement primitive (DMP) for each re-grasp strategy (not necessary from human motion data), which is also associated with its preconditions and effects. The resulting DMPs can be used for task planning (Canal et al. 2018), and potentially for explainability purposes as well, since the learning process makes explicit the conditions that enable to execute the primitive and the expected outcomes. We envisage the CloM Graph as a common ground representation where information at the different robotic levels (planning, perception and execution) can be stored.

Conclusions

We have introduced a compact and generic representation of states of a manipulation task in the context of cloth manipulation. The representations are vast simplifications of the complexity of a cloth manipulation state, but we showed how they are enough to segment a manipulation task into relevant and coherent manipulation primitives. In addition, from the sequences of states and transitions, we have defined the CloM Graph that encodes the diversity of strategies to accomplish the task. We have shown two examples of common cloth manipulation tasks for which the CloM Graph is learned from an experiment with 8 human subjects. Learning from human demonstrations allows to identify manipulation primitives not used so far by robots that could be especially handy for the versatile manipulation of clothing items. In addition, we have defined full domain models of these two tasks to enable PDDL planning and demonstrate that the proposed representation is compatible for describing task planners that allows to enable high-level planning of complete cloth manipulation tasks.

The CloM Graph we have proposed complies with the desideratum that ”low-complexity representations for the deformable objects should be the objective” (Smith et al. 2012). This manipulation-oriented representation would permit probabilistic planning of actions to ensure reaching the desired cloth configuration without requiring high accuracy in perception nor searching in high-dimensional configuration spaces. In addition, our encoding of manipulation tasks facilitates the definition of metrics and measures of complexity of a given strategy, which is very useful to define benchmark tasks with increasing complexity.

In future research, we will work towards the state recognition and the definition of the motion primitives performing transitions between states. Additionally, this work will lead to a database of labelled video data synchronized with motion data of different cloth manipulation tasks, which could be of great utility for the whole manipulation community working on highly deformable objects.

Acknowledgments

The research leading to these results receives funding from the European Research Council (ERC) from

the European Union Horizon 2020 Programme under grant agreement no. 741930 (CLOTHILDE: CLOTH manIpulation Learning from DEmonstrations); by MCIN/ AEI /10.13039/501100011033, Spain, under the project CHLOE-GRAPH (PID2020-119244GB-I00); by MCIN/ AEI /10.13039/501100011033, Spain, and by the "European Union NextGenerationEU/PRTR, Spain, under the project COHERENT (PCI2020-120718-2); and by the European Commission- NextGenerationEU, through CSIC's Thematic Platforms (PTI+ Neuro-Aging).

References

Bersch, C.; Pitzer, B.; and Kammel, S. 2011. Bimanual robotic cloth manipulation for laundry folding. In *IEEE/RSJ Int. Conf. on Intel. Rob. and Sys.*, 1413–1419.

Borràs, J.; Alenyà, G.; and Torras, C. 2020. A Grasping-Centered Analysis for Cloth Manipulation. *IEEE Trans. on Rob.*, 36(3): 924–936.

Canal, G.; Pignat, E.; Alenyà, G.; Calinon, S.; and Torras, C. 2018. Joining high-level symbolic planning with low-level motion primitives in adaptive HRI: application to dressing assistance. In *IEEE Int. Conf. on Rob. and Autom.*, 1–9.

Corona, E.; Alenyà, G.; Gabas, A.; and Torras, C. 2018. Active garment recognition and target grasping point detection using deep learning. *Pattern Recognition*, 74: 629–641.

Cusumano-Towner, M.; Singh, A.; Miller, S.; O'Brien, J. F.; and Abbeel, P. 2011. Bringing clothing into desired configurations with limited perception. In *IEEE Int. Conf. on Rob. and Autom.*, 3893–3900.

Doumanoglou, A.; Stria, J.; Peleka, G.; Mariolis, I.; Petrik, V.; Kargakos, A.; Wagner, L.; Hlavac, V.; Kim, T.-K.; and Malassiotis, S. 2016. Folding Clothes Autonomously: A Complete Pipeline. *IEEE Trans. on Rob.*, 32(6): 1461–1478.

Garcia-Camacho, I.; Lippi, M.; Welle, M. C.; Yin, H.; Antonova, R.; Varava, A.; Borras, J.; Torras, C.; Marino, A.; Alenyà, G.; et al. 2020. Benchmarking bimanual cloth manipulation. *IEEE Rob. and Autom. Letters*, 5(2): 1111–1118.

Hoque, R.; Seita, D.; Balakrishna, A.; Ganapathi, A.; Tanwani, A. K.; Jamali, N.; Yamane, K.; Iba, S.; and Goldberg, K. 2020. Visuo-Spatial Foresight for Multi-Step, Multi-Task Fabric Manipulation. In *Rob.: Sci. and Sys. (RSS)*.

Jangir, R.; Alenya, G.; and Torras, C. 2020. Dynamic Cloth Manipulation with Deep Reinforcement Learning. In *IEEE Int. Conf. on Rob. and Autom.*, 4630–4636.

Jia, B.; Pan, Z.; Hu, Z.; Pan, J.; and Manocha, D. 2019. Cloth Manipulation Using Random-Forest-Based Imitation Learning. *IEEE Robotics and Automation Letters*, 4(2): 2086–2093.

Kapusta, A.; Erickson, Z.; Clever, H. M.; Yu, W.; Liu, C. K.; Turk, G.; and Kemp, C. C. 2019. Personalized collaborative plans for robot-assisted dressing via optimization and simulation. *Auto. Rob.*, 43(8): 2183–2207.

Li, Y.; Yue, Y.; Xu, D.; Grinspun, E.; and Allen, P. K. 2015. Folding deformable objects using predictive simulation and trajectory optimization. In *IEEE/RSJ Int. Conf. on Intel. Rob. and Sys.*, 6000–6006.

Lippi, M.; Poklukar, P.; Welle, M. C.; Varava, A.; Yin, H.; Marino, A.; and Kragic, D. 2020. Latent Space Roadmap for Visual Action Planning of Deformable and Rigid Object Manipulation. *IEEE/RSJ Int. Conf. on Intel. Rob. and Syst.*, 5619–5626.

Liu, Z.; Yan, S.; Luo, P.; Wang, X.; and Tang, X. 2016. Fashion landmark detection in the wild. In *European Conference on Computer Vision*, 229–245. Springer.

Maitin-Shepard, J.; Cusumano-Towner, M.; Lei, J.; and Abbeel, P. 2010. Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *IEEE Int. Conf. on Rob. and Autom.*, 2308–2315.

Matas, J.; James, S.; and Davison, A. J. 2018. Sim-to-real reinforcement learning for deformable object manipulation. In *Proc. of Conf. on Rob. Learning*.

Miller, S.; Fritz, M.; Darrell, T.; and Abbeel, P. 2011. Parametrized shape models for clothing. In *IEEE Int. Conf. on Rob. and Autom.*, 4861–4868.

Miller, S.; Van Den Berg, J.; Fritz, M.; Darrell, T.; Goldberg, K.; and Abbeel, P. 2012. A geometric approach to robotic laundry folding. *Int. J. of Rob. Res.*, 31(2): 249–267.

Petrík, V.; Smutnỳ, V.; Krsek, P.; and Hlaváč, V. 2016. Physics-based model of a rectangular garment for robotic folding. In *IEEE/RSJ Int. Conf. on Intel. Rob. and Syst.*, 951–956.

Qian, J.; Weng, T.; Zhang, L.; Okorn, B.; and Held, D. 2020. Cloth Region Segmentation for Robust Grasp Selection. In *IEEE/RSJ Int. Conf. on Intel. Rob. and Syst.*, 9553–9560.

Ramisa, A.; Alenya, G.; Moreno-Noguer, F.; and Torras, C. 2013. Finddd: A fast 3d descriptor to characterize textiles for robot manipulation. In *IEEE/RSJ Int. Conf. on Intel. Rob. and Syst.*, 824–830.

Ramisa, A.; Alenyà, G.; Moreno-Noguer, F.; and Torras, C. 2016. A 3D descriptor to detect task-oriented grasping points in clothing. *Pattern Recognition*, 60: 936–948.

Sanchez, J.; Corrales, J.-A.; Bouzgarrou, B.-C.; and Mezouar, Y. 2018. Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey. *Int. J. of Rob. Res.*, 37(7): 688–716.

Seita, D.; Jamali, N.; Laskey, M.; Tanwani, A. K.; Berenstein, R.; Baskaran, P.; Iba, S.; Canny, J.; and Goldberg, K. 2019. Deep transfer learning of pick points on fabric for robot bed-making. In *International Symposium on Robotics Research (ISRR)*.

Smith, C.; Karayiannidis, Y.; Nalpantidis, L.; Gratal, X.; Qi, P.; Dimarogonas, D. V.; and Kragic, D. 2012. Dual arm manipulation—A survey. *Rob. and Auton. Sys.*, 60(10): 1340–1353.

Tanaka, D.; Arnold, S.; and Yamazaki, K. 2018. EMD Net: An encode–manipulate–decode network for cloth manipulation. *IEEE Rob. and Autom. Letters*, 3(3): 1771–1778.

Triantafyllou, D.; Mariolis, I.; Kargakos, A.; Malassiotis, S.; and Aspragathos, N. 2016. A geometric approach to robotic unfolding of garments. *Robotics and Autonomous Systems*, 75: 233–243.

Tzelepis, G.; Aksoy, E. E.; Borràs, J.; and Alenyà, G. 2022. Semantic State Estimation in Cloth Manipulation Tasks. *arXiv preprint arXiv:2203.11647*.

Verleysen, A.; Biondina, M.; and Wyffels, F. 2020. Video dataset of human demonstrations of folding clothing for robotic folding. *Int. J. of Rob. Res.*, 39(9): 1031–1036.

Yan, W.; Vangipuram, A.; Abbeel, P.; and Pinto, L. 2020. Learning Predictive Representations for Deformable Objects Using Contrastive Estimation. *arXiv preprint arXiv:2003.05436*.

Yang, P.-C.; Sasaki, K.; Suzuki, K.; Kase, K.; Sugano, S.; and Ogata, T. 2016. Repeatable folding task by humanoid robot worker using deep learning. *IEEE Rob. and Autom. Letters*, 2(2): 397–403.

Yin, H.; Varava, A.; and Kragic, D. 2021. Modeling, learning, perception, and control methods for deformable object manipulation. *Science Robotics*, 6(54): eabd8803.

Zoliner, R.; Pardowitz, M.; Knoop, S.; and Dillmann, R. 2005. Towards cognitive robots: Building hierarchical task representations of manipulations from human demonstration. In *IEEE Int. Conf. On Rob. and Auto.*, 1535–1540.