Towards vision-based manipulation of plastic materials

Andrea Cherubini, Jürgen Leitner, Valerio Ortenzi, Peter I Corke

Towards vision-based manipulation of plastic materials

Andrea Cherubini*, Jürgen Leitner†, Valerio Ortenzi† and Peter Corke†

***Abstract*— This paper represents a step towards vision-based manipulation of plastic materials. Manipulating deformable objects is made challenging by: 1) the absence of a model for the object deformation, 2) the inherent difficulty of visual tracking of deformable objects, 3) the difficulty in defining a visual error and 4) the difficulty in generating control inputs to minimise the visual error. We propose a novel representation of the task of manipulating deformable objects. In this preliminary case study, the shaping of kinetic sand, we assume a finite set of actions: *pushing, tapping* and *incising*. We consider that these action types affect only a subset of the state, *i.e.*, their effect does not affect the entire state of the system (*specialized actions*). We report the results of a user study to validate these hypotheses and release the recorded dataset. The actions (pushing, tapping and incising) are clearly adopted during the task, although it is clear that 1) participants use also mixed actions and 2) actions' effects can marginally affect the entire state, requesting a relaxation of our *specialized actions* hypothesis. Moreover, we compute task errors and corresponding control inputs (in the image space) using image processing. Finally, we show how machine learning can be applied to infer the mapping from error to action on the data extracted from the user study.**

***Index Terms*— Manipulation, visual servoing, human studies, learning.**

I. INTRODUCTION

Robots have been manipulating a vast variety of objects in manufacturing environments for years. This relies on simple pre-programmed actions, on exactly known objects (whose physical model is known) and controlled environments. However, in most of cases, rigid objects are being manipulated. Nonetheless, the manipulation of soft materials is present in many everyday tasks: in domestic environments, like making a dough or folding clothes; in industrial environments, like fixing cables; and in medical environments, *e.g.*, for surgery or physiotherapy.

Many difficulties arise when manipulating soft materials. Firstly, the object deformation model (involving elasticity or plasticity) should be known in order to derive the robot control inputs needed to change its shape. Ideally, this model should be derived online, while manipulating the object, with a simultaneous estimation and control approach, as commonly done in active perception. Hence the robot senses, and particularly vision, will be indispensable. This leads to a second major difficulty: visual tracking of deformable objects. In fact, most current visual object tracking algorithms rely on rigidity, an assumption that is not valid here. Only very few approaches take into account either plasticity or elasticity, thus performing well only on rigid objects. A third challenge consists in generating control inputs that servo the deformable material to the desired shape.

Fig. 1. Kinetic sand represents a good example of plastic material whose dynamic model is unknown. The four pictures show different shapes formed by the participants of our user study.

In this work, we focus on defining the control problem of manipulating a deformable object whose physical model is unknown. Specifically on *molding plastic materials*, *i.e.*, materials that undergo non-reversible shape changes in response to force application. We assume that a finite set of action types is available, such that each action affects only a subset of the system state (*specialized actions*).

We carry out a study, where participants are asked to mold kinetic sand[1] into desired shapes (see Fig. 1), while being recorded on video. The main contributions herein are:

- A framework for the manipulation of plastic objects, including a preliminary task modeling step, followed by a sequencing algorithm. We believe that both steps can be generalized to many other applications where tasks are complex and possibly coupled.
- Results of a "sand manipulation user study" where the analysis of the participants' actions suggests that 1) a finite set of action types is used for the task and 2) these actions are marginally coupled, so the assumption of *specialized actions* has to be relaxed.
- A dataset to enable comparative investigations and benchmarking.
- We show how learning can be used to bridge the gap represented by the lack of a model to choose the action that reduces the visual task error. In particular, we propose to use neural networks and show the results on our dataset.

*CNRS-UM LIRMM, Interactive Digital Human group, 161 Rue Ada, 34090 Montpellier, France cherubini@lirmm.fr.

†ARC Centre of Excellence for Robotic Vision, Queensland University of Technology, Brisbane QLD 4001, Australia. http://www.roboticvision.org

[1] https://en.wikipedia.org/wiki/Kinetic_Sand

II. RELATED WORK

One of the first works on deformable object manipulation (specifically, assembly) addresses insertion of a beam into a hole [1], by properly designing the robot tool motion trajectory. Planning is also used in [2], to compute paths among minimal energy configurations, for deforming flexible wires, subject to manipulation constraints. More recently [3] targeted the assembly of an o-ring into a cylinder, using a heuristic approach to compute key postures of the robot arms. Dynamic Movement Primitives are used in [4] for clothing assistance, while a control of both motion and deformation of soft objects is proposed in [5]. A PID control law with and without the deformable object model is presented in [6]; differently, planning and control are both used in the approach described in [7]. Diminishing rigidity is used in [8] to quickly compute an approximation of the Jacobian of the deformable object. Excessive stretching is also avoided.

While the cited works rely on motion planning, other researchers propose to rely more on sensor feedback to manipulate deformable objects, as we do here. In [9], stereovision was exploited, to insert a flexible wire into a hole. Vision is also used in [10] for assembling a rubber belt and fixed pulleys. In [11], [12], compliant objects are actively deformed using a novel visual servoing scheme that explicitly deals with elastic deformations, by adapting online the interaction matrix relating tool velocities and optical flow. The controller is model-free, but focuses mainly on shape control, *i.e.*, on manipulating the object to a desired configuration, without dealing with its global deformation over a long time window. Dissimilarly, force control is used to design a strategy for dual manipulation of a flexible metal sheet in [13]. Another approach for modeling the dynamics of two manipulators handling deformable objects is proposed in [14] by dividing the closed chain into two subsystems, one flexible for the object, the other rigid for the manipulators. Tactile servoing is used for in-hand manipulation of deformable objects in [15], [16].

Here, we use monocular vision alone, but in contrast with [11], [12]: we consider applications where different actions can affect different characteristics of the material, and we propose to learn such actions by observing humans.

III. METHOD

A. Problem statement

Let us define a discrete-time system with $\mathbf{x}_k \in I\!R^n$ the *state* of the plastic material at time $k \in I\!N$. We assume that $\mathbf{x}_k$ is time-invariant, and depends only on the state at the previous iteration $\mathbf{x}_{k-1}$ and on the external *action* $\mathbf{u}_k$ applied to the material at time k:

$$\mathbf{x}_k = f\left(\mathbf{x}_{k-1}, \mathbf{u}_k\right). \tag{1}$$

We also assume that $\mathbf{x}_k$ can be measured visually at each iteration, from image $\mathcal{I}_k$. Vectors $\mathbf{x}$ and $\mathbf{x}^*$ can embed heterogeneous features of the current and desired shape, provided these are observable in the images. Action $\mathbf{u}$ can also vary in nature and dimension at each iteration, provided it is feasible by the robotic system. Examples of visual features and actions will be given in Subsect. III-C.

We define the global task as that of servoing $\mathbf{x}$ to a *desired* state $\mathbf{x}^*$ (*e.g.*, one of the shapes in Fig. 1). This is equivalent to regulating the error $\mathbf{e}_k$ to zero within a finite number of iterations K

$$\mathbf{e}_k = \mathbf{x}_k - \mathbf{x}^* = f\left(\mathbf{x}_{k-1}, \mathbf{u}_k\right) - \mathbf{x}^* = \mathbf{e}_k\left(\mathbf{x}_{k-1}, \mathbf{u}_k\right). \tag{2}$$

Eq. (2) is a discrete non-linear system with unknown (and hardly derivable) dynamic model. Furthermore, the interdependencies between the components of $\mathbf{x}$ are also unknown, and coupling in the dynamics of such features might occur (*e.g*, different features may be affected by the same action). For this reason, we make two hypotheses in this work.

Hypothesis 1: finite set of action types There exists a finite set of action types $\mathcal{U} = \left\{\mathbf{u}^1, \ldots, \mathbf{u}^N\right\}$, such that the task can be successfully completed by using only actions within the set: $\mathbf{u}_k \in \mathcal{U}$, $\forall k = 1, \ldots, K$. Furthermore, each action type i can be parameterised by a vector $\mathbf{p}$, which can vary, while staying within *that* action type parameter set $\mathcal{P}^i$: $\mathbf{u}^i = \mathbf{u}^i\left(\mathbf{p}\right)$, $\mathbf{p} \in \mathcal{P}^i$.

Hypothesis 2: specialized actions Each action type $\mathbf{u}^i$, $i = 1, \ldots, N$ regulates only a *subset* of error $\mathbf{e}$, that we denote $\overline{\mathbf{e}}^i$, without modifying the rest (*i.e.*, the other components of $\mathbf{e}$, $\underline{\mathbf{e}}^i$).

B. Proposed Approach

We start by breaking $\mathbf{x}$ into N *substates*, each being a vector $\overline{\mathbf{x}}^i$, $i = 1, \ldots, N$ with only some appropriately chosen components of $\mathbf{x}$. The substates should be chosen so that each $\overline{\mathbf{x}}^i$ can be controlled by a specialized action $\mathbf{u}^i$. Each $\overline{\mathbf{x}}^i$ is obtained by orthogonal projection of $\mathbf{x}$ using a constant diagonal binary square matrix of size n, $\mathbf{P}_i$. We also associate $\overline{\mathbf{x}}^i$ to its complementary, $\underline{\mathbf{x}}^i$, and to the corresponding task error, $\overline{\mathbf{e}}^i$

$$\begin{aligned} \overline{\mathbf{x}}^i &= \mathbf{P}_i \mathbf{x}, \\ \underline{\mathbf{x}}^i &= \left(\mathbf{I}_n - \mathbf{P}_i\right) \mathbf{x}, \\ \overline{\mathbf{e}}^i &= \overline{\mathbf{x}}^i - \overline{\mathbf{x}}^{i*} = \mathbf{P}_i f\left(\mathbf{x}_{k-1}, \mathbf{u}_k\right) - \overline{\mathbf{x}}^{i*}. \end{aligned} \tag{3}$$

Having defined all substates and corresponding actions, at each iteration k where the goal is to regulate $\overline{\mathbf{e}}^i$, we apply:

$$\mathbf{u}_k^i = \underset{\mathbf{p} \in \mathcal{P}^i}{\operatorname{argmin}} \left\| \overline{\mathbf{e}}_k^i\left(\mathbf{x}_{k-1}, \mathbf{u}_k^i\left(\mathbf{p}\right)\right) \right\| \tag{4a}$$

$$\text{subject to } \left\| \underline{\mathbf{x}}_k^i - \underline{\mathbf{x}}_{k-1}^i \right\| \leq \underline{\epsilon}^i, \tag{4b}$$

with $\underline{\epsilon}^i \geq 0$. Note that (4a) aims at finding the optimal (minimizing $\overline{\mathbf{e}}^i$) parameters $\mathbf{p}$ for action type $\mathbf{u}^i$, given the current state $\mathbf{x}_{k-1}$. This is done while bounding, via (4b), the action *side effects* on the complementary substate $\underline{\mathbf{x}}^i$. Practically, action type $\mathbf{u}^i$ is determined by substate $\overline{\mathbf{x}}^i$, and constraint (4b) is always verified under *Hypothesis 2*.

To regulate the global task error $\mathbf{e}$, we propose the method shown in **Algorithm 1**.

A qualitative discussion of the convergence of this algorithm follows this line of reasoning: assuming that $\mathbf{u}_k^i$ verifying (4) exists for all k, the error norm will monotonously decrease, since $\mathbf{u}_k^i = \operatorname{argmin} \left\| \overline{\mathbf{e}}_k^i \right\|$. Although this will not

Algorithm 1: Robotic plastic material molding

Input: Desired material state $\mathbf{x}^*$, image $\mathcal{I}$ at each iteration.
Output: Material state $\mathbf{x}$ such that $\|\mathbf{x} - \mathbf{x}^*\| \leq \epsilon$.

1: Initialize: $i = 1$, $k = 1$
2: Measure $\mathbf{x}_0$ from $\mathcal{I}_0$
3: $\mathbf{e}_0 \leftarrow \mathbf{x}_0 - \mathbf{x}^*$
4: **while** $\|\mathbf{e}_{k-1}\| > \epsilon$ **do**
5: **while** $\left\|\overline{\mathbf{e}}^i_{k-1}\right\| > \overline{\epsilon}^i$ **do**
6: $\mathbf{u}^i_k \leftarrow \text{argmin} \left\|\overline{\mathbf{e}}^i_k\right\|$ such that $\left\|\underline{\mathbf{x}}^i_k - \underline{\mathbf{x}}^i_{k-1}\right\| \leq \underline{\epsilon}^i$
7: Apply $\mathbf{u}^i_k$ to control the robot motion
8: $k \leftarrow k + 1$
9: Measure $\mathbf{x}_{k-1}$ from $\mathcal{I}_{k-1}$
10: $\mathbf{e}_{k-1} \leftarrow \mathbf{x}_{k-1} - \mathbf{x}^*$
11: **end while**
12: **if** $i < N$ **then**
13: $i \leftarrow i + 1$
14: **else**
15: $i \leftarrow 1$
16: **end if**
17: **end while**
18: **return** $\mathbf{x}_{k-1}$

necessarily nullify the error, it will make it converge to a bounded value after K iterations. The same reasoning can be applied to the other substates (and suberrors). Interestingly, due to *Hypothesis 2*, $\left\|\underline{\mathbf{x}}^i_k - \underline{\mathbf{x}}^i_{k-1}\right\| = 0$, therefore the order of execution of the actions is irrelevant, since the new actions do not change the state already regulated by the former ones.

In practice though, *Hypothesis 2* might have to be relaxed to admit side effects on the other substates. In this case, the actions may have to be sequenced in an appropriate order to fasten (or at least attempt to guarantee) convergence of the global task error (2) to zero. Also, since an undesired change in complementary substates, whilst bounded by $\underline{\epsilon}^i$, is inevitable, it can be necessary to iterate the sequence of actions $\mathbf{u}^i$ (restart from $\mathbf{u}^1$ after $\mathbf{u}^N$: lines 14 to 16).

At each iteration of **Algorithm 1**, the action $\mathbf{u}^i_k$ to be applied is obtained by solving optimization (4a) on line 6. This is a difficult problem, since we have no analytic form of $f\left(\mathbf{x}_{k-1}, \mathbf{u}^i_k\right)$, hence no analytic form of

$$\overline{\mathbf{e}}^i_k = \mathbf{P}_i f\left(\mathbf{x}_{k-1}, \mathbf{u}^i_k\right) - \overline{\mathbf{x}}^*. \tag{5}$$

In such cases, since gradient-based optimization is not applicable, we propose to acquire this knowledge by *looking at how humans perform the task*. Then, we will rely on machine learning to infer the mapping from error $\overline{\mathbf{e}}^i_k$ to the *best possible corresponding action* $\mathbf{u}^i_k$, and particularly to its parameters $\mathbf{p}$. More specifically, we learn directly the mapping g from $\left(\overline{\mathbf{x}}^i_{k-1}, \overline{\mathbf{x}}^{i*}\right)$ to $\mathbf{u}^i_k$. In these cases (*i.e.*, when an analytical expression of f is not available) we replace line 6 of **Algorithm 1** with the following:

$$\mathbf{u}^i_k \leftarrow g\left(\overline{\mathbf{x}}^i_{k-1}, \overline{\mathbf{x}}^{i*}\right). \tag{6}$$

In the following section, we clarify the definition of $\mathbf{u}^i$ and $\overline{\mathbf{x}}^i$, by referring to the case study of Fig. 2.

Fig. 2. Visual features that characterize the desired shape: external contours (red), surface texture (green) and internal incisions (blue).

C. Case study

The case study shown in Fig. 2 consists in molding kinetic sand from the initial state (left) to the desired one (right). To create this shape three action types are considered ($N = 3$):

- *push*: to modify the external contours of the sand pile,
- *tap*: to homogenise the texture of the surface and
- *incise*: to draw the *S* shape on the sand.

We divide the state $\mathbf{x}$ into the 3 substates, consequent to the 3 types of actions: *external contours*, *internal texture*, and *internal incisions* (respectively red, green and blue in Fig. 2). These are defined by, respectively: a series of contour points coordinates $(X_1, Y_1, \ldots, X_{n_1}, Y_{n_1})$, a set of pixel luminosity values $(I_1, \ldots, I_{n_2})$ and the parametric scalar values characterizing the incision geometry $(p_1, \ldots, p_{n_3})$. The actions $\mathbf{u}^1$, $\mathbf{u}^2$ and $\mathbf{u}^3$ represent respectively *pushing*, *tapping* and *incising*. Action formalization is omitted here, as it depends on the robot platform and tool (spatula, robotic hand, etc.). The state representation in this example is:

$$\mathbf{x} = \begin{bmatrix} X_1 \\ Y_1 \\ \vdots \\ X_{n_1} \\ Y_{n_1} \\ I_1 \\ \vdots \\ I_{n_2} \\ p_1 \\ \vdots \\ p_{n_3} \end{bmatrix} \quad \mathbf{P}_1 = \begin{bmatrix} \mathbf{I_{2n_1}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad \mathbf{P}_2 = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I_{n_2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad \mathbf{P}_3 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I_{n_3}} \end{bmatrix}. \tag{7}$$

IV. USER STUDY AND DATASET ACQUISITION

To provide a robot with the knowledge needed for plastic manipulation, we ran a pilot user study, with 9 healthy volunteers (age range: 20-40; 6 male, 3 female). Each participant was asked to form a shape with the kinetic sand in a sandbox, while being recorded with a fixed RGB-D camera (Intel RealSense, resolution 640×480). The camera was pointing at the sandbox from above, with optical axis perpendicular to it, as shown in Fig. 3.

After being given time to familiarise and train manipulating the sand, each participant was requested to produce a shape of their choice three times: a) using both

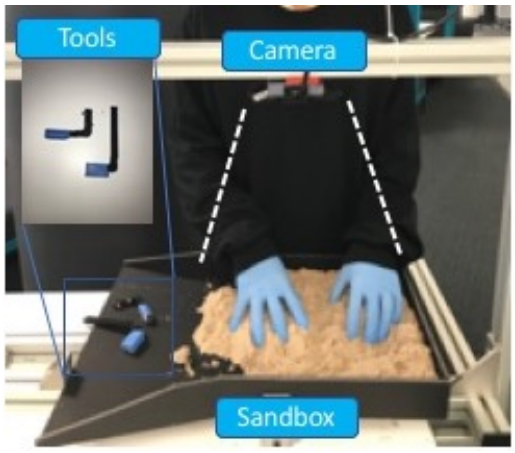

Fig. 3. Experimental setup: a sandbox is placed under an RGB-D camera (RealSense) while participants are asked to mold the sand into a shape. First with one, then with two hands and third, using one of two provided tools.

hands, b) using only one hand and finally c) using one of two provided tools (Fig. 3). The shape contours could touch the sandbox but this was not solicited. Four examples of shapes formed by the participants are shown in Fig. 1. Less than 30 minutes were necessary for each participant to complete the full trial. We report that all participants have given their consent to be recorded and the dataset is publicly available: https://cloudstor.aarnet.edu.au/plus/s/Vii90T72WFM8Qwp (password: sandman). Afterwards a questionnaire was filled in by the partakers, to help inferring the adopted strategy while manipulating the sand. In particular, we were interested in verifying our hypothesis that people use only a limited set of action types. None of the participants had previous direct experience with sculpting, although 5 declared to have had experience in manipulating deformable objects such as bread, dough, and air drying clay. Only one person perceived fatigue while performing the task with one hand, whilst 4 participants perceived some fatigue when using the tool, albeit minimally. No such issues were reported when using both hands.

The information extracted from the questionnaires can be summarised as follows: 66.7% of the participants stated that they had realised a clear sequence of different actions while using their hands, while the percentage raised to 88.9% when using the tool. This information leads us to believe that our hypothesis 1, the use of a set of action types, holds. In particular, the participants identified *pushing* as an action type. In our opinion, this is due to the fact that pushing has a very clear outcome (the sand moving and consequently the change of the contours of the sand shape). The other two action types we introduced in Sect. III-C, *tapping* and *incising*, were also identified. An analysis of the recordings leads us to introduce also mixed action types, *i.e.*, very often an action was performed which was a combination of pushing and tapping simultaneously (especially when the pushing action was performed on the sand and not at the contour, having the effect of also smoothing the surface of the sand).

All of the participants, whether using the hands or using the tool, reported either having relied on vision "very much" or "much". Haptic feedback was only partially important during the trials (77.7% and 66.6% of the participants reported either having relied on haptic feedback "little" or "neutral" respectively when using their hands and the tool). We believe that although the kinetic sand offers some force resistance while sculpting, this is not as valuable a feedback as vision. In our opinion, it is more natural to rely on visual feedback as a measure of how distant the current shape is from the imagined/desired shape. Haptic feedback is more valuable in understanding the force required to overcome the sand resistance, but offers little information on which action to perform next to get to the desired shape. The importance of visual feedback endorses our design choice of using cameras.

We manually labeled the images from the recorded videos (available in the dataset). Only the images where the three action types (pushing, tapping and incising) were clearly identifiable were labeled. Specifically, 13.34% of the images were labeled as "pushing", 9.31% as "tapping" and 8.68% as "incising". Analysing the videos, we noticed that the participants often performed mixed actions, as is clearly shown in 27.12% of the images, where the participants performed a pushing-tapping action (described earlier). The rest of the images were left unlabeled (41.55%). Unused images contain: unclear actions, *i.e.*, actions that it was not possible to categorize into any of the three labels; retreating actions and transitions between actions; and occlusions, *i.e.*, the tool was not visible, or the effects of the current action were not visible or were only visible after a few images. The analysis of the videos leads us to infer that a bigger set of action types should be taken into consideration and that *Hypothesis 2* should be relaxed: *i.e.*, $\underline{\epsilon}^i > 0$ in (4b).

V. TRANSFERRING PLASTIC MANIPULATION CAPABILITIES FROM HUMANS TO ROBOTS

In Sect. III, we explained that when the state dynamics f are unknown, standard optimization is hardly applicable. However, machine learning can provide an alternative solution, mapping the current error (in the perceived features) to the *best possible* action to reduce it. We train a neural network to learn this mapping using the data acquired during the user study presented above. This enables us to verify whether the dataset can be effectively exploited to transfer plastic manipulation capabilities from humans to robots.

Instead of deploying the complete *Robotic plastic material molding* algorithm presented in Sect. III-B, we validate our approach on a single pair of action type and corresponding subtask ($N = 1$): $\mathbf{u}$, $\overline{\mathbf{x}}$. Specifically, we focus only on *pushing* with a *tool*. In future work, we plan to exploit the whole dataset to learn multiple actions, also with other means (*e.g., one* or *two hands*) and extract features through convolutional layers.

The pushing task is characterised by current and desired

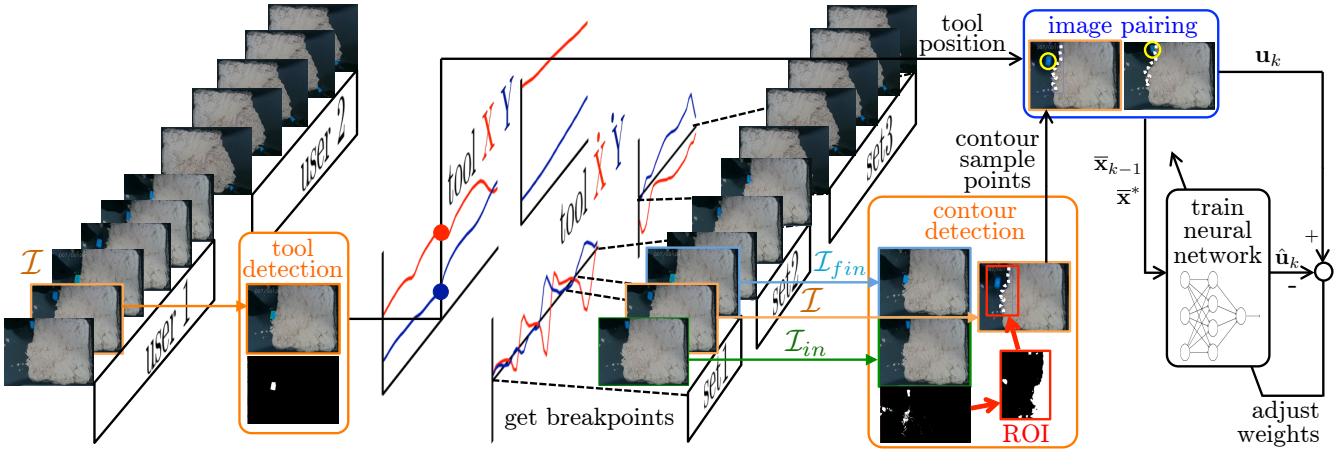

Fig. 4. The image processing pipeline. Left to right: sequence of valid "pushing" images, tool detection, tool position and velocity processing for reducing and breaking the sequence in smaller sets with similar velocity direction, contour detection, image pairing and using the data to train a neural network.

points along the external sand contour (m samples):

$$\begin{aligned} \overline{\mathbf{x}}_{k-1} &= \left[X^1_{k-1}\ Y^1_{k-1} \dots X^m_{k-1}\ Y^m_{k-1}\right]^\top \\ \overline{\mathbf{x}}^* &= \left[X^{1*}\ Y^{1*} \dots X^{m*}\ Y^{m*}\right]^\top. \end{aligned} \tag{8}$$

Since the pushing action $\mathbf{u}$ can be seen as a translation of the tool in the sandbox plane – parallel to the image plane – we parameterise it using the current and desired tool image coordinates:

$$\mathbf{u}_k\left(\mathbf{p}\right) = \left[X^t_{k-1}\ Y^t_{k-1}\ X^{t*}\ Y^{t*}\right]^\top. \tag{9}$$

Here, X^{t*} and Y^{t*} are the tool coordinates when the user has obtained the desired contour $\overline{\mathbf{x}}^*$.

With this representation, a mapping g (in (6)) can be learned from a large set of observed triplets $\{\mathbf{u}_k, \overline{\mathbf{x}}_{k-1}, \overline{\mathbf{x}}^*\}$. Obtaining many values of $\mathbf{u}_k$ and $\overline{\mathbf{x}}_{k-1}$ is straightforward. (6753 annotated *pushing* images have been annotated within our dataset) Defining $\overline{\mathbf{x}}^*$ as the final shape molded by each user would yield an insufficient training dataset (only 9 samples, one per user). Furthermore, the state error would be very large, potentially jeopardising convergence. To address this, we use all the sand contours in the dataset that have been obtained after a "sufficiently large" change in both contour and tool positions, as $\overline{\mathbf{x}}^*$. More formally, we take all $\overline{\mathbf{x}}^* = \overline{\mathbf{x}}_j$ such that:

$$\begin{cases} j > k-1 \\ \sqrt{\left(X^t_j - X^t_{k-1}\right)^2 + \left(Y^t_j - Y^t_{k-1}\right)^2} > \tau_u \\ \|\overline{\mathbf{x}}_j - \overline{\mathbf{x}}_{k-1}\| > \tau_{\overline{x}}, \end{cases} \tag{10}$$

with τ_u and $\tau_{\overline{x}}$ constant, hand-tuned thresholds. This representation differs slightly from the one presented in the case study of Sect. III-C and Fig. 2, it is though still covered in the general formalism of Sect. III-B. Indeed, it will be sufficient to label each of these "local" contours on the "global" shape as a different substate i, to be regulated sequentially.

A. Extraction of visual features

Although the RealSense camera provides also a depth image, we decided to use only the RGB images. We apply HSV space segmentation, since the tool tip is colored in blue, and the users wear black gloves and black long sleeved shirts, for the sand (light brown) and sandbox (black). However, when the user pushes the sand and partially occludes it with their hand, our algorithm cannot distinguish contours between sand and sandbox from contours between sand and gloves. This reduces the usable images from the original 6917 (from 8 users) annotated in the database as "pushing" to 4248 images of 3 users in which no sand occlusions occurred.

The pipeline, shown in Fig. 4, is as follows (left to right):

1) All 4248 valid "pushing" images are loaded sequentially and split into sets for each of the 3 users.
2) On each image $\mathcal{I}$, we perform a *tool detection*, yielding a tool position as the centroid of a blue blob, segmented in the HSV space. We discard images where the tool is not detected, and output tool *coordinates* X^t, Y^t for all other images.
3) After processing all images, the tool positions are low-pass filtered and derived to obtain the *tool velocities* per image $\mathcal{I}$: $\dot{X}^t$, $\dot{Y}^t$.
4) The original image sequence is reduced and broken into smaller sequences. This is done by detecting images where the tool has either stopped (null velocities) or changed direction (negative scalar product between consecutive velocities). We remove images where the tool is not detected, and split the sequence into smaller sets, with *breakpoints* corresponding to velocity changes. Within each of these new sets, the tool velocity does not change direction.
5) Each image set is processed by a *contour detection* algorithm. Subtracting final $\mathcal{I}_{fin}$ from initial image $\mathcal{I}_{in}$ in each set, a ROI (Region of Interest) is found wherein the sand configuration has changed the most. Within this ROI, the sand contour is detected on each image of the set, using the OpenCV `findContours` function. The sand contour is sampled with constant m (here, 10), to obtain $\overline{\mathbf{x}}$. We discard images where the contour is not detected.

TABLE I

TRAINED NEURAL NETWORK ERROR FOR TOOL POSITIONS (IN PIXELS).

component	mean error	σ of the error
X^t_{k-1}	3.1	4.0
Y^t_{k-1}	3.6	7.8
X^{t*}	2.3	3.1
Y^{t*}	2.8	5.0

6) At this stage, we have obtained 1858 samples each containing, for a given image $\mathcal{I}$: tool position $(X^t\ Y^t)$ (step 2) and contour samples $\overline{\mathbf{x}}$ (step 5). For our supervised learning approach, we require triplets of the form $\{\mathbf{u}_k, \overline{\mathbf{x}}_{k-1}, \overline{\mathbf{x}}^*\}$, with $\overline{\mathbf{x}}^* = \overline{\mathbf{x}}_j$ designed according to (10). Each set is now explored to *find pairs of images* $\mathcal{I}_{k-1}$, $\mathcal{I}_j$ with sufficient contour and tool change, *i.e*, pairs that comply with Eq. (10) (we use $\tau_u = 5$ and $\tau_{\overline{x}} = 3$ pixels). This operation augments the data to 7565 sample triplets employed to train a neural network to approximate g in Eq. (6).

B. Machine learning

Learning the mapping function (6) is a *fitting problem*, for which we decided to us a neural network to approximate the mapping function. The network consists of 40 input neurons (sum of the sizes of $\overline{\mathbf{x}}_{k-1}$ and $\overline{\mathbf{x}}^*$ with $m = 10$ contour samples) and 4 outputs (size of $\mathbf{u}_k$).

We design a simple multi-layer perceptron with one hidden layer, containing 100 neurons with sigmoidal activation function. As we have a supervised learning setup, backpropagation of errors is used to train the network (specifically the Levenberg-Marquardt Matlab implementation). We split the dataset of 7565 samples into training (70%), test (15%) and validation (15%) sets.

The network converges to a mean error of less than 4 pixels (mean and standard deviation σ of the error – both in pixels – are shown in Table I) after approximately 30 iterations, requiring slightly less than an hour on a Mac Book Pro 2013,

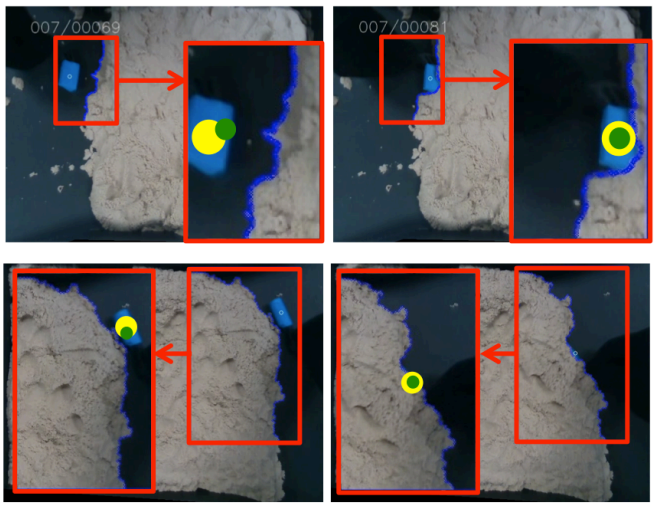

Fig. 5. Estimated (by the neural network) position of the tool in two pairs of images $\mathcal{I}_{k-1}$ (left), $\mathcal{I}_j$ (right). On each image, we show: the ROI (red), sand contour (dark blue), ground truth (yellow) and estimated (green) tool positions.

This is very promising, particularly when considering the tool size in the image (approximately 20×40 pixels) and the size of the area covered by a pixel in our setup (approximately 0.65×0.65 mm). To further illustrate these results, we compare the image processing (yellow circle) and network output (green circle) tool positions on two image pairs (Fig. 5). More and more varied data will yield better estimates. By releasing our user study images, we hope to encourage others to provide similar datasets, that will enable learning of better, more robust models.

VI. CONCLUSIONS AND PERSPECTIVES

This paper presents a step towards robotic manipulation of plastic material, herein molding kinetic sand. Our framework relies on the hypotheses that 1) the task is achievable using a finite set of action types and 2) those action types can regulate "mainly" a part of the task, without excessively disturbing the rest of it. We conduct a user study to verify the two hypotheses. The study shows that the participants use a finite set of action types, confirming our first hypothesis. The second hypothesis is only partially verified, as some undesired side effects of the actions are present.

The user study generated a dataset of images, which we publish alongside this paper. A pipeline consisting of image processing and machine learning is presented, which can provide the tool positions required to push the sand from a current to a desired configuration.

The accuracy obtained by our pipeline encourages us to pursue this line of research, and to deploy these results on an autonomous robot molding sand using visual feedback. Herein we proposed a formulation of this as a visual servoing task (Eq. 9). In the future, we will collect richer datasets and learn other action types such as tapping and incising.

ACKNOWLEDGMENTS

This work has been partly funded by the PHC FASIC project FlexBot, and it has been conducted by the Australian Research Council Centre of Excellence for Robotic Vision (project number CE140100016). We would also like to thank Anders Eriksson and Stephane Caron for the useful discussions on constrained optimization.

REFERENCES

[1] Y. Zheng, R. Pei, and C. Chen, "Strategies for automatic assembly of deformable objects," in *IEEE Int. Conf. on Robotics and Automation, ICRA*, 1991.

[2] M. Moll and L. E. Kavraki, "Path planning for deformable linear objects," *IEEE Trans. on Robotics*, vol. 22, no. 4, pp. 625–636, 2006.

[3] I. G. Ramirez-Alpizar, K. Harada, and E. Yoshida, "Motion planning for dual-arm assembly of ring-shaped elastic objects," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2014.

[4] R. P. Joshi, N. Koganti, and T. Shibata, "Robotic cloth manipulation for clothing assistance task using dynamic movement primitives," in *Proc. of the Advances in Robotics*, vol. 14, pp. 1–6, 2017.

[5] M. Shibata and S. Hirai, "Soft object manipulation by simultaneous control of motion and deformation," in *IEEE Int. Conf. on Robotics and Automation, ICRA*, pp. 2460–2465, May 2006.

[6] T. Wada, S. Hirai, S. Kawamura, and N. Kamiji, "Robust manipulation of deformable objects by a simple pid feedback," in *IEEE Int. Conf. on Robotics and Automation, ICRA*, vol. 1, pp. 85–90 vol.1, 2001.

[7] D. McConachie, M. Ruan, and D. Berenson, "Interleaving planning and control for deformable object manipulation," in *International Symposium on Robotics Research (ISRR)*, 2017.
[8] D. Berenson, "Manipulation of deformable objects without modeling and simulating deformation," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4525–4532, Nov 2013.
[9] H. Nakagaki, K. Kitagaki, T. Ogasawara, and H. Tsukune, "Study of deformation and insertion tasks of a flexible wire," in *IEEE Int. Conf. on Robotics and Automation, ICRA*, 1997.
[10] J. Miura and K. Ikeuchi, "Task planning of assembly of flexible objects and vison-based verification," *Robotica*, vol. 16, no. 3, pp. 297–307, 1998.
[11] D. Navarro-Alarcon, Y.-H. Liu, J. G. Romero, and P. Li, "Model-free visually servoed deformation control of elastic objects by robot manipulators.," *IEEE Trans. on Robotics*, vol. 29, no. 6, pp. 1457–1468, 2013.
[12] D. Navarro-Alarcon, Y. Liu, J. G. Romero, and P. Li, "On the visual deformation servoing of compliant objects: Uncalibrated control methods and experiments," *Int. Journal of Robotics Research*, 2014.
[13] K. Kosuge, H. Yoshida, T. Fukuda, M. Sakai, and K. Kanitani, "Manipulation of a flexible object by dual manipulators," in *IEEE Int. Conf. on Robotics and Automation, ICRA*, 1995.
[14] P. Long, W. Khalil, and P. Martinet, "Dynamic modeling of cooperative robots holding flexible objects," in *Int. Conf. on Advanced Robotics, ICAR*, 2015.
[15] A. Delgado, C. Jara, and F. Torres, "In-hand recognition and manipulation of elastic objects using a servo-tactile control strategy," *Robotics and Computer-Integrated Manufacturing*, vol. 48, pp. 102 – 112, 2017.
[16] A. Delgado, C. A. Jara, and F. Torres, "Adaptive tactile control for in-hand manipulation tasks of deformable objects," *Int. Journal of Advanced Manufacturing Technology*, vol. 91, pp. 4127–4140, Aug 2017.