



**HAL**  
open science

# Active Deformation through Visual Servoing of Soft Objects

Romain Lagneau, Alexandre Krupa, Maud Marchal

► **To cite this version:**

Romain Lagneau, Alexandre Krupa, Maud Marchal. Active Deformation through Visual Servoing of Soft Objects. ICRA 2020 - IEEE International Conference on Robotics and Automation, May 2020, Paris, France. pp.1-7. hal-02499488

**HAL Id: hal-02499488**

**<https://inria.hal.science/hal-02499488v1>**

Submitted on 5 Mar 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Active Deformation through Visual Servoing of Soft Objects

Romain Lagneau<sup>1</sup>, Alexandre Krupa<sup>2</sup> and Maud Marchal<sup>3</sup>

**Abstract**—In this paper, we propose the ADVISED (Active Deformation through Visual Servoing) method, a novel model-free deformation servoing method able to deform a soft object towards a desired shape. ADVISED relies on an online estimation of the deformation Jacobian that relates the motion of the robot end-effector to the deformation behavior of the object. The estimation is based on a weighted least-squares minimization with a sliding window. The robustness of the method to observation noise is ensured using an eigenvalue-based confidence criterion. The ADVISED method is validated through comparisons with a model-based and a model-free state-of-the-art methods. Two experimental setups are proposed to compare the methods, one to perform a marker-based active shaping task and one to perform several marker-less active shaping and shape preservation tasks. Experiments showed that our approach can interactively control the deformations of an object in different tasks while ensuring better robustness to external perturbations than the state-of-the-art methods.

## I. INTRODUCTION

Controlling the deformation of an object can be of importance, for instance in surgery [1], car manufacturing [2], or food manipulation [3]. The use of robots might bring repeatability, safety and high throughput in the manipulation of deformable objects. However, most of the current industrial robots are not able to control the deformation of an object because it requires both knowledge of the object physical properties and real-time tracking of the deformations. Several studies have focused on real-time deformation tracking [4]–[7] but not simultaneously with deformation control. A few methods have been proposed to find a grasp configuration on a deformable object [8] but without interactive control of the deformation. In this paper, we focus on methods that can perform interactive deformation servoing tasks, i.e. the manipulation of the shape of a deformable object at an interactive framerate.

Our contributions are as follows:

- the ADVISED method, a novel deformation servoing method that can work without knowledge about the physical properties of the object;
- two experimental setups to compare the ADVISED method with state-of-the-art methods.

The remainder of this paper is organized as follows: Sect. II presents the related work on deformation servoing, Sect. III describes the ADVISED method while Sect. IV and Sect. V detail the experimental setups used for the comparison with state-of-the-art methods. Finally Sect. VI ends the paper with a discussion on the approach.

## II. RELATED WORK

A possible categorization of deformation servoing methods is to distinguish those that depend on a prior model of the object (model-based methods) and those that do not rely on any characterization of the object (model-free methods). Model-based methods require prior knowledge about the object that is deformed, such as the geometry and/or the physical properties of the soft object. These methods are mostly combined with physically-based simulations to predict the deformation of the object and compare it with the real observation. For instance, Smolen and Patriciu proposed a physics-inspired model based on particles which does not require a mesh of the object [9]. Another possibility is to model the contact between the object and the end-effector instead of modeling the whole object. Navarro-Alarcon *et al.* proposed such a method where the contacts are modeled as springs [10]. It combines offline information, the system acquiring at different locations the effect of the end-effector motion on the object deformation, with online adaptation of variable parameters. The same authors extended their deformation servoing method in [11] by performing simultaneously positioning and shaping tasks. Other approaches use the mass-spring formulation, which exhibits good computation time performances. These mass-spring models were even adapted to model the plastic behavior of the object [12]. However, mass-spring models do not express properly actual physical properties of the object and therefore often lack accuracy. For more accurate simulation, the Finite Element Method (FEM) has been introduced to better estimate the deformation of an object. Thus, Zhang *et al.* used it for controlling soft robots [13]. Frank *et al.* presented a method for robot end-effector path planning while avoiding the deformation of a soft object [14]. The FEM physics simulation has also been used for automatic flexible needle insertion in deformable tissue using a predefined path, for instance in the work of Adagolodjo *et al.* [15]. Assuming that the geometry and the physical properties of the object are available, model-based methods are able to provide accurate representations of the deformation of an object. However, the object representation and mesh reconstruction as well as the mechanical properties measurement remain still very time-consuming as-of-today and most of the proposed approaches are not able to reach on a real system the same accuracy that it could be expected from simulations.

Some deformation servoing methods that do not require any prior knowledge of the object have been proposed, such as the one of Berenson able to locally deform 1D and 2D soft objects [16]. Alonso-Mora *et al.* proposed a

<sup>1</sup>Univ. Rennes, INSA, IRISA, Inria, CNRS

<sup>2</sup>Univ. Rennes, Inria, IRISA

<sup>3</sup>Univ. Rennes, INSA, IRISA, Inria, CNRS, IUF

e-mail: {romain.lagneau, alexandre.krupa, maud.marchal}@irisa.fr.

method to perform simultaneously the positioning and the shape preservation of 2D deformable objects [17]. However, shape preservation is only a secondary task neglected when its achievement induces a collision. Navarro-Alarcon *et al.* proposed a method to deform objects based on an online Jacobian estimation [18]. The online estimation is based on the Broyden algorithm that infers the variation of the Jacobian from the current and previous observations [19]. However, Broyden algorithm is very sensitive to noise: when the system is converging towards the desired configuration, some directions of the Jacobian are not stimulated and their update is performed using only measurement noise. Navarro-Alarcon *et al.* proposed another method that considers the contour of the object to perform shape deformation [20]. This method was based on the expression of the 2D shape of the object with a compact vector of Fourier coefficients. However, the system falls in local minima if the desired contour is not reachable or if the number of Fourier coefficients is not enough to well describe the shape of the object.

#### Positioning of our work

Our method is inspired by the Uncalibrated Image-Based Visual Servoing (IBVS) methods and belongs to the online model-free method category. We aim at controlling the deformation of an object without requiring any prior model or physical characteristics of the targeted object. Our approach is based on an online estimation of a deformation Jacobian that links the variation of the object deformation to the velocity screw vector of the robot end-effector that is in contact with the deformable object. This deformation Jacobian is used in the control law to deform the object.

There exist several ways for estimating online the Jacobian of a function. Hosoda and Asada proposed a method using the Lyapunov theory to tackle the noise sensitivity issue of the Broyden algorithm [21]. Their method was extended by Navarro-Alarcon *et al.* for deformation servoing [18]. This later method needs an accurate tuning of several parameters to ensure robustness. Moreover, it relies on 2D information to deform the object, which makes it difficult to control the whole shape of the object. In our approach, we propose to use only a small set of parameters to tune for estimating online the deformation Jacobian. We use a RGB-D sensor to quickly get 3D deformation observations allowing to control the global object shape. We combine these real-time observations with a weighted least-squares minimization with a sliding window [22]. Compared to the method proposed in [19], this method is more robust to noise since each row of the deformation Jacobian matrix is updated using a confidence criterion computed from the deformation observations. To the best of our knowledge, this method has been used only for standard IBVS by Krupa [23]. His objective was to control the orientation of a 2-Degree-Of-Freedom (2DOF) surgical instrument in the context of laparoscopic surgery using the image 2D coordinates of a laser spot projected from the instrument on the scene. Our approach aims at controlling the 3D shape of a deformable object by automatically applying a motion to the end-effector that performs the shaping task.

### III. DESCRIPTION OF OUR METHOD

This paper aims at proposing a novel adaptive model-free deformation servoing method to control the 3D shape of an object. The formulation that will be used in this article is the following:

- $k$  denote the current discrete time. It is precised only when confusion can arise in a formula,
- $\mathbf{v}^{\text{ctrl}} \in \mathbb{R}^m$  is the computed velocity control law vector expressed in the robot end-effector frame, with  $m$  the number of DOFs of the end-effector,
- $\mathbf{e} = \mathbf{s} - \mathbf{s}^* \in \mathbb{R}^n$  is the deformation error vector,  $\mathbf{s}$  and  $\mathbf{s}^*$  being respectively the current and desired deformation feature vectors. The size of the deformation and deformation error vectors is denoted  $n$ ,
- $\hat{\mathbf{J}}^+$  is the Moore-Penrose pseudo-inverse of an approximation of the deformation Jacobian  $\mathbf{J}$ , which links the evolution of the deformation with the end-effector actual velocity  $\mathbf{v}^e$  i.e.  $\dot{\mathbf{s}} = \mathbf{J}\mathbf{v}^e$ .

Our novel model-free deformation servoing method estimates the deformation Jacobian using a weighted least-squares minimization method with a sliding window. The Jacobian is updated only w.r.t. the DOFs for which a confidence criterion has been met. This confidence criterion permits to filter noise measurement when the system converges towards the desired deformation. Assuming that the variation of the deformation features  $\dot{\mathbf{s}}$  is linear with regards to small motions of the end-effector, their relation can be written as:

$$\dot{\mathbf{s}}_i = \mathbf{v}^{eT} \hat{\mathbf{J}}_{i,:}^T + \mathbf{v}_i \in \mathbb{R} \text{ for } i \in \{0, \dots, n-1\} \quad (1)$$

where the vector  $\mathbf{v}^e$  designates the velocity of the end-effector, the vector  $\hat{\mathbf{J}}_{i,:}$  designates the  $i^{\text{th}}$  row of the deformation Jacobian matrix that we want to estimate and  $\mathbf{v}_i$  is the  $i^{\text{th}}$  element of the noise measurement vector.

To minimize the predictive error  $\dot{\mathbf{s}}_i(k+1) - \mathbf{v}^{eT}(k)\hat{\mathbf{J}}_{i,:}^T(k)$ , we propose the following quadratic cost function  $C(\hat{\mathbf{J}}_{i,:}^T) \in \mathbb{R}$ :

$$C(\hat{\mathbf{J}}_{i,:}^T) = \sum_{j=k-N}^k \left( \frac{\lambda^{k-j}}{m^2(j-1)} (\dot{\mathbf{s}}_i(j) - \mathbf{v}^{eT}(j-1)\hat{\mathbf{J}}_{i,:}^T(j))^2 \right) \quad (2)$$

where  $N$  is the size of the sliding window, the index  $j$  indicates that the sum is performed over previous discrete times,  $0 < \lambda \leq 1$  is a constant forgetting factor giving less importance to the oldest measures and  $m(k) = \sqrt{1 + \mathbf{v}^{eT}(k)\mathbf{v}^e(k)}$  is a normalization signal. Let's define:

$$\mathbf{R} = \sum_{j=k-N}^k \left( \frac{\lambda^{k-j}}{m^2(j-1)} \mathbf{v}^e(j-1)\mathbf{v}^{eT}(j-1) \right) \in \mathbb{R}^{m,m} \quad (3)$$

$$\mathbf{Q} = \sum_{j=k-N}^k \left( \frac{\lambda^{k-j}}{m^2(j-1)} \mathbf{v}^e(j-1)\dot{\mathbf{s}}_i(j) \right) \in \mathbb{R}^{m,1} \quad (4)$$

The cost function can be rewritten as follows:

$$C = \sum_{j=k-N}^k \left( \frac{\lambda^{k-j}}{m^2(j-1)} \dot{\mathbf{s}}_i^2(j) \right) + \hat{\mathbf{J}}_{i,:} (\mathbf{R}\hat{\mathbf{J}}_{i,:}^T - 2\mathbf{Q}) \quad (5)$$

The cost function is convex with regards to  $\hat{\mathbf{J}}_{i,:}^T$ , thus its minimum is obtained for the value of  $\hat{\mathbf{J}}_{i,:}^T$  that nullifies its gradient  $\nabla C$  with regards to  $\hat{\mathbf{J}}_{i,:}^T$ , where  $\nabla C = -2\mathbf{Q} + 2\mathbf{R}\hat{\mathbf{J}}_{i,:}^T$ .

The estimated  $\hat{\mathbf{J}}_{i,:}^T$  that nullifies the gradient can be written:

$$\hat{\mathbf{J}}_{i,:}^T = \mathbf{R}^{-1}\mathbf{Q} \in \mathbb{R}^{m,1} \quad (6)$$

In practice, it is not possible to implement directly (6). Indeed, if the system is not well stimulated, which is notably the case when the error tends towards 0, the eigenvalues of the matrix  $\mathbf{R}$  tend towards 0. Consequently, it is not possible to invert the matrix. To avoid this problem, a confidence criterion based on eigenvalue decomposition is used to determine how the Jacobian will be updated. The matrix  $\mathbf{R}$  being a positive-definite symmetric  $m \times m$  matrix, it can be decomposed such as:

$$\begin{aligned} \mathbf{R} &= \mathbf{\Phi}\mathbf{\Gamma}\mathbf{\Phi} & (7) \\ \text{with } \mathbf{\Phi} &= [\boldsymbol{\phi}_0 \quad \dots \quad \boldsymbol{\phi}_{m-1}] \in \mathbb{R}^{m,m} \\ \text{and } \mathbf{\Gamma} &= \begin{bmatrix} \gamma_0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \gamma_{m-1} \end{bmatrix} \in \mathbb{R}^{m,m} \end{aligned}$$

where  $\gamma_0 \geq \dots \geq \gamma_{m-1} \geq 0$  are the eigenvalues of  $\mathbf{R}$  and  $\boldsymbol{\phi}_0, \dots, \boldsymbol{\phi}_{m-1}$  are its eigenvectors. Finally, the update of the estimated deformation Jacobian is performed depending on a user-defined confidence threshold  $\epsilon$ :

$$\begin{aligned} \hat{\mathbf{J}}_{i,:}^T(k) &= \begin{cases} \mathbf{R}^{-1}\mathbf{Q} & \text{if } \gamma_0 > \epsilon, \dots, \gamma_{m-1} > \epsilon \\ \hat{\mathbf{J}}_{i,:}^T(k-1) & \text{if } \gamma_0 \leq \epsilon, \dots, \gamma_{m-1} \leq \epsilon \\ \mathbf{V}_1\mathbf{Q} + \mathbf{V}_2\hat{\mathbf{J}}_{i,:}^T(k-1) & \text{otherwise} \end{cases} & (8) \\ \text{with } \mathbf{V}_1 &= \mathbf{\Gamma}_{:,0:j}\mathbf{\Phi}_{0:j,0:j}^{-1}\mathbf{\Gamma}_{:,0:j}^T \\ \text{and } \mathbf{V}_2 &= \mathbf{\Gamma}_{:,j+1:m-1}\mathbf{\Gamma}_{:,j+1:m-1}^T \\ & \text{with } \phi \text{ sorted such as } \phi_j > \epsilon \text{ and } \phi_{j+1} \leq \epsilon \end{aligned}$$

The subscript “:” indicates that all the rows (respectively columns) are selected when it appears before (respectively after) the comma, while the subscript “0:j” indicates that all the rows (or columns) from index 0 to index j are selected. The  $\mathbf{V}_1$  term in (8) allows to update the Jacobian in the directions where the eigen-value-based confidence criterion is met, while the term  $\mathbf{V}_2$  allows to ignore the information in the noisy directions.

Once the Jacobian has been updated, the velocity control law is computed such as:

$$\mathbf{v}^{\text{ctrl}} = -\alpha\hat{\mathbf{J}}^+ \mathbf{e} \in \mathbb{R}^m \quad (9)$$

where  $\alpha$  is the control law gain.

#### IV. COMPARISON WITH STATE-OF-THE-ART METHODS ON A MARKER-BASED ACTIVE SHAPING TASK

To evaluate our approach, we compared it with two state-of-the-art methods: a model-free deformation method proposed by Navarro-Alarcon et al. [18] and a model-based deformation method similar to the one proposed by Adagoldjo et al. [15]. First, we recall the main principles of the two state-of-the-art approaches. Then, we present a performance

evaluation of the three methods when performing a marker-based active shaping task.

##### A. Description of the state-of-the-art methods

1) *Model-based deformation method*: We implemented a model-based deformation servoing method similar to the one proposed in [15]. We used a physically-based corotational FEM simulation using the SOFA framework [24] to estimate the deformation Jacobian. The velocity control law is given by  $\mathbf{v}^{\text{ctrl}} = -\alpha\hat{\mathbf{J}}^+ \mathbf{e} \in \mathbb{R}^m$  and  $\alpha$  is the control law gain. Several physically-based simulations are run to estimate the deformation Jacobian. Each simulation performs a stimulation of the end-effector on the object by applying a translation or rotation  $dir_j$  along the axis of the end-effector frame and observe the resulting simulated object deformation. The partial Jacobians  $\hat{\mathbf{J}}^{dir_j} \in \mathbb{R}^{n,1}$  are estimated such as:

$$\hat{\mathbf{J}}_i^{dir_j} = \frac{\Delta \mathbf{s}_i}{\mathbf{v}_j \Delta t}, \quad i \in \{0, \dots, n-1\}, \quad j \in \{0, \dots, m-1\}$$

where  $\mathbf{v}_j$  is the constant input velocity of the simulated end-effector in the  $j^{\text{th}}$  direction and the subscripts indicate the indices of an element in a vector or a matrix.

Then, all the individual deformation Jacobians are gathered in order to form the global deformation Jacobian matrix  $\hat{\mathbf{J}}$  such as  $\hat{\mathbf{J}} = [\hat{\mathbf{J}}^{dir_0} \quad \dots \quad \hat{\mathbf{J}}^{dir_{m-1}}] \in \mathbb{R}^{n,m}$ . This updated Jacobian will be used until the next update to control the deformation of the object.

2) *Model-free deformation servoing*: We also compared our approach with the model-free deformation servoing method proposed by Navarro-Alarc3n et al. in [18]. This method iteratively estimates the deformation Jacobian matrix  $\hat{\mathbf{J}}$  from the deformation feature vector and the robot velocity observation using the Broyden update rule [19]:

$$\hat{\mathbf{J}}(k) = \hat{\mathbf{J}}(k-1) + \varepsilon \frac{\Delta \mathbf{s}(k) - \hat{\mathbf{J}}(k-1)\Delta \mathbf{x}(k)}{\Delta \mathbf{x}^T(k)\Delta \mathbf{x}(k)} \Delta \mathbf{x}^T(k) \in \mathbb{R}^{n,m} \quad (10)$$

where  $0 < \varepsilon \leq 1 \in \mathbb{R}$  is a user-defined responsiveness gain and  $\mathbf{x} \in \mathbb{R}^3$  is the relative linear displacement of the end-effector position. Only the three linear velocities of the end-effector are controlled with this method, i.e.  $m = 3$ .

The estimated deformation Jacobian is thereafter used for the computation of the control law. The control law was developed according to the Lyapunov theory to ensure the robustness of the controller. The velocity control law is:

$$\begin{aligned} \mathbf{v}^{\text{ctrl}} &= \alpha\hat{\mathbf{J}}^+ (\mathbf{p} - c \text{sat}(K_s \mathbf{e})) \in \mathbb{R}^m & (11) \\ & \text{with } \mathbf{p} = -\text{sat}(K_s \mathbf{e}) - \mathbf{C}\mathbf{p} \in \mathbb{R}^n \end{aligned}$$

where  $\mathbf{p}$  is a numerical state vector,  $\mathbf{C} \in \mathbb{R}^{m,m}$  is a symmetric positive damping matrix,  $c$  is a positive scalar acting like a damping-like feedback gain,  $\text{sat}$  is a saturation function,  $K_s \in \mathbb{R}^+$  is a saturation gain and  $\alpha$  is the control law gain.

##### B. Initialization of the Jacobian

The deformation Jacobian of both model-free deformation methods is initialized with a coarse guess of its value. An offline open-loop deformation Jacobian estimation is done: the end-effector is moved in each direction successively to

simulate the deformable object while recording the observed deformation. Then, the Jacobian is initialized as follows:

$$\hat{\mathbf{J}}_{i,j} = \frac{\Delta \mathbf{s}_i}{in_j \Delta t} \text{ for } i \in \{0, \dots, n-1\}, j \in \{0, \dots, m-1\} \quad (12)$$

with  $\Delta \mathbf{s}_i = \mathbf{s}_i(k) - \mathbf{s}_i(k-1) \in \mathbb{R}$

where  $in_j$  represents a constant velocity input that is applied successively on each axis during a period  $\Delta t$ , and the subscript  $i$  denotes the  $i^{\text{th}}$  row of a vector or a matrix. This offline initialization takes a few seconds.

### C. Experimental setup for the marker-based active shaping task

The experimental setup is composed of a 6-DOF anthropomorphic robot arm (Viper 650 from ADEPT) where the end-effector distal tool corresponds to a 3D-printed stylus (see Fig.3 for the setup, the foam is represented in an inset). A sheet of foam is deformed in order to reach a desired curved shape. A remote Intel Realsense D435 RGB-D camera is used for tracking both the robot and the soft object. The depth information is accurate within  $\pm 0.5\text{mm}$ .

For the marker-based active shaping task, the deformation of the object is tracked using  $\eta$  passive visual markers attached to the object surface. These markers are tracked in the RGB image using ViSP blob tracking abilities [25]. The deformation vector groups the 3D position of each marker w.r.t. the camera that can directly be retrieved from the depth sensor data, i.e.  $\mathbf{s} = (x_0 \ y_0 \ z_0 \ \dots \ x_{\eta-1} \ y_{\eta-1} \ z_{\eta-1})^T \in \mathbb{R}^n$ .

To determine the constant transformation between the remote camera frame  $R_c$  and the robot base frame  $R_r$ , we used a computer vision approach to determine the homogeneous transformation between the robot end-effector frame  $R_e$  and the camera frame. We then chain this transformation to the homogeneous transformation between the end-effector frame  $R_e$  and robot base frames frame  $R_r$  provided by the robot forward kinematics. This determination is performed at the initialization of the experiment.

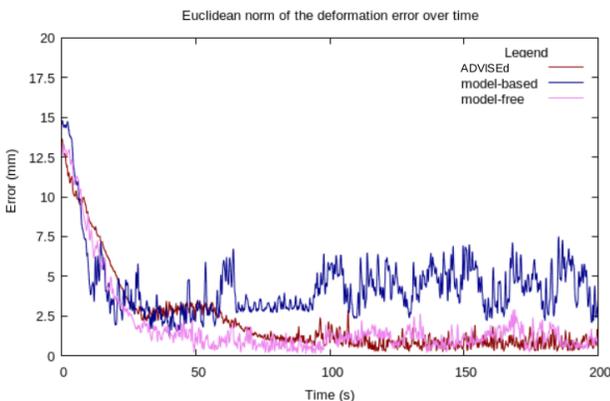


Fig. 1: Euclidean norm of the deformation error for the three methods during a marker-based active shaping task.

The comparison of performances between the different methods takes into account several criteria:

- $t_r$ : response time of the system, expressed in seconds, which corresponds to the time needed to reduce the error to 10% of its initial value  $e_0$ ,
- $t_s$ : settling time of the system, expressed in seconds, which corresponds to the time after which the system stays in the interval  $[e_\infty - 0.05e_0 ; e_\infty + 0.05e_0]$ ,
- $e_\infty$ : final error, expressed in millimeters, between the desired and observed deformation vectors when the system finally is in steady-state,
- $d_{eff}$ : total distance traveled by the end-effector during the manipulation, expressed in millimeters

For the marker-based active shaping task, the considered error function is the Euclidean norm of the deformation error vector i.e.  $e_0 = \|\mathbf{e}(0)\|$  and  $e_\infty = \|\mathbf{e}(k \rightarrow \infty)\|$ .

Fig.1 illustrates the evolution of the Euclidean norm of the deformation error vector obtained with the different methods when performing an active shaping task consisting in reaching a desired curved shape. Two passive markers were used during this experiment, so the size of the deformation and deformation error vectors was  $n = 6$ . Each algorithm was optimally tuned to achieve the best trade-off between accuracy, stability and responsiveness. For ADVISED, the parameters were:  $\lambda = 0.99$ ,  $N = 10$ ,  $\epsilon = 0.001$ ,  $\alpha = 0.05$ . For the model-based method, the parameters were: Young modulus = 13kPa, Poisson's ratio = 0.3, Rayleigh stiffness = 0.125, mass density =  $40\text{kg}\cdot\text{m}^{-3}$ . For the model-free method, the parameters were:  $\mathbf{C} = 0.5\mathbf{I}^{6 \times 6}$ ,  $c = 1$ ,  $K_s = 3$ ,  $\epsilon = 0.3$ ,  $\alpha = 0.0125$ , saturation threshold = 0.02.

The model-based method, whose results are depicted in blue in Fig.1, was less stable ( $d_{eff} = 63.5\text{mm}$  w.r.t. 16.3mm for both model-free methods) and less accurate ( $e_\infty = 3.5\text{mm}$ ) than the two model-free methods (0.7mm for ADVISED and 0.8mm for the other model-free method). This is due to the difficulty to accurately obtain the physical parameters that are required for modeling the geometry and physical properties of a deformable object as well as the surrounding scene. This enforced our idea of proposing a model-free method to avoid modeling constraints. The ADVISED method and model-free state-of-the-art method had similar results in terms of stability and accuracy. The model-free state-of-the-art method was more responsive than the ADVISED method ( $t_r = 35\text{s}$  compared to 70s) but needed more time to become stable than the ADVISED method ( $t_s = 175\text{s}$  compared to 125s). We thus decided to conduct additional experiments to compare these two methods.

## V. COMPARISON WITH A MODEL-FREE STATE-OF-THE-ART METHOD ON A MARKER-LESS ACTIVE SHAPING TASK

### A. Experimental Setup

We used the same hardware as presented in Sect. IV-C. Several soft objects were manipulated: a flat piece of foam, a soft ball and a soft toy. The experimental setup is shown in Fig. 3, the soft objects being presented as insets.

For this set of experiments, the deformation of the object is tracked using a marker-less method. A 3D point-cloud composed of  $\eta$  3D points located in a given

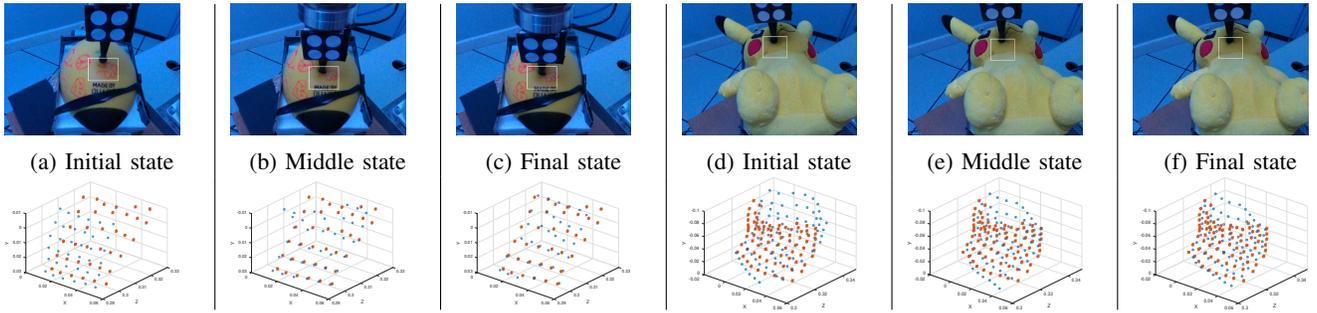


Fig. 2: Examples of active shaping (a to c) and shape preservation tasks(d to f). Blue dots and red stars represent the current and desired position of the sampled points of the SoI (white rectangles in the RGB pictures). Outliers have not been removed.

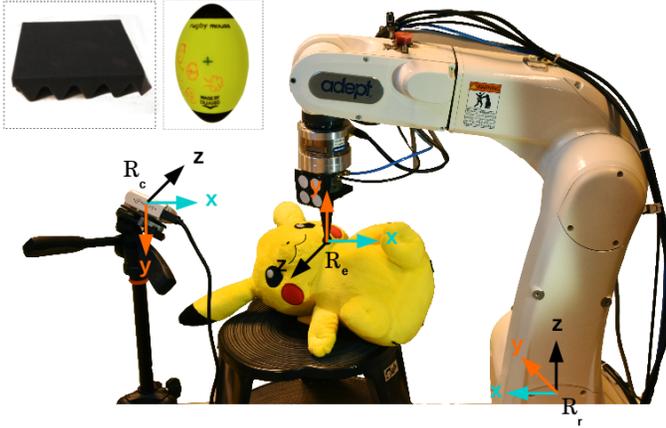


Fig. 3: Experimental setup showing the RGB-D camera and the robot performing a soft object shape preservation task.

Surface of Interest (SoI) and captured by the RGB-D camera is now selected as deformation feature vector  $\mathbf{s} = (x_0 \ y_0 \ z_0 \ \dots \ x_{\eta-1} \ y_{\eta-1} \ z_{\eta-1})^T$ . The projected area of the SoI on the surface of the object was set to  $3\text{cm} \times 3\text{cm}$ . Different downsamplings of the point-cloud are tested in order to determine the impact of the number of 3D points onto the convergence to a desired configuration  $\mathbf{s}^*$ . To reduce measurement noise, M-estimation is used in order to remove outliers from the deformation error vector, such as the lack of depth data at one point due to sensor failure or occlusion due to the end-effector presence [26]. We apply Iteratively Reweighted Least-Squares (IRLS) method in order to convert the M-estimation problem into an equivalent least-squares problem, as proposed in [27]. The error used in the computation of the control law becomes  $\mathbf{e}' = \mathbf{W}(\mathbf{s} - \mathbf{s}^*) \in \mathbb{R}^n$ , where  $\mathbf{W}$  is a diagonal weighting matrix containing the weights reflecting the confidence in the deformation features.  $\mathbf{W}$  is recomputed each time a new deformation vector is made available using Tukey M-estimator [28]. This new expression of the deformation error vector changes the control law (9) in:

$$\mathbf{v}^{\text{ctrl}} = -\alpha(\mathbf{W}\hat{\mathbf{J}})^+ \mathbf{e}' \in \mathbb{R}^m \quad (13)$$

To ensure a fair comparison between the ADVISED method and this method, the velocity control law shown in

TABLE I: Comparison of the algorithms results, times being in seconds,  $e_{H\infty}$  and  $d_{eff}$  in millimeters. "div." means that the system diverged, "N/A" stands for "Not Applicable".

Exp.	SoI	Method	$t_r$	$t_s$	$e_{H\infty}$	$d_{eff}$
a	$4 \times 4$	ADVISED	30	80	2.1	21.3
a	$4 \times 4$	model-free	20	50	2.1	26.2
a	$5 \times 5$	ADVISED	40	90	1.0	35.0
a	$5 \times 5$	model-free	5	90	2.1	21.1
b	$6 \times 6$	ADVISED	40	90	2.0	28.2
b	$6 \times 6$	model-free	20	90	2.0	23.4
b	$12 \times 6$	ADVISED	150	160	1.0	16.4
b	$12 \times 6$	model-free	5	50	1.0	23.5
c	$5 \times 5$	ADVISED	90	90	2.1	34.5
c	$5 \times 5$	model-free	N/A	N/A	div.	N/A
c	$6 \times 6$	ADVISED	200	210	1.0	48.0
c	$6 \times 6$	model-free	10	20	1.0	42.4
c	$9 \times 9$	ADVISED	180	220	2.1	131.1
c	$9 \times 9$	model-free	75	220	3.1	66.5

(11) has been adapted as follows:

$$\mathbf{v}^{\text{ctrl}} = \alpha(\mathbf{W}\hat{\mathbf{J}})^+ \mathbf{W}(\mathbf{p} - \text{csat}(K_s \mathbf{e})) \in \mathbb{R}^m \quad (14)$$

The Hausdorff distance is a metrics commonly used to compare two surfaces [29]. Let  $e_H(k)$  be the Hausdorff distance between the desired and observed surface at the discrete time  $k$ . For the different marker-less tasks, the final error that is considered is  $e_{H\infty} = e_H(k \rightarrow \infty)$ .

Several kind of experiments were conducted:

- a) folding a piece of foam to reach a desired shape,
- b) deforming the surface of a soft ball to reach a desired shape (Fig.2a to 2c),
- c) bringing back a deformed shape of the soft toy to a desired less deformed shape (Fig.2d to 2f),
- d) same as exp. a) with external disturbances (upwards and downwards 3-centimeter displacements).

Experiments a) and b) correspond to active shaping tasks of soft objects. Experiment c) is a shape preservation task. Experiment d) permits to evaluate the robustness of the method with regards to external disturbances. Before all of these experiments, the end-effector is moved in order to deform the object. The resulting deformed surface will be used as desired surface. The object and end-effector are thereafter put back in their resting state.

## B. Comparison of the results

During each of these experiments, the state-of-the-art method parameters used were as follows:  $\mathbf{C} = 0.5\mathbf{I}^{6 \times 6}$ ,  $c = 1$ , saturation threshold = 0.02,  $K_s = 3$ ,  $\varepsilon = 0.3$  and  $\alpha = 0.045$ .

For experiments a) b) and c), several downsamplings of the point-cloud located in the SoI were tested to determine the impact of the number of points  $\eta$  on task completion. During these experiments, the ADVISED method parameters were as follows:  $\alpha = 0.045$ ,  $\lambda = 0.99$ ,  $N = 10$  and  $\epsilon = 0.001$ . The results of these experiments are presented in Table I. One can notice that the ADVISED method was able to reach the same accuracy as the model-free state-of-the-art method. The number of considered points  $\eta$  in the SoI impacts the convergence speed of the methods because they are more constrained. The state-of-the-art method is more reactive than the ADVISED method in terms of response time. This is due to the forgetting window of the ADVISED method that tends to smooth the update of the deformation Jacobian.

TABLE II: Evaluation of the performances of the ADVISED method when subject to external disturbances.  $t_s$  denotes the duration between the last disturbance and the steady state.

$\lambda$	N	$\epsilon$	$t_r$ (s)	$t_s$ (s)	$e_\infty$ (mm)	$d_{eff}$ (mm)
0.75	10	0.001	80	5	2.1	61
0.99	10	0.001	80	75	2.1	25.2
0.99	5	0.001	125	100	3.1	106.2
0.99	20	0.001	175	10	3.1	47.2
0.99	10	0.01	50	10	2.1	34.9

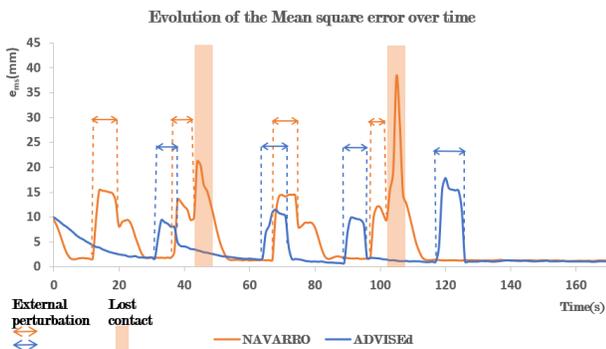


Fig. 4: Evolution of the mean square error function during an active shaping task when external disturbances occur.

Finally, experiment d) has been conducted to evaluate the robustness of the methods to external disturbances. The disturbances could either move the object upwards or downwards. Different sets of parameters have been tested for the ADVISED method. The results are summarized in Table II. Increasing the size of the weighting window and decreasing the value of the forgetting factor both increase the robustness of the method w.r.t. external perturbations by decreasing the impact of occasional outliers. Increasing the value of the confidence threshold increases the robustness of the method by filtering noise measurements.

The Hausdorff distance being sensitive to the camera sensor noise, we chose to represent instead the mean square

error  $e_{ms}(k) = \frac{\sqrt{\mathbf{e}^T(k) \cdot \mathbf{e}(k)}}{\eta}$  for display purpose only. Fig. 4 depicts the evolution of  $e_{ms}$  for each method when external disturbances were applied during an active shaping task. During this experiment, the ADVISED method parameters were as follows:  $\alpha = 0.045$ ,  $\lambda = 0.99$ ,  $N = 10$  and  $\epsilon = 0.001$ . The disturbances occurred when the system reached a steady-state and are depicted by arrows. The state-of-the-art method was more responsive than the ADVISED method. However, the high responsiveness of the state-of-the-art method made it lose contact with the object in several occasions, depicted by filled areas in Fig. 4. On the other hand, the ADVISED method had a smooth response to the external perturbations ensuring an exponential decrease of the error and a permanent contact with the object. This different behavior between the two methods make them suitable for different kind of applications: the ADVISED method is well-adapted for applications where smooth behavior is required whereas the state-of-the-art method is more suitable when high responsiveness is desired and overshoot is not critical.

## VI. CONCLUSION

In this paper we presented the ADVISED method, a model-free deformation servoing method based on a numerical estimation of the deformation Jacobian. Our method combines the advantages of requiring a small number of parameters while not relying on any prior knowledge on the manipulated soft object. Our method is based on a weighted least-mean squares minimization with a sliding window to online estimate the deformation Jacobian. This later is efficiently updated by determining the reliability of the observations using eigenvalue decomposition and a confidence threshold.

Several experiments were conducted to compare the ADVISED method to model-based and model-free state-of-the-art methods, and to evaluate its ability to control the deformation of a soft object. A marker-based active shaping task showed that the model-based method required accurate knowledge of the physical parameters of the object and its neighborhood to ensure correct results whereas the our method and the model-free method did not suffer from this issue.

Further experiments of both active shaping and surface preservation tasks have been conducted to compare the ADVISED method to the state-of-the-art model-free method. The accuracy of the ADVISED method is independent of the number of points located in the Surface of Interest as well as external disturbances. Experiments showed that the ADVISED method was more robust to external disturbances and as accurate as the state-of-the-art methods while relying on a smaller amount of parameters to tune. As such, the ADVISED method is a promising alternative for interactively controlling the deformations of soft objects.

## ACKNOWLEDGMENT

This research has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 731761 for project "Imagine".

## REFERENCES

- [1] B. Protyniak, J. Jorden, and R. Farmer, "Multi-quadrant robotic colorectal surgery: the da vinci xi vs si comparison," *Journal of robotic surgery*, vol. 12, no. 1, pp. 67–74, 2018.
- [2] M. Doroudian, M. Kolich, G. Almasi, M. Medoro, R. Dwarampudi, S. Salokhe *et al.*, "Virtual temperature controlled seat performance test," SAE Technical Paper, Tech. Rep., 2018.
- [3] Z. Wang and S. Hirai, "Modeling and estimation of rheological properties of food products for manufacturing simulations," *Journal of food engineering*, vol. 102, no. 2, pp. 136–144, 2011.
- [4] A. Petit, V. Lippiello, G. A. Fontanelli, and B. Sicialiano, "Tracking elastic deformable objects with an rgb-d sensor for a pizza chef robot," *Robotics and Autonomous Systems*, vol. 88, pp. 187 – 201, 2017.
- [5] N. Haouchine, J. Dequidt, I. Peterlik, E. Kerrien, M.-O. Berger, and S. Cotin, "Towards an accurate tracking of liver tumors for augmented reality in robotic assisted surgery," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2014, pp. 4121–4126.
- [6] L. Royer, A. Krupa, G. Dardenne, A. Le Bras, E. Marchand, and M. Marchal, "Real-time Target Tracking of Soft Tissues in 3d Ultrasound Images Based on Robust Visual Information and Mechanical Simulation," *Medical Image Analysis*, vol. 35, pp. 582–598, Jan. 2017.
- [7] P. Patlan-Rosales and A. Krupa, "Strain estimation of moving tissue based on automatic motion compensation by ultrasound visual servoing," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2017, pp. 2941–2946.
- [8] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—a survey," *Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2014.
- [9] J. Smolen and A. Patriciu, "Deformation planning for robotic soft tissue manipulation," in *Proceedings of the IEEE International Conference on Advances in Computer-Human Interactions*, 2009, pp. 199–204.
- [10] D. Navarro-Alarcón, Y. Liu, J. Romero, and P. Li, "On the visual deformation servoing of compliant objects: Uncalibrated control methods and experiments," *The International Journal of Robotics Research*, vol. Vol 33, no. 11, pp. 1462– 1480, 2014.
- [11] D. Navarro-Alarcón, H. M. Yip, Z. Wang, Y. H. Liu, F. Zhong, T. Zhang, and P. Li, "Automatic 3-D Manipulation of Soft Objects by Robotic Arms With an Adaptive Deformation Model," *Transactions on Robotics*, vol. Vol 32, no. 2, pp. 429–441, 2016.
- [12] J. Das and N. Sarkar, "Autonomous shape control of a deformable object by multiple manipulators," *Journal of Intelligent and Robotic Systems*, vol. Vol 62, no. 1, pp. 3–27, 2011.
- [13] Z. Zhang, T. M. Bieze, J. Dequidt, A. Kruszewski, and C. Duriez, "Visual servoing control of soft robots based on finite element model," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2017, pp. 2895–2901.
- [14] B. Frank, C. Stachniss, N. Abdo, and W. Burgard, "Efficient motion planning for manipulation robots in environments with deformable objects," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2011, pp. 2180–2185.
- [15] Y. Adagolodjo, L. Goffin, M. De Mathelin, and H. Courtecuisse, "Inverse real-time finite element simulation for robotic control of flexible needle insertion in deformable tissues," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2016, pp. 2717–2722.
- [16] D. Berenson, "Manipulation of deformable objects without modeling and simulating deformation," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2013, pp. 4525–4532.
- [17] J. Alonso-Mora, R. Knepper, R. Siegwart, and D. Rus, "Local motion planning for collaborative multi-robot manipulation of deformable objects," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2015, pp. 5495–5502.
- [18] D. Navarro-Alarcón, Y.-H. Liu, J. G. Romero, and P. Li, "Model-free visually servoed deformation control of elastic objects by robot manipulators," *Transactions on Robotics*, vol. Vol 29, no. 6, pp. 1457–1468, 2013.
- [19] C. G. Broyden, "A class of methods for solving nonlinear simultaneous equations," *Mathematics of computation*, vol. 19, no. 92, pp. 577–593, 1965.
- [20] D. Navarro-Alarcon and Y.-H. Liu, "Fourier-based shape servoing: A new feedback method to actively deform soft objects into desired 2-d image contours," *Transactions on Robotics*, vol. 34, no. 1, pp. 272–279, 2018.
- [21] K. Hosoda and M. Asada, "Versatile visual servoing without knowledge of true jacobian," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, vol. 1, 1994, pp. 186–193.
- [22] M. De Mathelin and R. Lozano, "Robust adaptive identification of slowly time-varying parameters with bounded disturbances," *Automatica*, vol. 35, no. 7, pp. 1291–1305, 1999.
- [23] A. Krupa, "Commande par vision d'un robot de chirurgie laparoscopique," Ph.D. dissertation, Institut National Polytechnique de Lorraine, July 2003.
- [24] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik, and S. Cotin, "SOFA: A Multi-Model Framework for Interactive Physical Simulation," in *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, ser. Studies in Mechanobiology, Tissue Engineering and Biomaterials, Y. Payan, Ed. Springer, 2012, vol. 11, pp. 283–321.
- [25] ViSP. Blob tracking. [Online]. Available: <http://visp-doc.inria.fr/doxygen/visp-daily/classvpDot2.html>
- [26] P.-J. Huber, *Robust Statistics*, Wiley, Ed., 1981.
- [27] M. Pressigout and E. Marchand, "Real-time 3d model-based tracking: Combining edge and texture information," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006, pp. 2726–2731.
- [28] N. Cressie and D. M. Hawkins, "Robust estimation of the variogram: I," *Journal of the International Association for Mathematical Geology*, vol. 12, no. 2, pp. 115–125, 1980.
- [29] N. Aspert, D. Santa-Cruz, and T. Ebrahimi, "Mesh: Measuring errors between surfaces using the hausdorff distance," in *Proceedings of the IEEE International Conference on multimedia and expo*, vol. 1, 2002, pp. 705–708.